

# Руководство по LABPP\_Automat для ARCHICAD

## Table of contents

---

Введение .....	18
Особенности языка .....	22
Моя первая программа .....	22
Кодировка у файлов программных модулей .....	23
Структура каталогов у LabPP_Automat .....	24
Конфигурационный файл config.cpp у конфигурации .....	26
create_button .....	26
create_iconbutton .....	27
set_palette_size_and_message_place .....	27
Списки для выборки элементов из проекта .....	27
Генезис свойств .....	28
Окна LABPP .....	28
Семантика языка C++ для ARCHICAD .....	29
Общая структура программы .....	29
Функция main() .....	29
Идентификаторы .....	30
Типы переменных, видимость и виды .....	30
Пакетное объявление и обращение к переменным через идентификаторы, задаваемые текстом - var_by_txt() .....	31
init .....	32
is_exist .....	34
set .....	34
get .....	34
Комментарии .....	35
Организация циклов .....	35
Цикл for .....	35
Цикл do-while .....	35
Цикл while .....	35
Логические операции .....	35
Условные переходы .....	36
Оператор if .....	36
Конструкция if-else .....	36
Конструкция if-else if-else .....	36
Оператор switch .....	36
Создание собственных функций (подпрограмм) .....	36
Директива #include .....	37
Особая конструкция cpp для ARCHICAD - управляемый #include .....	38
Директива #pragma region .....	38
Поиск по шаблону (коды REGEX) .....	38
Логическое И и логическое ИЛИ в регулярных выражениях .....	39
Функции для работы с оболочкой shell_func .....	40
get_path .....	40
set_cur_dir .....	40
shellexecute .....	41
ExecCmd .....	41
file .....	42
copy .....	42

is_exist .....	43
delete .....	43
get_SLANG_CODE .....	43
Управление окнами labpp .....	44
Буффер обмена Windows .....	44
clipboard_get .....	44
clipboard_set .....	44
Функции автоматизации – Automate .....	44
CloseCurWindow .....	44
Zoom .....	44
GoToView .....	44
ChangeWindow .....	45
ShowSelectionIn3D .....	45
ShowAllIn3D .....	45
ZoomToElements .....	45
Redraw .....	45
Rebuild .....	45
Функции окружения - Environment .....	45
AddLibraries .....	45
Story_GoTo .....	46
Функции для работы с базами проекта - database .....	46
GetCurrent .....	46
Select .....	46
SetTypeID .....	46
GetTypeID .....	47
GetLocOrigo .....	47
get_section_infoltext .....	48
get_sectionmarkers2dpositions .....	48
GetRVMChanges .....	49
GetRVMLayoutCurrentRevisionChanges .....	49
GetTypeName .....	50
GetSizeX, GetSizeY .....	50
leftMargin,topMargin,rightMargin,bottomMargin .....	51
layoutPageNumber .....	51
layoutName .....	51
Вывод в окно сообщений .....	52
cout .....	52
coutvar .....	52
setcfg("COUT",...) .....	52
Файловые операции .....	52
open .....	52
write .....	52
read .....	53
close .....	53
SetIsUNICODE .....	53
SetIsUTF16 .....	53
Строковые функции .....	54
strcmp .....	54
tolower .....	54
toupper .....	54

alltrim .....	54
strcontains .....	55
strreplace .....	55
strlen .....	55
substr .....	55
strposcharset .....	55
strposcharset .....	56
strleft .....	56
strright .....	56
str_regex_search .....	56
str_regex_match .....	57
str_regex_replace .....	57
str_regex_substr .....	58
straddslashesregexcodes .....	59
strremovebaddelimiters .....	59
str2threekeys .....	59
strCompareLikeABC10_a .....	59
strreplace_escape_u .....	60
strisnumber .....	60
smallnums .....	60
ts_text_mc .....	60
Объект ac_element_guid (ts_guid) .....	61
Объект ac_element .....	61
ac_request .....	61
get_floor_name_by_floor_index .....	61
get_floor_index_by_floor_name .....	61
store_cur_element_to_descr .....	62
set_current_element_from_descr .....	62
set_element_by_guidstr_as_current .....	62
get_guid_from_element .....	62
load_element_from_guid .....	63
load_elements_list .....	63
set_current_element_from_list .....	63
add_elements_list .....	64
load_elements_list_from_selection и add_elements_list_from_selection .....	64
load_elements_list_curdb .....	64
clear_list .....	64
store_current_element_to_list .....	64
get_loaded_elements_list_count .....	64
select_elements_from_list .....	64
Слой - layer .....	65
create .....	65
get_index .....	65
set_layer_visible .....	65
get_element_overall_dimensions .....	65
get_quantity_value .....	66
get_layer_by_substring .....	71
elem_user_property .....	72
get .....	72
set .....	73

create .....	73
Property_Import .....	74
get_object_property_value .....	74
set_object_property_value .....	75
set_object_property_value_curdb .....	75
get_element_value .....	75
set_element_value .....	76
assign_element_values .....	76
load_element_default_values .....	76
set_element_infidtext .....	76
create_element_on_project .....	77
get_element_infidtext .....	77
project_property .....	77
get .....	77
CalcUnitsAreaDecimals .....	77
autotext .....	77
get .....	77
set .....	78
create .....	78
get_as_table .....	78
interface_input2point .....	79
interface_input2dline .....	79
interface_input3dline .....	79
interface_input2dpoly .....	79
get_levelfromprojectnull_by_floorindexandlevel .....	80
get_floorindexandlevel_by_levelfromprojectnull .....	82
get_storyinfo_as_table .....	82
elem_classification_get_item_value .....	83
elem_classification_get_parent_item_value .....	83
hotlinks .....	83
GetHotlinkNodes .....	83
geometry_calc_2d .....	86
is_point_on_element_polygon .....	86
is_curelem_inside_element_polygon .....	86
is_point_on_element_refside .....	87
where_C_leftorright .....	87
getDistPointFrom2dLine .....	87
geometry_calc_3d .....	87
RotatePoint3dAroundVect3d .....	87
GetBounds .....	88
classification .....	88
find_item_by_name .....	88
check_class_is_inbranch .....	88
CreateClassificationSystem .....	88
CreateClassificationItem .....	89
DeleteClassificationItem .....	89
set_item_to_curelem .....	89
GetClassificationSystems .....	90
GetClassificationSystem_guid_by_name .....	90
Classification_Import .....	91

GetClassifSystem_By_ClassTtemGuid .....	91
do_move_rot_and_scale_elemlist .....	92
clearmessagewindow .....	93
get_labpp_product_name .....	93
ac_request_special .....	93
get_pie .....	93
load_elements_list .....	93
add_elements_list .....	95
load_elements_list_from_selection .....	95
add_elements_list_from_selection .....	95
load_elements_list_curdb .....	95
copy_elements_to_list_from_list_by_filter .....	96
check_cur_element_by_filter .....	96
GetMaterialSurfaceSquare .....	96
GetMaterialSurfaceSquareTable .....	96
get_element_value .....	97
buildingMaterialName_thickness .....	97
buildingMaterialName_used .....	98
set_element_value .....	98
set_element_value_curdb .....	98
RunGDLParScript .....	98
get_building_material .....	98
get_build_material_value .....	99
get_material .....	99
load_zonerelatedwallstolist .....	100
load_zonerelatedwallparttable .....	100
Attribute_GetNum .....	100
clipboard_set .....	101
clipboard_get .....	101
element_user_data .....	101
read .....	101
write .....	101
write_to_file .....	101
clear .....	103
delete_variable .....	103
get_variable_value .....	103
set_variable_value .....	103
linkingElems .....	103
uplinkBiWardByFlags .....	103
getLinkedElemsByFlags .....	104
set_TSMAX_ELEMLIST_NUMBER .....	104
get_TSMAX_ELEMLIST_NUMBER .....	104
set_elemlistname .....	104
Get_Elements_Collisions .....	105
delete_elements_from_list .....	105
delete_elements_from_project .....	105
ac_request_attributes .....	106
Таблица типов атрибутов .....	106
create_attr .....	106
get_num .....	107

delete_attr .....	107
set_attr_value .....	107
get_attr_value .....	107
ac_typeidfromstring .....	108
ac_getresvaluetype .....	108
ac_getstrvalue .....	108
ac_getnumvalue .....	108
ac_getupvaluestatus .....	108
Гравитация на поверхность .....	109
do_elements_landing - приземление элементов .....	109
do_surface_landing - приземление по точкам .....	109
do_point_landing - приземление точки X,Y .....	109
3d сетки .....	110
get_lands_combined_contour_coords .....	110
do_create_contour_around_points .....	110
get_lands_topcoords .....	111
do_create_mesh_by_points_and_lines .....	111
Диалоги .....	112
Сообщения или выбор варианта - tsalert .....	112
Поиск и выбор файла - dialog_get_filename .....	113
editdoubledialog - ввод числа .....	113
editintdialog - ввод целого числа .....	114
editstringdialog - ввод строки .....	114
editstringlistdialog - выбор строки из списка .....	115
editcheckstringsdialog - выбор нескольких строк из списка .....	116
editbooldialog - выбор Да/Нет .....	117
editstringlistandcustomdialog - ввод строки или выбор из списка .....	117
Работа с Excel .....	119
excel_attach .....	119
excel_detach .....	119
excel_putnumvalue .....	119
excel_putstrvalue .....	119
excel_select_range .....	119
excel_visible .....	120
excel_speedup .....	120
excel_getnumvalue .....	120
excel_getstrvalue .....	120
excel_request .....	120
set_column_width .....	120
get_column_width .....	120
set_row_height .....	120
get_row_height .....	121
set_borders .....	121
put_selection_values .....	121
put_selection_fontvalues .....	121
get_selection_area .....	122
merged_cell_info .....	122
is_merge_cells .....	122
set_backcolor .....	123
get_backcolor .....	123

set_interior .....	123
get_interior .....	123
selection_varvalues .....	123
selection_font_varvalues .....	123
sheet_select .....	123
range_copy .....	123
booknamedcell .....	123
save_active_sheet_as_csv .....	123
save_copy_as .....	124
workbook_open .....	124
workbook_close .....	125
работа с примечаниями .....	125
getcomment .....	125
clearcomments .....	125
setcomment .....	125
Загрузка и выгрузка табличных данных .....	125
excel_chart - диаграммы в Excel .....	125
AddChart .....	125
SetChartType .....	125
SetChartTitle .....	127
SetSourceDataRange .....	127
Пример .....	128
Работа с Word .....	128
word_attach .....	128
word_detach .....	128
word_visible .....	128
word_request .....	128
НСчитывание и запись полей переменных - docfield .....	129
get .....	129
set .....	129
update_all_docfields .....	129
Работа через ts_idispatch .....	129
Работа с AutoCAD .....	130
autocad_request .....	130
attach .....	130
detach .....	131
specify_color_RGB .....	131
draw_text .....	131
draw_polyline .....	131
draw_line .....	132
draw_arc .....	132
Пример - нарисовать контуры зон из ARCHICAD в AutoCAD полилиниями ...	132
Функция object .....	134
create .....	134
delete .....	134
Объект ts_table .....	135
Функция ts_table .....	135
init (clear) .....	135
clear_rows .....	135
add_column .....	135



is_column_exist .....	136
set_first_key .....	136
add_row .....	136
add_row_sum .....	137
sort .....	137
sort_back .....	138
set_columns_to_sort_as_LikeABC10_a .....	138
search .....	139
search_regex .....	139
search_extended .....	139
search_extended_table .....	139
search_bounds .....	140
select_row .....	140
seek_from_top, seek_from_bottom, seek_from_current .....	140
add_row_from_eq_table .....	140
add_rows_from_eq_table_by_filter .....	141
add_rows_from_eq_table_by_regex_filter .....	141
add_rows_from_eq_table_by_extended_filter .....	141
add_rows_from_eq_table_by_extended_filter_table .....	142
get_value_of .....	142
get_rows_count .....	142
get_columns_count .....	142
set_columnstypes .....	143
get_heading_of .....	143
print_to_str .....	143
set_columns_to_add_sum .....	143
Функции для выгрузки и загрузки таблиц .....	144
export_to_excel .....	144
import_from_excel .....	145
import_columns_from_excel .....	145
import_columns_from_table .....	145
import_columns_from_dialogcontrol .....	145
import_from_dialogcontrol .....	145
export_to_dialogcontrol .....	146
export_to_dialogcontrol_by_filter .....	146
export_to_dialogcontrol_by_regex_filter .....	146
export_to_dialogcontrol_by_extended_filter .....	146
export_to_dialogcontrol_by_extended_filter_table .....	147
resettotffromexport .....	147
export_to_LABPP_Table .....	148
set_columns_width_when_export .....	149
mapping_columns_to_export .....	149
get_base_colindex_by_mapped_column .....	150
get_count_of_mapped_columns .....	151
export_to_csv .....	151
import_from_csv .....	152
export_to_ac_object_parameter .....	152
set_column_user_value .....	153
get_column_user_value .....	153
set_column_user_genesis .....	153

get_column_user_genesis .....	153
delete_rows .....	154
set_value_of_toallrows .....	154
Функции стандартной библиотеки .....	154
Преобразование .....	154
atoi .....	154
itoa .....	154
atof .....	154
ctos .....	154
grad_to_radian .....	155
radian_to_grad .....	155
sprintf .....	155
ecvt_french .....	155
ecvt .....	156
Математические функции .....	156
abs .....	156
max .....	156
min .....	156
rand .....	156
ln .....	156
log .....	156
sqrt .....	157
sqr .....	157
pow .....	157
percent .....	157
tsround .....	157
tsround_best .....	157
floor .....	157
ceil .....	157
math_const .....	157
inrange .....	158
Тригонометрические функции .....	158
cos .....	158
sin .....	158
arcsin .....	158
arccos .....	158
tg .....	158
arctg .....	158
ctg .....	158
arcctg .....	158
Функции геометрического преобразования .....	159
is_curelem_inside_element_polygon .....	159
is_point_on_element_polygon .....	159
rotate_point_and_move .....	160
get_cross_point_of_2lines .....	160
get_rot_and_move_point .....	160
get_length_2point .....	160
is_point_on_line .....	161
get_line_angle .....	161
Специальные функции .....	161

codemeter - измерение времени выполнения фрагмента кода .....	161
ac_save_messages_to_file - сохранить текст из окна сообщений в файл .....	161
Связь с квартирографией .....	162
solaris_test .....	162
get_flat_rooms .....	162
runtimecontrol .....	162
workline .....	162
Команды для управления в приложении LabPP_Calc .....	164
interface .....	164
calc_field .....	164
get_active .....	164
Внешние переменные .....	165
var_extern_set .....	165
var_extern_get .....	165
var_extern_delete .....	165
Объект ts_idispatcher .....	166
CreateInstance .....	168
attach .....	168
detach .....	168
AutoWrap .....	169
Управление выполнением и поведением программы .....	169
Прерывание выполнения длительных процессов ac_process_window .....	169
Init .....	170
SetPhase .....	170
IsCanceled .....	170
Close .....	170
throw_process .....	171
Поведение программы при ошибках считывания/записи переменных элементов	
172	
Остановка процесса помещением файла в каталоге .....	173
Объект ts_variant .....	174
get_pdispVal .....	174
get_value_simple .....	174
set_value_simple .....	174
run_cpp - запуск другой программы .....	175
run_from_file .....	175
run_from_variable .....	175
Получение аргументов внутри программы .....	175
Обработка ошибок .....	176
Коды ошибок .....	176
Функции интерфейса LabPP_Automat .....	176
create_iconbutton .....	176
create_button .....	177
set_palette_size_and_message_place .....	177
Диалоги на основе окон .....	177
Класс ts_dialog .....	177
init_dialog .....	177
set_as_main_panel .....	177
SetClientWH .....	178
SetTitle .....	178

SendCloseRequest .....	178
PostCloseRequest .....	178
eventreaction .....	178
Invoke .....	178
set_sDescr .....	179
Класс ts_dialogcontrol .....	179
Button .....	179
SetText .....	179
GetText .....	179
SetFontSzecodeStyle .....	179
IconButton .....	180
TextEdit .....	180
SetText .....	180
GetText .....	180
SetFontSzecodeStyle .....	180
RealEdit .....	180
SetFontSzecodeStyle .....	180
CheckBox .....	181
SetText .....	181
GetText .....	181
SetCheck .....	181
GetCheck .....	181
SetFontSzecodeStyle .....	181
LeftText, CenterText, RightText .....	182
SetText .....	182
GetText .....	182
SetFontSzecodeStyle .....	182
PopUp .....	182
SelectItem .....	183
DisableItem .....	183
DeleteItem .....	183
EnableItem .....	183
InsertItem .....	183
AppendItem .....	183
SetFontSzecodeStyle .....	183
GetItemIndexByText .....	184
GetItemIndexByText .....	184
RadioButton .....	184
IsSelected .....	184
Select .....	184
SetText .....	184
GetText .....	184
SetFontSzecodeStyle .....	185
IconRadioButton .....	185
IsSelected .....	185
Select .....	185
SingleSelListBox .....	185
RepaintBackgroundItemsByColumnValue .....	185
SetItemBackgroundColor .....	186
SetItemColor .....	186

SelectItem .....	186
GetTabItemText .....	186
GetMouseClickedPosXY .....	187
GetTabFieldPosition .....	187
SetTabItemIcon .....	187
SetTabItemText .....	187
GetTabItemIconId .....	187
SetTabItemIconId .....	187
SetOnTabItem .....	188
SetTabFieldCount .....	188
SetHeaderItemSize .....	188
SetTabFieldProperties .....	188
SetItemHeight .....	188
SetHeaderItemSizeableFlag .....	188
SetHeadersSizeableFlag .....	188
SetHeaderItemText .....	189
SetHeaderItemSize .....	189
InsertItem .....	189
AppendItem .....	189
DisableItem .....	189
EnableItem .....	189
DeleteItem .....	189
SetFontSizeModeStyle .....	189
MultiSellListBox .....	190
SelectItem .....	190
GetTabItemText .....	190
GetMouseClickedPosXY .....	190
GetTabFieldPosition .....	190
SetTabItemIcon .....	190
SetTabItemText .....	190
GetTabItemIconId .....	191
SetTabItemIconId .....	191
SetOnTabItem .....	191
SetTabFieldCount .....	191
SetTabFieldCount .....	191
SetHeaderItemSize .....	191
SetTabFieldProperties .....	191
SetItemHeight .....	192
SetHeaderItemSizeableFlag .....	192
SetHeadersSizeableFlag .....	192
SetHeaderItemText .....	192
SetHeaderItemSize .....	192
InsertItem .....	192
AppendItem .....	193
DisableItem .....	193
EnableItem .....	193
DeleteItem .....	193
SetFontSizeModeStyle .....	193
MultiLineEdit .....	193
SetText .....	194

GetText .....	194
SetFontSizecodeStyle .....	194
IconItem .....	194
IconCheckBox .....	194
SetCheck .....	194
GetCheck .....	194
PushCheck .....	195
SetText .....	195
GetText .....	195
SetCheck .....	195
GetCheck .....	195
SetFontSizecodeStyle .....	195
IconPushCheck .....	195
SetCheck .....	195
GetCheck .....	196
IconPushRadio .....	196
SingleSelTreeView .....	196
SelectItem .....	196
TreeViewInsertItem .....	196
DeleteItem .....	196
InsertItem .....	197
SetFontSizecodeStyle .....	197
GetTVItemIconId .....	197
SetTVItemIconId .....	197
GetItemCount .....	197
import_classificator .....	198
MultiSelTreeView .....	198
SelectItem .....	198
TreeViewInsertItem .....	199
DeleteItem .....	199
InsertItem .....	199
AppendItem .....	199
SetFontSizecodeStyle .....	199
GetTVItemIconId .....	199
SetTVItemIconId .....	200
GetItemCount .....	200
SingleSelListView .....	200
DeleteItem .....	200
InsertItem .....	200
AppendItem .....	200
SetFontSizecodeStyle .....	200
GetItemCount .....	201
MultiSelListView .....	201
DeleteItem .....	201
InsertItem .....	201
AppendItem .....	201
SetFontSizecodeStyle .....	201
GetItemCount .....	201
Splitter .....	201
Attach_controlled_element .....	201

ProgressBar .....	202
SetMin .....	202
SetMax .....	202
SetValue .....	202
GetMin .....	202
GetMax .....	202
GetValue .....	202
NormalTab .....	202
SelectItem .....	202
DisableItem .....	202
EnableItem .....	203
DeleteItem .....	203
InsertItem .....	203
AppendItem .....	203
SetFontSizeModeStyle .....	203
TabPage .....	203
SetToolTip .....	203
GroupBox .....	204
SetFontSizeModeStyle .....	204
PushMenuCheck .....	204
SetItemIcon .....	207
GetCheck .....	207
SetCheck .....	207
SetItemCount .....	207
SelectItem .....	207
GetSelectedItem .....	207
SetToolTip .....	207
SetToolTipItem .....	208
WasStateChange .....	208
PreviousMenuItem .....	208
PushMenuRadio .....	208
SetItemIcon .....	208
Select .....	209
IsSelected .....	209
SetItemCount .....	209
SelectItem .....	209
GetSelectedItem .....	209
SetToolTip .....	209
SetToolTipItem .....	209
PreviousMenuItem .....	209
WasSelectionChange .....	209
Eventreaction .....	209
SetAnchorToPanelResize .....	212
Hide .....	212
Show .....	212
SetPosition .....	212
Работа со структурами данных в формате JSON .....	212
Класс ts_json .....	212
init (clear) .....	212
DecodeFromURL .....	213

DecodeFromString .....	213
EncodeToStr .....	214
Класс ts_array .....	214
Класс ts_class .....	214
Класс ts_num .....	214
Класс ts_bool .....	214
Класс ts_string .....	214
ReadFromURL .....	214
TSVariator .....	214
Редактор таблиц LABPP_Table для ARCHICAD .....	216
Структура данных объекта LABPP_Table (для программистов) .....	220
Непереносимые пробелы и дефисы .....	221
Работа с СУБД MySQL .....	222
connection_init .....	222
connection_open .....	223
real_query .....	223
store_result .....	224
get_num_fields .....	224
get_row_count .....	224
export_to_table .....	224
free_result .....	225
connection_close .....	225
Пример - получение таблицы с сайта из базы данных MySQL в диалог ARCHICAD .....	225
Работа с СУБД PostgreSQL .....	228
init .....	228
pqconnectdb .....	229
pqexec .....	229
pqnfields .....	229
pqntuples - кол-во строк .....	230
export_to_table .....	230
pqclear .....	230
pqfinish .....	231
Пример - получение таблицы с сайта из базы данных PostgreSQL в диалог ARCHICAD .....	231
Работа с СУБД SQLite .....	234
init .....	234
open_db .....	234
prepare_v2_query - запрос к базе данных .....	234
column_count .....	235
row_count .....	235
export_to_table .....	235
finalize_stmt .....	236
close_db .....	236
Пример - получение таблицы с сайта из базы данных PostgreSQL в диалог ARCHICAD .....	236
Искусственный интеллект .....	239
Нейросеть на основе перцептрона Розенблатта - ts_aipercrozen .....	239
init .....	239
task_add_to_list .....	240



learn_task_list .....	241
run_task_list .....	241
get_task_result_to_table .....	241
task_list_clear .....	241
store_matrices и restore_matrices .....	242
clear .....	242
ts_aipercrozen_matrices .....	242
init .....	242
get_from_percrozen .....	242
set_to_percrozen .....	242
export_to_table .....	243
import_from_table .....	243
clear .....	244
Пример .....	244
BIM Master .....	245
breakpoint .....	247
programcodegen .....	247
get_bimmaster_cur_selected_parameter_info .....	247
get_bimmaster_cur_element_guidastext .....	248
show_bimmaster_as_codegen .....	248
show_bimmaster .....	248
Уроки .....	249
Вычислить середину высоты элемента Морф относительно 0 проекта и записать в поле LabPP_Calc .....	249
Заполнить таблицу с расчетом плитки по элементам Штриховка и записать общую площадь в поле LabPP_Calc .....	251
FAQ (часто задаваемые вопросы) .....	253
Как посмотреть значения параметров, заданных формулами .....	253

## Введение

---



# LABPP.net

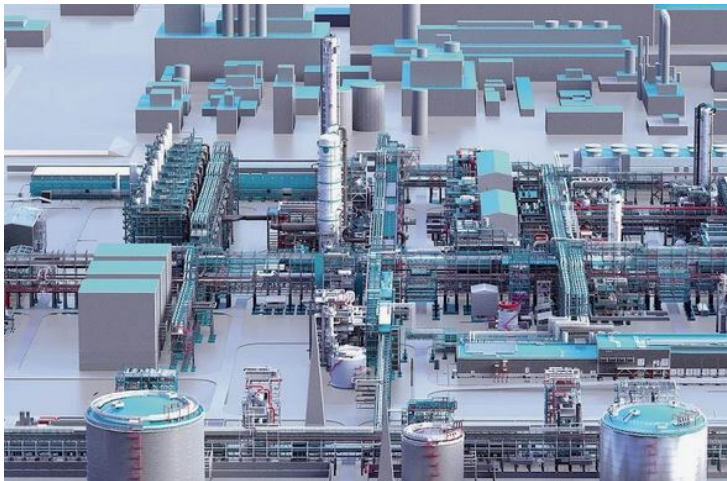
## Automate ARCHICAD

ARCHICAD –

ARCHICAD.

2007

LABPP



- [Adobe Acrobat PDF](#)
- [Windows help- CHM](#)
- [Word DOC](#)
- [EPUB](#)
- [HTML](#)

ARCHICAD

SQL

" - ".

Microsoft®  
SQL Server®

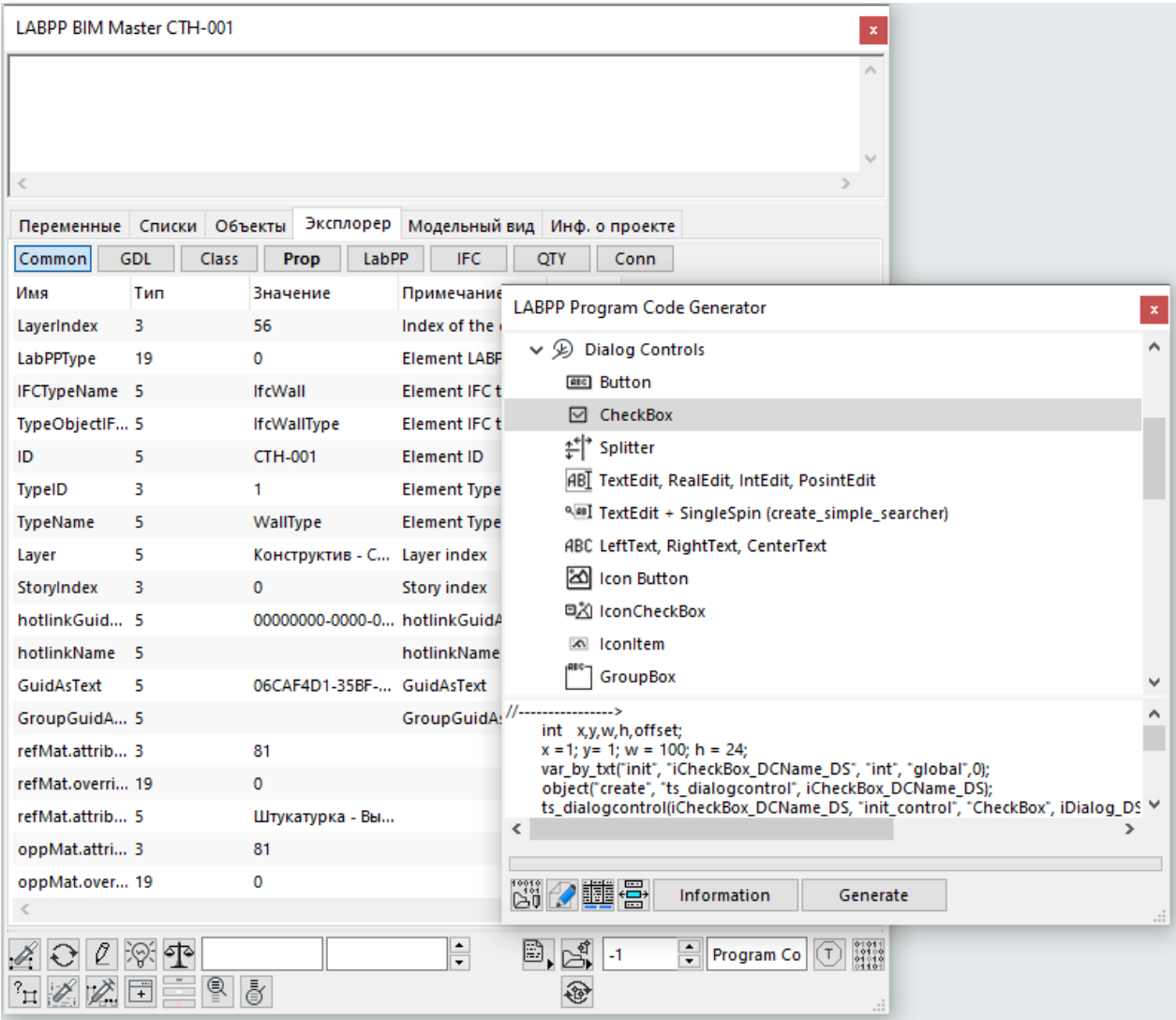
PostgreSQL



SQLite

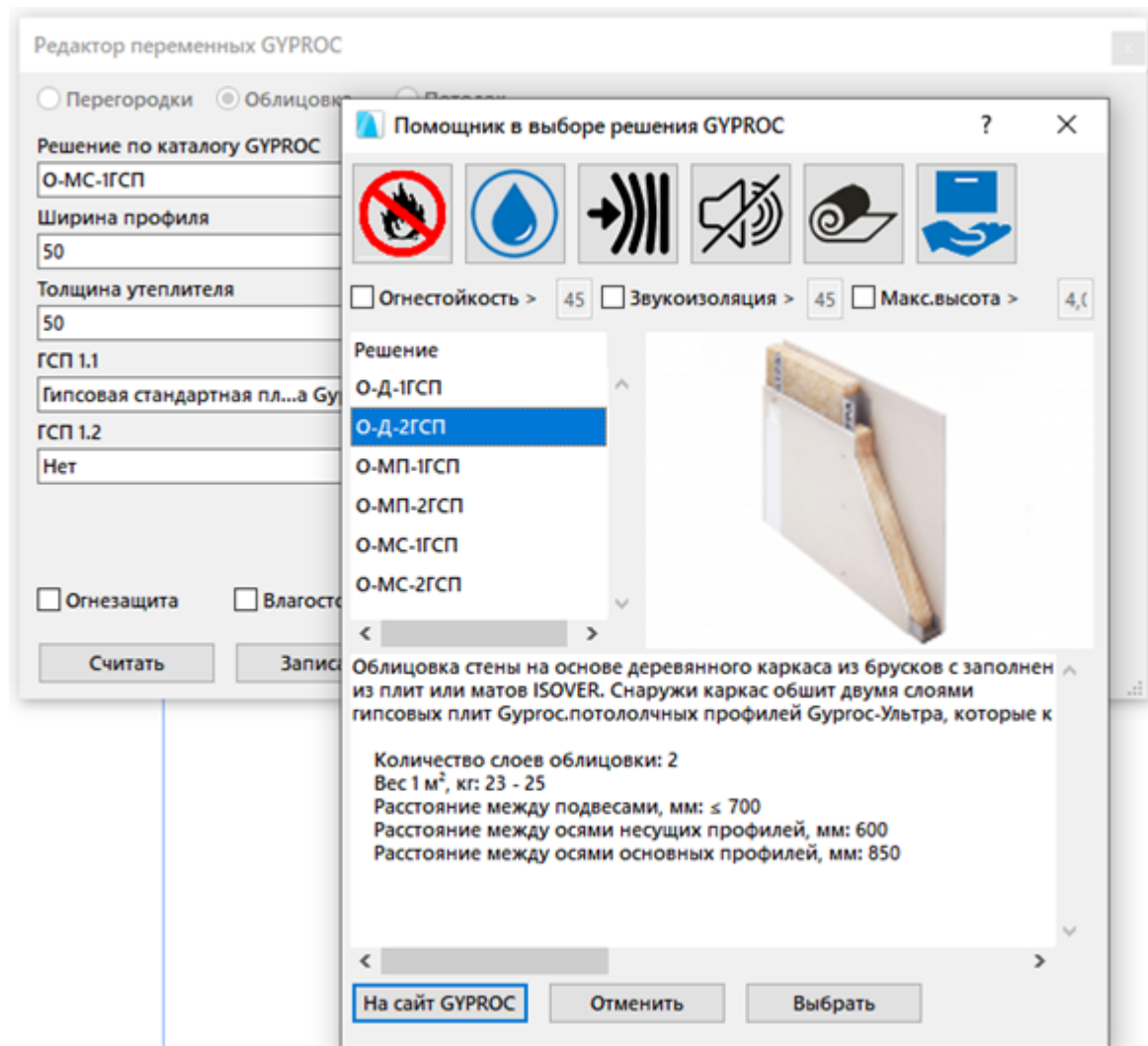
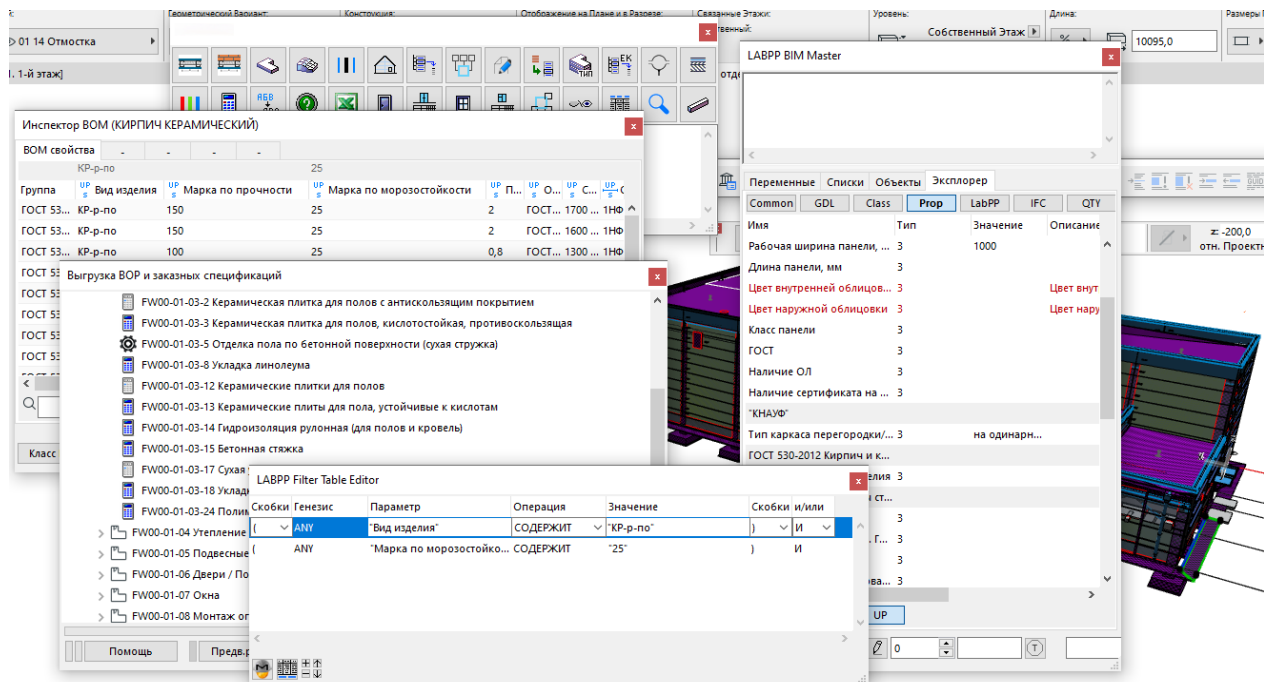


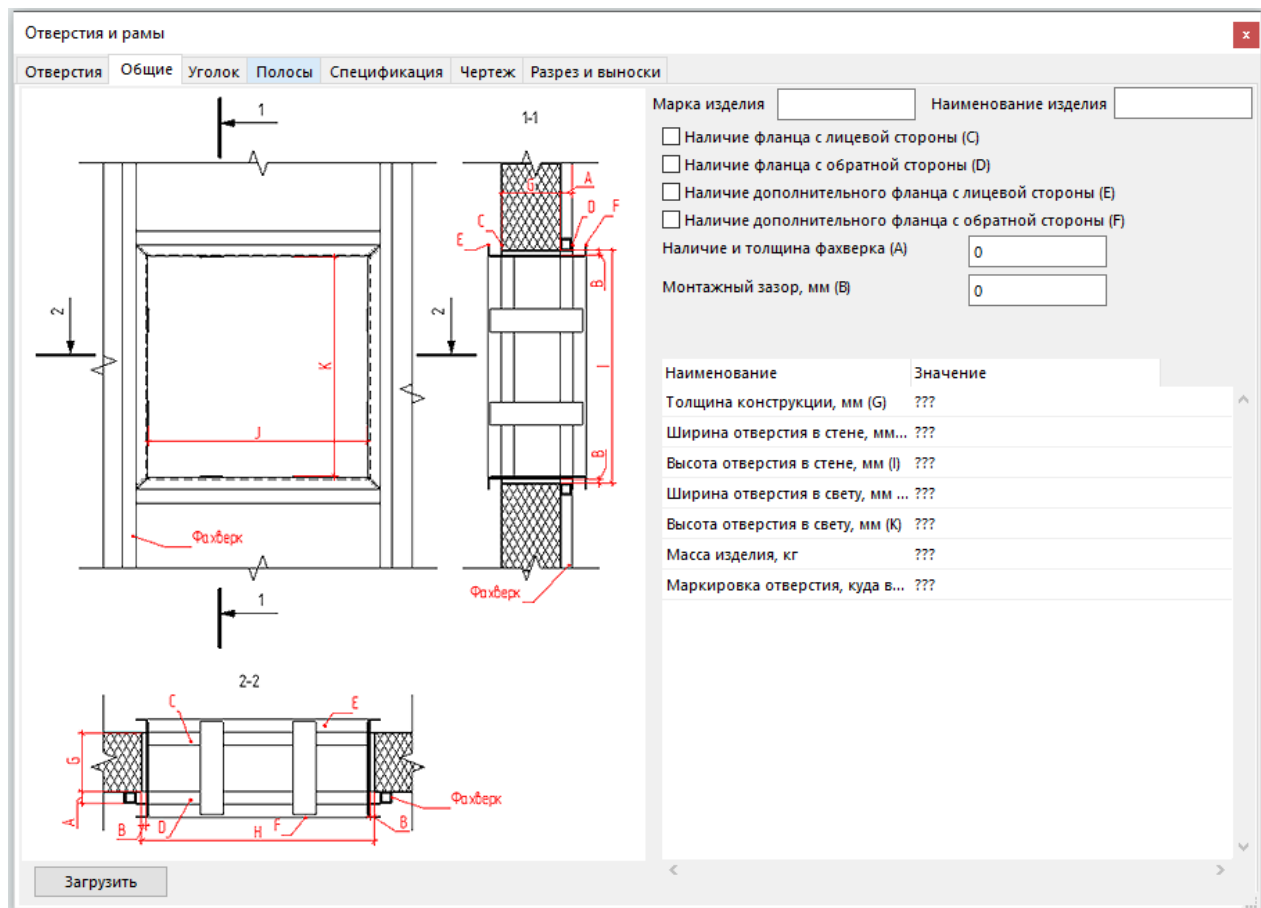
BIM Master.



(   ) .  
(   )  
-

LabPP\_Automat:





## Особенности языка

LabPP\_Automat

++.

•

•

•  
•  
•  
•  
•

'

•  
,

LabPP\_Automat

string.

MyStringFunc,

•

•

```
string MyString = MyStringFunc();
```

" " MyString

MyStringFunc().

:

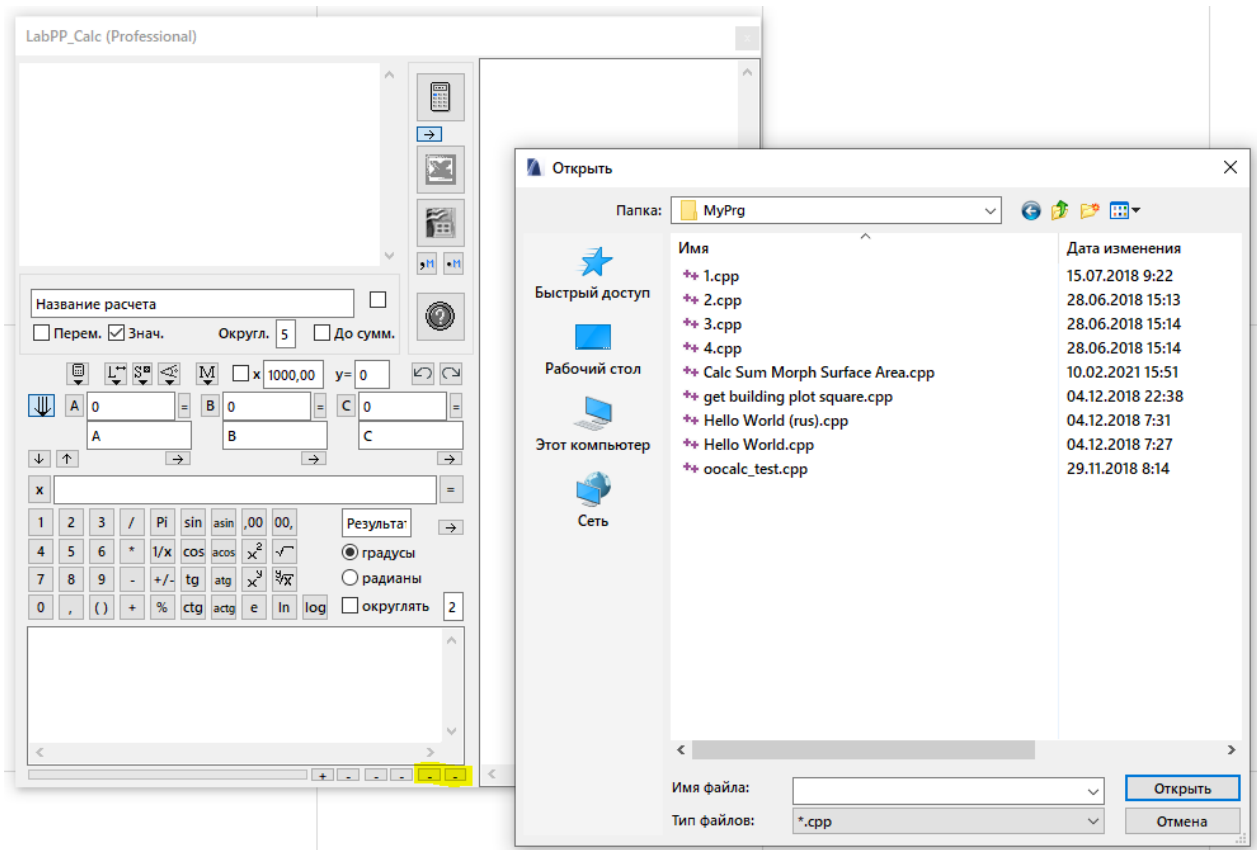
- 1.
- 2.
- 3.

" "

## Моя первая программа

```
int main()
{
    cout << "Здравствуй Мир!)))";
}
```

labpp

[LabPP\\_Calc](#)

LabPP.

4

config.cpp).

## Кодировка у файлов программных модулей

ANSI.

" 2", UTF8

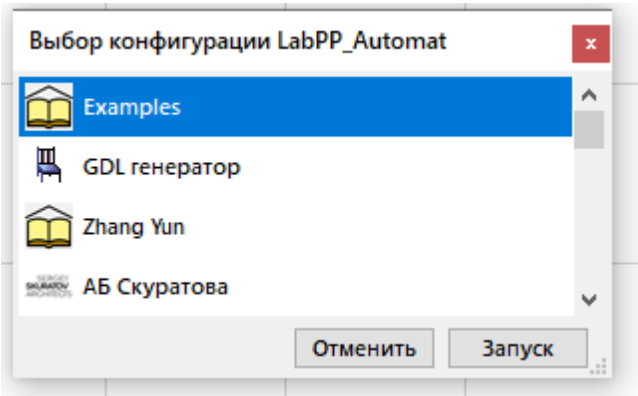
Excel

LabPP\_Automat

UTF-8

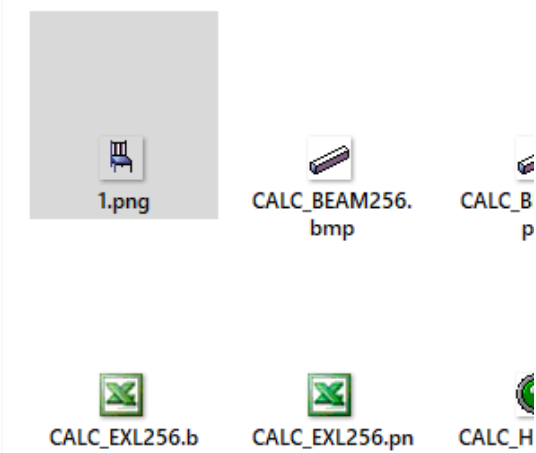






tsimages -

1.png



tsprg

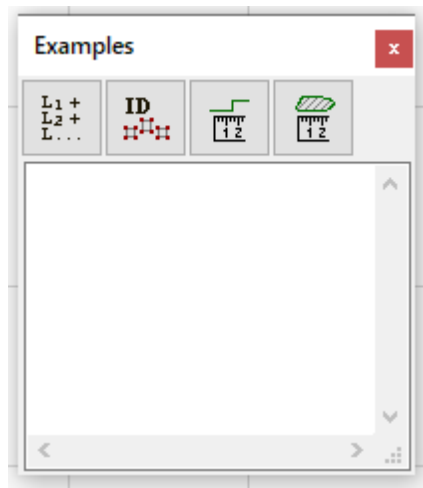
config.cpp

```

** config.cpp
** get_points_array.cpp
** make_extrude_code.cpp
** make_revolve_code.cpp
** make_ruled_code.cpp
** make_sweep_code.cpp
** make_tube_code.cpp
** multiple_around_center.cpp
```

config.cpp

( \*.cpp).



LabPP\_AutomatAC...  
LabPP\_Automat.key.

LabPP\_Automat.apx

## Конфигурационный файл config.cpp у конфигурации

config.cpp

### create\_button

```

:

ac_request("create_button", string button_caption, int start_x, int start_y, int width,
int height, string tooltip, string name_cpp);

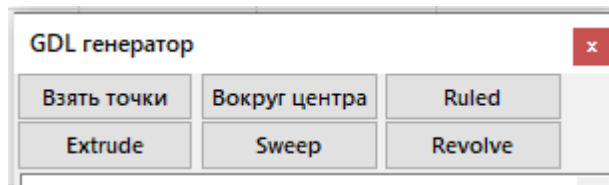
или
ac_request("create_button", string button_caption, int start_x, int start_y, int width,
int height, string tooltip, string name_cpp, int iarg, double darg, string sarg);

:
button_caption - ;
start_x, start_y - ;
width, height - ;
tooltip - ;
name_cpp - ( tsprg).
iarg, darg sarg - , ,

run_cpp("get_args"... :

int iArg1;
double dArg2;
string sArg3;
run_cpp("get_args", iArg1, dArg2, sArg3);

```

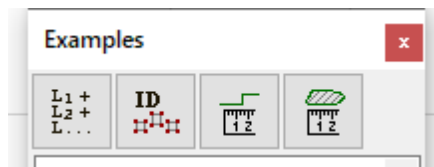


## create\_iconbutton

```

        :
ac_request("create_iconbutton", string icon_filename, int start_x, int start_y, int
width, int height, string tooltip, string name_cpp);
или
ac_request("create_iconbutton", string icon_filename, int start_x, int start_y, int
width, int height, string tooltip, string name_cpp, int iarg, double darg, string sarg);
        :
icon_filename -
start_x, start_y -
width, height -
tooltip -
name_cpp -
iarg, darg sarg -
run_cpp("get_args"... :
int iArg1;
double dArg2;
string sArg3;
run_cpp("get_args", iArg1, dArg2, sArg3);

```



# Руководство по LABPP\_Automat для ARCHICAD

```
ac_request("set_palette_size_and_message_place",int start_x, int start_y, int width, int height, int
messageplace_start_x, int messageplace_y, int messageplace_width, int messageplace_height);
:
start_x, int start_y - ;
width, height - ;
messageplace_start_x, int messageplace_y -
;
messageplace_width, messageplace_height - .
```

## Списки для выборки элементов из проекта

```
"load_elements_list".
```

```
, , , ,
```

```
. .
```

```
,
```

```
.
```

```
10,
```

```
0 9.
```

```
,
```

```
"set_TSMAX_ELEMLIST_NUMBER"  
  
"get_TSMAX_ELEMLIST_NUMBER"
```

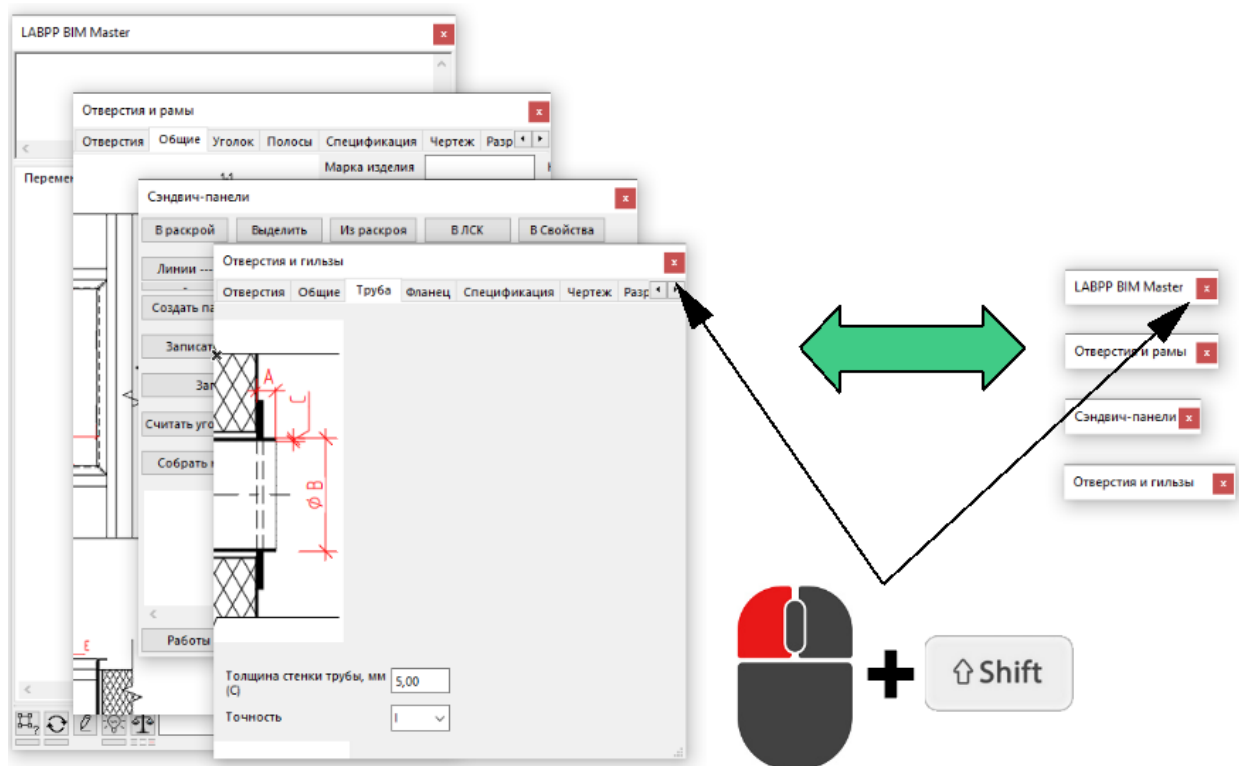
Генезис свойств

"ANY"	0	( . . )
"GDL "	1	GDL
"EP"	2	
"UP"	3	
"UPb"	4	
"UPub"	5	
"UPa"	6	
"Cls"	7	
"IFC"	8	IFC
"UTS"	9	LABPP
"Q"	10	Quantity -
"AP"	11	
"MV"	12	
"MV_GDL "	13	GDL
"PInfo"	14	

Окна LABPP

LABPP    ARCHICAD

ARCHICAD  
SHIFT



## Семантика языка C++ для ARCHICAD

### Общая структура программы

```
main(),
```

```
int my_global_var1 = 1; //
```

```
double my_global_var2; //
```

```
int main()
{
    int my_local_var = 2; // объявление переменной типа целое число
    my_local_var = my_func(); // вызов функции my_func() с присвоением результата
    // объявление собственной функции, возвращающей целое число
    int my_func()
    {
        int my_local_var = 3; // объявление переменной типа целое число внутри функции
        return 10; // значение, которое вернет функция
    }
}
```

### Функция main()

```
main().
C++
```

```
int main()
```

```
{
    // тело функции заключено в фигурные скобки
    return 0;
}
```

```
(run_cpp("get_args",...))
    (var_extern_get).
```

## Идентификаторы

```
int iValue    int IVALUE
```

## Типы переменных, видимость и виды

	<pre>int function() {     int i;     double d;     string s; }</pre>	
	<pre>int i; double d; string s;  int function() { }</pre>	cpp
<p>9) ( - 0</p>	<p>Собрать элементы зон во внутренний список элементов labrr № 7 и в цикле перебрать их для получения номера помещения и отображения их в окне сообщений.</p> <pre>ac_request("load_elements_list_from_selection", 7,            "ZoneType", "MainFilter", 2+1024            , "Classif",            "Классификация ARCHICAD, "ClassifVal",            "Помещение"); ac_request("get_loaded_elements_list_count", 7); int icount = ac_getnumvalue();  ac_request("get_loaded_elements_list_count", 7); icount = ac_getnumvalue(); cout &lt;&lt; "icount=" &lt;&lt; icount &lt;&lt; "\n"; if (icount == 0) {     cout &lt;&lt; "Нет зон в проекте\n";     return -1; }  int i; int ires; string szonename; for (i = 0; i &lt; icount; i++) {</pre>	cpp

	<pre>         ires = ac_request("set_current_element_from_list",         7, i);         ires = ac_request("get_element_value", "ZoneName");         szonename = ac_getstrvalue();         coutvar &lt;&lt; szonename;     } </pre>	
	<pre> int iTable; object("create","ts_table", iTable); object("KeepInMemory", iTable); </pre>	<pre> . cpp . (iTable) . </pre>
	<pre> var_extern_set("sWorkBookUpFile", "c:\ \filename.txt"); string sWorkBookUpFile; int ires = var_extern_get("sWorkBookUpFile", sWorkBookUpFile, 0); if (ires != 0) {     cout &lt;&lt; "Не задана внешняя переменная sWorkBookUpFile";     return -1; } </pre>	<pre> . cpp . </pre>

bool - ( 0- /1- ).  
int - .  
double - ( ).  
string - .

```

string mystring;
mystring = "fragment1" + "fragment2";
mystring += "fragment3";

```

## Пакетное объявление и обращение к переменным через идентификаторы, задаваемые текстом - var\_by\_txt()

```

,
.
"var_1_1_1" "var_100_100_100".
n- .
.
:
```

s = var\_67\_45\_34;

var\_34\_23\_23 = " 34\_23\_23";

```

for (i = 1; i <= 100; i++)
{

```

```

for (j = 1; j <= 100; j++)
{
    for (k = 1; k <= 100; k++)
    {
        var_by_txt("set", "var_"+itoa(i)+"_"+itoa(j)+"_"+ itoa(k),
                    "значение для переменной " + itoa(i) +
                    "_" + itoa(j) + "_" + itoa(k));
    }
}

```

## init

```

:
var_by_txt("init", string string_with_varnames, string vartype, string local_or_global,
void preset_value);

```

Здесь:

string\_with\_varnames - строка, содержащая один или список идентификаторов переменных через запятую ("var1" или "var1,var2,var3").

vartype - тип объявляемой переменной - "string", "int", "double", "bool".

local\_or\_global - в каком уровне видимости создавать переменные - "global" или "local".

preset\_value - значение, присваиваемое переменным по умолчанию.

Глобальные переменные можно создавать в цикле.

Локальные переменные в цикле или в других случаях с применением фигурных скобок, создавать нельзя, т.к. после выхода из понижающей области {} они удаляются.

Это как бы мы написали:

```

if(...)
{
    // объявляем локальную переменную k в пониженной области видимости
    int k = 1;
}
// здесь, за пределами фигурных скобок уже локальная переменная k не существует.

```

Чтобы выполнить массовое объявление локальных переменных - сначала сформируйте строку с идентификаторами через запятую, а затем выполните функцию var\_by\_txt("init"...).

Повторное объявление глобальных переменных может привести к ошибке.

Чтобы избежать этого используйте var\_by\_txt("is\_exist",...).

```

//*****
// Создание переменных с идентификаторами из текстовой строки
// функция var_by_txt()
// LABPP 2021
//*****
int main()
{
    test_global_variables();
    test_local_variables();
}

// Сформировать имя для переменной
string make_varname(string base, int ii, int jj)
{
    return base + "_" + itoa(ii) + "_" + itoa(jj)
}

int test_global_variables()

```



```

{
    int i,j;
    string basename = "glob_var";

    cout << "-----\n";
    cout << "test_global_variables\n";

    // если ранее не создана переменная с именем "glob_var_0_0", то создать все
    переменные от "glob_var_0_0" до "glob_var_9_9"
    if (var_by_txt("is_exist",make_varname(basename,0,0))==false)
    {
        for (i = 0; i < 10; i++)
        {
            for (j = 0; j < 10; j++)
            {
                cout << "объявляем переменную " << make_varname(basename,i,j)
<< "\n";
                var_by_txt("init", make_varname(basename, i, j), "string",
"global", "glob value " + itoa(i) + "_" + itoa(j));
            }
        }

        string s = glob_var_7_7;
        coutvar << s;
        glob_var_7_7 = "new value for 7_7";

        cout << "Все значения созданных глобальных переменных:\n";
        string varname;
        for (i = 0; i < 10; i++)
        {
            for (j = 0; j < 10; j++)
            {
                varname = make_varname(basename, i, j);
                cout << varname + "=" << var_by_txt("get", varname) << "\n";
            }
        }
        return 0;
    }
    // тест для локальных переменных
    // в отличии от глобальных - локальные переменные должны создаваться в теле функции не
    // глубже других фигурных скобок
    // { }
    // иначе это понижает область видимости.
    // Поэтому для локальных переменных сначала формируем строку с именами переменных через
    // запятую.
    // Такой прием возможен и для глобальных переменных.
    int test_local_variables()
    {
        int i, j;
        string basename = "local_var";
        cout << "-----\n";
        cout << "test_local_variables\n";
        string string_with_varnames;

        cout << "формируем список имен переменных в строку через запятую\n";
        for (i = 0; i < 10; i++)
        {
            for (j = 0; j < 10; j++)
            {
                string_with_varnames += make_varname(basename, i, j) + ",";
            }
        }
    }
}

```

```

coutvar << string_with_varnames;

// создаем переменные от "local_var_0_0" до "local_var_9_9", иницилируем пустой
строкой (можно другим значением)
var_by_txt("init", string_with_varnames, "string", "local", "");

// присвоение значений
for (i = 0; i < 10; i++)
{
    for (j = 0; j < 10; j++)
    {
        var_by_txt("set", make_varname(basename, i, j), "local value " +
itoa(i) + "_" + itoa(j));
    }
}

// прямое обращение к переменной
string s = local_var_7_7;
coutvar << s;
local_var_7_7 = "new value for 7_7";

cout << "Все значения созданных локальных переменных:\n";
string varname;
for (i = 0; i < 10; i++)
{
    for (j = 0; j < 10; j++)
    {
        varname = make_varname(basename, i, j);
        cout << varname + "=" << var_by_txt("get", varname) << "\n";
    }
}
return 0;
}

```

## is\_exist

```

bool var_by_txt("is_exist", string varname);
varname -

```

## set

```

:
var_by_txt("set", string varname, void varvalue);
:
varname - ("var_0_10_343_34").
varvalue -

```

## get

```

:
string s = var_by_txt("get", string varname);
double d = var_by_txt("get", string varname);
int i = var_by_txt("get", string varname);
bool b = var_by_txt("get", string varname);

varname - ("var_320_ini" . .).
:
s = var_320_ini;

```

## Комментарии

```

//
:
int i = 0; // здесь мы объявляем переменную - целое число и присваиваем ему значение 0

/* */
/* */

/*Если нужно написать длинное пояснение на нескольких строках в тексте программы
или закомментировать фрагмент программы,
то удобно сделать так*/

```

## Организация циклов

### Цикл for

```

C++.

break;

for(...)

"
!" 10

int i;
for (i = 0; i < 10; i++)
{
    cout << "Здравствуй Мир!";
}

```

### Цикл do-while

```

int i;
i = 0;
do {
    cout << i; // Вывести в окно сообщений LabPP_Automat значение i
    i++; // Увеличить i на 1
} while (i < 10) // Выполнять цикл покуда i меньше 10

```

### Цикл while

```

int i;
i = 0;
while (i < 10) {
    cout << i;
    i++;
}

```

## Логические операции

```

==
!=
<
<=
>
>=
&&

||

```

## Условные переходы

### Оператор if

```
if (i < 0)
{
    cout << "да";
}
```

### Конструкция if-else

```
if (i < 0) // Если выражение в скобках истинно, то выполняется первый фрагмент
{
    cout << "да";
}
else      // Если выражение в скобках ложно, то выполняется второй фрагмент
{
    cout << "нет";
}
```

### Конструкция if-else if-else

```
if (i < 0) // Если выражение в скобках истинно, то выполняется первый фрагмент
{
    cout << "i < 0";
}
else if (i == 0) // Если выражение в скобках истинно, то выполняется второй фрагмент
{
    cout << "i = 0";
}
else // Если ни одно из выражений не истинно (пунктов else if может быть много)
{
    cout << "i > 0";
}
```

### Оператор switch

```
switch (i)
{
    case 1:
        cout << "i=1";
        break;
    case 2:
        cout << "i=2";
        break;
    case 10:
        cout << "i=10";
        break;
    default:
        // операторы, выполняемые при любом другом значении i
}
```

## Создание собственных функций (подпрограмм)

main()

```

int main()
{
    cout << "Строка из функции с арг. 1= " << get_string(1) << ", то же но с арг. 2 = " <<
get_string(2);
}
// объявление функции, принимает аргумент arg - целое число.
// возвращает строковое значение.
string get_string(int arg)
{
    string sresult;
    if (arg == 1) // если arg равно 1
    {
        sresult = "строка 1"; // то присвоить переменной sresult это значение.
    }
    else if (arg == 2) // если же arg равно 2
    {
        sresult = "строка 2";
    }
    else // если же ни 1 ни 2, то при любом другом значении
    {
        sresult = "строка N"; // присвоить переменной sresult это значение.
    }
    return sresult; // вернуть в качестве значения функции содержимое переменной sresult.
}

```

( ).

```

double my_func(int a, int b, double c)
{
    return 10;
}

```

++

## Директива #include

C++

.h .hpp.

```
#include "my_heading.hpp"
```

my\_heading.hpp.

main()

cpp

```
#include "../my_heading.hpp"
```

## Особая конструкция сpp для ARCHICAD - управляемый #include

```
#include
#include <PInfo Имя пользовательской переменной в информации о проекте>
или
#include <PInfo Имя пользовательской переменной в информации о проекте|альтернативное
имя файла.ext?>
```

```

    :
PInfo - , #include (PInfo -
    ).

```

```
#include.
```

```
.ext =
```

## Директива #pragma region

```
#pragma region Создание диалога
// здесь код создания диалога, который может сворачиваться редактором текста
```

```
#pragma endregion
```

## Поиск по шаблону (коды REGEX)

The diagram illustrates the components of a regular expression (REGEX). It shows a sequence of characters: 'REGEX', 'ID', and various symbols like '(', ')', '[', ']', '^', '\$', and backslashes. Some symbols are grouped together, and some are shown with their escaped versions (e.g., '\(', '\)'). The diagram illustrates how these symbols are used to define patterns in text.

# -

```
"(^(?![\\w\\W]))|(?:.*[а-я].*)|(?:.*[#].*)"
```

"1" "2" :  
"(?:.\*подстрока1.\*)|(?:.\*подстрока2.\*)"

2  
1, 250

...	...	...
...	...	...

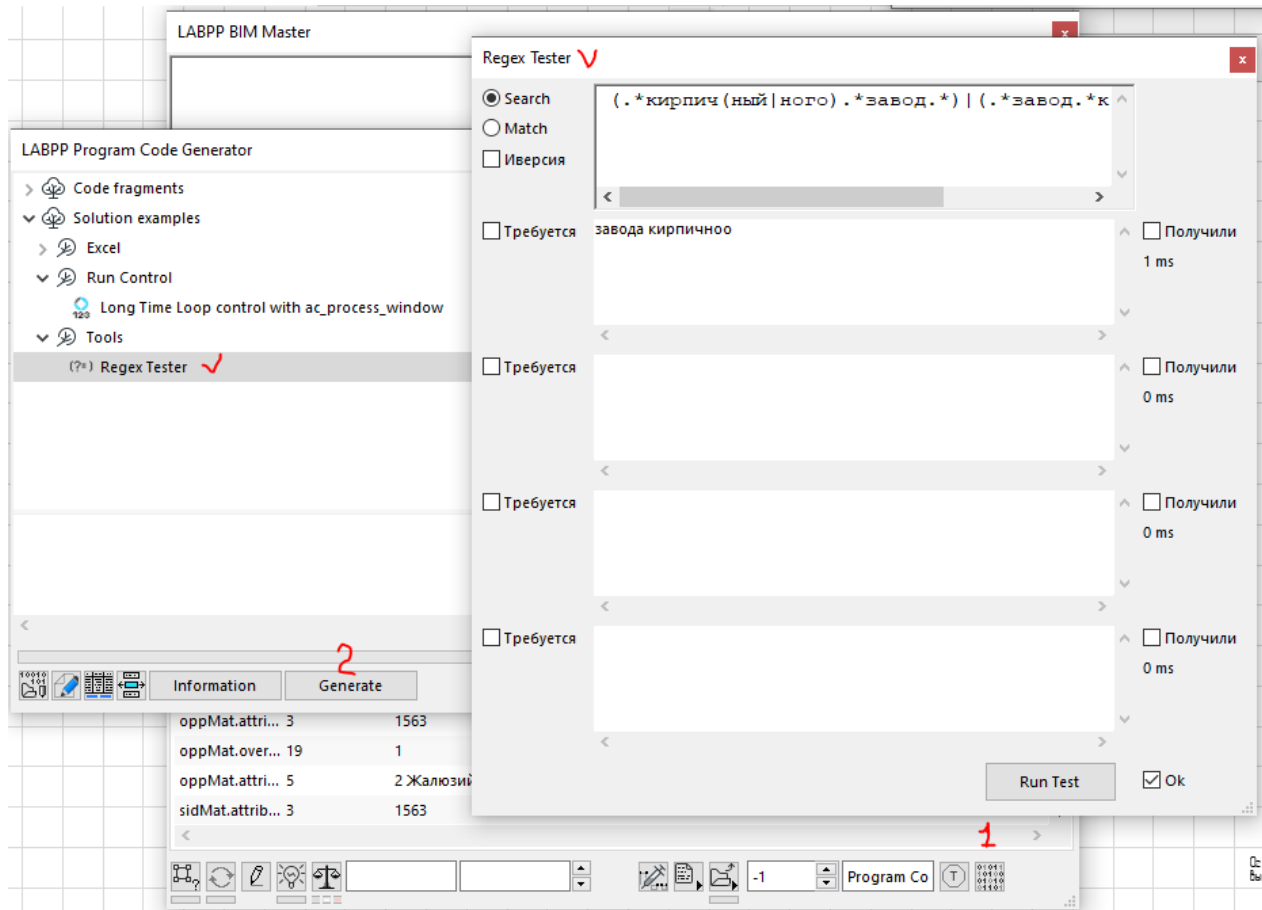
***	***	***
***	***	***

```
ac_request_special("load_elements_list", 1, "WallType", 2,
    "", "Cls", "Классификация АРХИКАД", "ВВЕТВИ", "Стена из кирпича", "", "И",
    "", "UPub", "Расположение", "=", "Внутри", "", "И",
    "", "EP", "buildingMaterialName_thickness(\"(?.*[Кк]ирпич.*керамич.*)|(?
:.*керамич.*[Кк]ирпич.*)\")", ">=", 0.250, "", "");
```

Для тестирования выражений используйте специальную панель, которая идет в комплекте с программами LABPP.

Она запускается через BIM Master.

Панель BIM Master вызывается через (Ctrl+Shift+F9).



## Логическое И и логическое ИЛИ в регулярных выражениях

Логическое И (AND) в регулярных выражениях обозначается символом `&`.  
 Логическое ИЛИ (OR) в регулярных выражениях обозначается символом `|`.  
 Например, регулярное выражение `(?=.* ( | ) .* .*)|(?=.* .* ( | ) .*)` означает: "найти строку, содержащую либо `( | )`, либо `.*`".

```

, - !
:
(?.* ( | ).* .*)(?=. * ( | ).*)(?!.* .*)

```

## Функции для работы с оболочкой shell\_func

### get\_path

```

:
int res = shell_func("get_path", string what, string result);

what :
"rootconfig" - , (tsprglist) LabPP_Automat;
"tsimages" - , LabPP_Automat;
"tsprg" - , LabPP_Automat;
"lib" - , LabPP_Automat;
"rootaddons" - , LabPP_Automat ,
;
"tmp" - , , SETCFG("TMPFolder",
string sTMPFolder);
"currentcpp" - , ;
"currentcppfiles" - "files".
"currentcppfilename" - (
);
"curdir" - ;
result

```

### set\_cur\_dir

```

:
int res = shell_func("set_cur_dir", string path_or_what);
path_or_what - ("c:\\my_dir\\") :
"rootconfig" - , (tsprglist) LabPP_Automat;
"tsimages" - , LabPP_Automat;
"tsprg" - , LabPP_Automat;
"lib" - , LabPP_Automat;
"tmp" - , , SETCFG("TMPFolder",
string sTMPFolder);
"rootaddons" - , LabPP_Automat ,
;
"currentcpp" - , ;
"currentcppfiles" - "files".

"C:\\MyDir":

string mydir = "C:\\MyDir";
int res = shell_func("set_cur_dir", mydir);
if (res == 0)
{
    cout << "Рабочий каталог успешно изменен";
}
else
{

```



```
    cout << "Установить каталог на " << mydir << " не удалось";
}
```

## shellexecute

Обращение.

```
int res = shell_func("shellexecute", "open", string spath);
```

Здесь:

spath - путь к программе или файлу, который надо открыть.

```
    ,
    .
    "
    .xls",
    LabPP_Automat.
```

```
int res = shell_func("set_cur_dir", "rootconfig");
if (res != 0)
{
    return -1;
}
res = shell_func("shellexecute", "open", "Пример.xls");
```

```
    .xls
    .xls
    EXCEL
    EXCEL ARCHICAD.
    EXCEL.
```

## ExecCmd

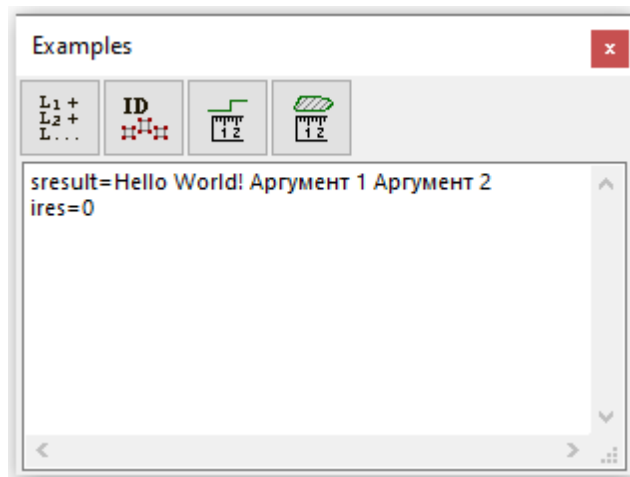
```
    :
    :
int ierr = shell_func("ExecCmd", string exenamewithargs, string sresult);
    :
exenamewithargs -
```

sresult -

```
    :
-1 -
0 -
```

```
    "
    2"
    "TSChild Process.exe"
    "
    1"
```

```
int main()
{
    string sresult;
    int ires = shell_func("ExecCmd", "\"C:\\source\\TSChild Process.exe\" \"Аргумент
1\" \"Аргумент 2\"", sresult);
    coutvar << sresult;
    coutvar << ires;
}
```



"TSChild Process.exe"

++

```

//*****
// Этот exe модуль запускается и пишет на консоли
//*****
#include <iostream>
using namespace std;
int main(int argc, char* argv[])
{
    if (argc == 2)
    {
        cout << "Hello World! Привет! " << argv[1] << "\n";
    }
    else if (argc == 3)
    {
        cout << "Hello World! Привет! " << argv[1] << " " << argv[2] << "\n";
    }
    else
    {
        cout << "Мало аргументов\n";
    }
    return 1;
}
//*****

```

## file

### copy

```

:
int err = shell_func("file", "copy", string spath_from, string spath_to);

```

```

:
spath_from - , ,
spath_to - , .

```

```

:
0 - ;
-2222 - ;
-2223 - ;
-2224 - ;
-2225 - .

```

-1 -

is\_exist

```
int ires = shell_func("file", "is_exist", string spath, bool bexist);

spath -
bexist - (1/0 - / ).
```

delete

```
int ires = shell_func("file", "delete", string spath);

spath -

string filename = "h:\\1.tmp";
ires = shell_func("file", "is_exist", filename, bexist);
if (bexist)
{
    cout << "Удаляем файл " << filename << "\n";
    ires = shell_func("file", "delete", filename);
}
```

get\_SLANG\_CODE

```
("RUS", "INT" . .)

shell_func("get_SLANG_CODE", sLANG);
```

"CHN"	
"DEU"	
"FRA"	
"INT"	
"ITA"	
"JPN"	
"KOR"	
"POL"	
"PRT"	
"RUS"	

## Управление окнами labpp

---

```

:
:
int ErrCode = labpp_windows_list_dialog(string sCommand);

:
sCommand -
"HideAllWindows" - LabPP;
"ShowAllWindows" - LabPP, BIM
Master' ;
"SwitchWindowsVisible" - ( , ).

```

## Буффер обмена Windows

---

### clipboard\_get

```

: Windows
:
ac_request_special("clipboard_get", string str);

```

### clipboard\_set

```
ac_request_special("clipboard_set", str);
```

## Функции автоматизации – Automate

---

### CloseCurWindow

```

:
int iret = ac_request("Automate",CloseCurWindow);
0

```

### Zoom

```

:
ac_request("Automate","Zoom",double minx, double miny, double maxx, double maxy);
:
minx, miny - ;
maxx, maxy - .

```

### GoToView

```

: ARCHICAD
:

```

```
int ires = ac_request("Automate","GoToView",int iViewLinkGuid);
:
iViewLinkGuid - ts_guid guid ,
.
0 .
```

## ChangeWindow

```
int ires = ac_request("automate","ChangeWindow", int iDatabaseDescr);
:
iDatabaseDescr - ac_database
0 .
```

## ShowSelectionIn3D

```
2d 3D. 3d
.
F5.
ac_request("Automate","ShowSelectionIn3D");
```

## ShowAllIn3D

```
3D.
F5 .
ac_request("Automate","ShowAllIn3D");
```

## ZoomToElements

```
(2d 3d).
ac_request("Automate","ZoomToElements", int iListNumber);
ac_request("Automate","ZoomToElements", int iListNumber, double doffset);
:
iListNumber -
doffset - , .
```

## Redraw

```
int res = ac_request("Automate","Redraw");
```

## Rebuild

```
int res = ac_request("Automate","Rebuild", bool doregenerate);
:
doregenarate - 0/1 - .
```

## Функции окружения - Environment

---

### AddLibraries

```
:
```

```
int ires = ac_request("Environment","AddLibraries",string LibPath);
```

Story\_GoTo

```
int ires = ac_request("Environment","Story_GoTo", int storyindex);
```

Функции для работы с базами проекта - database

GetCurrent

```
int iDatabaseDescr;
object("create", "ac_database", iDatabaseDescr);
int err = ac_request("database", "GetCurrent", iDatabaseDescr);
courvar << err;
```

Select

```
ARCHICAD
int iDatabaseDescr;
int err = ac_request("database", "Select", iDatabaseDescr);
iDatabaseDescr - 2D.
int iDatabase2d;
object("create", "ac_database", iDatabase2d);
ac_request("database", "SetType",iDatabase2d, 1);
ac_request("database", "Select", iDatabase2d);
```

SetTypeID

```
ac_request("database", "SetTypeID", int iDatabase, int itype);
iDatabase - 2d, 3d .);
itype -
```

0	ZombieWindow	

1	FloorPlan	2D
2	Section	
3	Detail	
4	3DModel	3D
5	Wind_Layout	layout window
6	Drawing	
7	MyText	custom text
8	MyDraw	custom draw
9	MasterLayout	master layout
10	Elevation	elevation window
11	InteriorElevation	interior elevation
12	Worksheet	worksheet
13	Report	report window
14	DocumentFrom3D	3D Document window
15	External3D	External 3D
16	Movie3D	movie 3D
17	MovieRendering	movie rendering
18	Rendering	rendering window
4096	IESCommonDrawing	Interactive Schedule window

## GetTypeID

```

:
:
ac_request("database", "GetTypeID", int iDatabase);

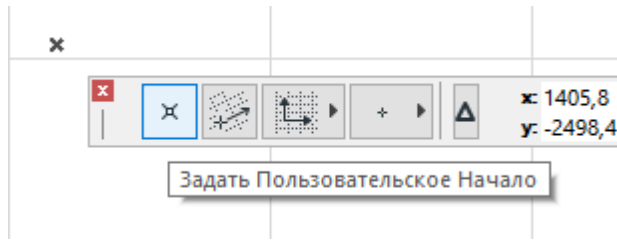
:
iDatabase -
itype - ( 2d, 3d );
ac_getnumvalue();

.

int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
ac_request("database", "GetType", iDatabase);
int itype = ac_getnumvalue();
coutvar << itype;

```

## GetLocOrigo



```

:
ac_request("database", "GetLocOrigo", int iDatabase, double& x_res, double& y_res,
double& z_res);

```

```

:
iDatabase - ;
x_res, y_res, z_res - ;

```

```

.

int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
double x_res, y_res, z_res;
ac_request("database", "GetLocOrigo", iDatabase, x_res, y_res, z_res);
coutvar << x_res;
coutvar << y_res;
coutvar << z_res;

```

## get\_section\_infoidtext

```

:
int ires = ac_request("database", "get_section_infoidtext", int iDatabase, string &
sectioninfoidtextres);

```

```

:
ires - 0;
iDatabase - ;
sectioninfoidtextres - ;

```

```

:
int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
string sectioninfoidtextres;
ac_request("database", "get_section_infoidtext", iDatabase, sectioninfoidtextres);
coutvar << sectioninfoidtextres;

```

## get\_sectionmarkers2dpositions

```

:
int ires = ac_request("database", "get_sectionmarkers2dpositions", iDatabase, double& sx,
double& sy, double& ex, double& ey, double& alph);

```

```

:
iDatabase - ;
sx,sy,ex,ey - 2d ;
alph - 2d .

```



```
double alph;
int ires = ac_request("database", "get_sectionmarkers2dpositions", idatabase, sx, sy, ex,
ey, alph);
```

## GetRVMChanges

```

:
int ires = ac_request("database","GetRVMChanges", int iDatabaseDescr, int iArrayDescr);

:
ires - 0;
iDatabaseDescr - ;
iArrayDescr - .

.
,
.

int iArrayDescr;
int iDatabaseDescr;

object("create","ts_array",iArrayDescr);
object("create","ac_database", iDatabaseDescr);
ac_request("database","GetCurrent",iDatabaseDescr);

ts_array(iArrayDescr,"init","ac_RVMChange");
ac_request("database","GetRVMChanges",iDatabaseDescr,iArrayDescr);

int iSize;
ts_array(iArrayDescr,"GetSize",iSize);
cout << "Вообще с листом связано изменений=" << iSize << "\n";

object("delete",iArrayDescr);
object("delete",iDatabaseDescr);
```

## GetRVMLayOutCurrentRevisionChanges

```

:
int ires = ac_request("database","GetRVMLayOutCurrentRevisionChanges", int
iDatabaseDescr, int iArrayDescr);

:
ires - 0;
iDatabaseDescr - ;
iArrayDescr - .

.
,
.

int iArrayDescr;
int iDatabaseDescr;

object("create","ts_array",iArrayDescr);
object("create","ac_database", iDatabaseDescr);
ac_request("database","GetCurrent",iDatabaseDescr);

ts_array(iArrayDescr,"init","ac_RVMChange");
ac_request("database","GetRVMLayOutCurrentRevisionChanges",iDatabaseDescr,iArrayDescr);

int iSize;
ts_array(iArrayDescr,"GetSize",iSize);
```

```
cout << "Число изменений, видимых на листе=" << iSize << "\n";
```

```
object("delete",iArrayDescr);
object("delete",iDatabaseDescr);
```

## GetTypeNames

```
int ires = ac_request("database","GetTypeNames", int iDatabaseDescr);
```

```
ires - 0;
iDatabaseDescr - ;
ac_getstrvalue();
```

```
string stypename;
int ires = ac_request("database","GetTypeNames", iDatabaseDescr);
if(ires == 0){
    stypename = ac_getstrvalue();
}
```

```
"ZombieWindow"
"FloorPlan";
"Wind_Section";
"Wind_Detail";
"Wind_3DModel";
"Wind_Layout";
"Wind_Drawing";
"Wind_MyText";
"Wind_MyDraw";
"Wind_MasterLayout";
"Wind_Elevation";
"Wind_InteriorElevation";
"Wind_Worksheet";
"Wind_DocumentFrom3D";
"Wind_IESCommonDrawing";
```

## GetSizeX, GetSizeY

```
int ires = ac_request("database","GetSizeX", int iDatabaseDescr);
int ires = ac_request("database","GetSizeY", int iDatabaseDescr);
```

```
ires - 0;
iDatabaseDescr - ;
ac_getnumvalue();
```

```
int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
ac_request("database","GetSizeX", iDatabaseDescr);
double sizeX = ac_getnumvalue();
ac_request("database","GetSizeY", iDatabaseDescr);
```

```
double sizeY = ac_getnumvalue();
object("delete", iDatabase);
```

## leftMargin,topMargin,rightMargin,bottomMargin

```

:
int ires = ac_request("database","leftMargin", int iDatabaseDescr);
int ires = ac_request("database","topMargin", int iDatabaseDescr);
int ires = ac_request("database","rightMargin", int iDatabaseDescr);
int ires = ac_request("database","bottomMargin", int iDatabaseDescr);
```

```

:
ires - 0;
iDatabaseDescr - ;
ac_getnumvalue();
```

```

int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
ac_request("database","leftMargin", iDatabaseDescr);
double leftmargin = ac_getnumvalue();
coutvar << leftmargin;
object("delete", iDatabase);
```

## layoutPageNumber

```

:
int ires = ac_request("database","layoutPageNumber", int iDatabaseDescr);
```

```

:
ires - 0;
iDatabaseDescr - ;
ac_getnumvalue();
```

```

int iDatabase;
object("create", "ac_database", iDatabase);
ac_request("database", "GetCurrent", iDatabase);
ac_request("database","layoutPageNumber", iDatabaseDescr);
double layoutPageNumber = ac_getnumvalue();
coutvar << layoutPageNumber;
object("delete", iDatabase);
```

## layoutName

```

:
int ires = ac_request("database","layoutName", int iDatabaseDescr);
```

```

:
ires - 0;
iDatabaseDescr - ;
ac_getstrvalue();
```

```

int iDatabase;
object("create", "ac_database", iDatabase);
```

```
ac_request("database", "GetCurrent", iDatabase);
ac_request("database", "layoutName", iDatabaseDescr);
string layoutName = ac_getstrvalue();
coutvar << layoutName;
object("delete", iDatabase);
```

## Вывод в окно сообщений

---

### cout

```
cout
:
:
cout << arg1 << arg2 << argN;
:
arg1, arg2, argN -
```

dsum

```
cout << "Площадь всех квартир = " << dsum;
```

### coutvar

```
:
coutvar << my_variable;
```

### setcfg("COUT",...)

```
:
setcfg("COUT", int onoff);
onoff - 0/1 /
0, cout coutvar
1.
```

## Файловые операции

---

### open

```
Int ires = ts_file(int iFileDescr, "open", string filepath, string what, string mode);
: iFileDescr - , what -
object("create"..., filepath - , "fail" - , "ignore" - ).
("create" - , "fail" - , "ignore" - ).
mode - :
"r" -
"w" -
"we" -
"rw" -
"a" -
ires = ts_file(iFileDescr, "open", filepath, "create", "we");
```

### write

```
int ires = ts_file(int iFileDescr, "write", string stowrite);
: iFileDescr - , stowrite - . 0,
```

```
ac_getnumvalue();
```

## read

```
int ires = ts_file(int iFileDescr, "read", string &storead);
    : iFileDescr - , storead -
    0,
    ac_getnumvalue().
```

## close

```
int ires = ts_file(Int iFileDescr, "close");
    : iFileDescr - 0

    "my_file.txt" "C:"

string filepath = "c:\\my_file.txt";
int iFileDescr;
object("create", "ts_file", iFileDescr); // создать объект типа файл в памяти
// открыть для записи чистый файл, если его нет, то создать
int ires = ts_file(iFileDescr, "open", filepath, "create", "we");
if (ires != 0)
{
    cout << "Файл не удалось открыть:" << filepath; // выдать в окно сообщений
    return;
}
ires = ts_file(iFileDescr, "write", "Первая строка\nВторая строка\n"); // записать в файл
две строки
if (ires != 0)
{
    cout << "Не удалось записать в файл";
    return;
}
ires = ts_file(iFileDescr, "write", "Третья строка"); // записать третью строку
ires = ts_file(iFileDescr, "close"); // закрыть файл
object("delete", iFileDescr); // удалить объект файла из памяти
cout << "Завершение программы \n";
```

## SetIsUNICODE

Вызов:

```
ts_file(iFileDescr, "SetIsUNICODE");
```

Команда указывает что файл содержит данные в кодировке UTF-8.

## SetIsUTF16

Вызов:

```
ts_file(iFileDescr, "SetIsUTF16");
```

Команда указывает что файл содержит данные в кодировке UTF-16.

Пример.

Считать таблицу из файла, созданного Excel командой "Сохранить текст в Unicode" (сохраняется таблица с разделителем - табуляция, в кодировке UTF-16).

```
int main()
{
    int iTable;
    object("create", "ts_table", iTable);
```

```

string sFullPath = "c:\\1.txt";
string csv_source = "";
int iFileDescr;
object("create", "ts_file", iFileDescr);
ts_file(iFileDescr, "SetIsUTF16");
int ires = ts_file(iFileDescr, "open", sFullPath, "ignore", "r");
if (ires == 0)
{
    ts_file(iFileDescr, "read", csv_source);
    ts_file(iFileDescr, "close");
    coutvar << csv_source;
    ts_table(iTable, "import_from_csv", csv_source, "\t");
    string str;
    ts_table(iTable, "print_to_str", str);
    coutvar << str;
}
else
{
    tsalert(-1, "Error", "Can't open file " + sFullPath, "Check availability of
the file", "Ok");
    return -1;
}

object("delete", iTable);
object("delete", iFileDescr);
}

```

## Строковые функции

---

### strcmp

```

:
bool result = strcmp(string s1, string s2);

:
string1 string2 -
1

```

### tolower

```

:
string sresult = tolower(string svalue);

```

### toupper

```

:
string sresult = toupper(string svalue);

```

### alltrim

```

:
string sresult = alltrim(string svalue);

```

## strcontains

```
int ires = strcontains(string sstring, string sfragment);
```

```

:
sstring - , sfragment. , ires 1.

```

## strreplace

```
string sresult = strreplace(string s, string fragmentold, string fragmentnew, int how, int pos, int count);
```

```

:
s - , fragmentold fragmentnew,
how - - 0/1/2 - / /
pos -
count - (-1 - ).

```

```

double d_value = 126.983;
string s_value = sprintf("%10.3f", d_value);
s_value = strreplace(s_value, ".", ",", 0, 0, -1);
Результат - 126,983

```

## strlen

```
int ilen = strlen(string s);
```

```

:
s -

```

## substr

```
string substr(string source, int startpos, int count);
```

```

:
source - ,
startpos - ,
count - ( -1, ).

```

## strposcharset

```

:
0, . . 0. - -1

```

```
int strposcharset(string sourcestr, int ipos, char chartoplace, string &strresult);
```

```

:
sourcestr - ;

```

```

ipos - , ;
chartoplace - , .
strresult - .

:
2 "ABCDE" ' ' 4 .

```

```

char ch;
string st, strresult;
st = "ABCDEF";
int iret = strposcharget(st, 2, ch);
coutvar << ch ;
ch = 'W';
iret = strposcharset(st, 1, ch, strresult);
coutvar << strresult;

```

:

B  
ABCD\_F

## strposcharget

```

:
- -1
:
int strposcharget(string sourcestr, int ipos, char &charres);

```

## strleft

```

:
(
):
string strleft(string str, string substr, bool isfromlast=false);

```

Здесь:  
str - исходная строка,  
substr - фрагмент-разделитель.

isfromlast true -

## strright

```

:
string strright(string str, string substr, bool isfromlast=false);

```

Здесь:  
str - исходная строка,  
substr - фрагмент-разделитель.

isfromlast true -

## str\_regex\_search

```

:
str_regex_match(), (?=text) (!text).
:
:

```



```
(?=...)(?=.*....*)(?=.*....*)(?!.*....*)
```

Digital Service	18-24	25-34	35-44	45-54	55-64	65-74	75+
Email	95%	90%	85%	80%	75%	70%	65%
Social media	90%	85%	80%	75%	70%	65%	60%
Video streaming	85%	80%	75%	70%	65%	60%	55%
E-commerce	80%	75%	70%	65%	60%	55%	50%
Mobile banking	75%	70%	65%	60%	55%	50%	45%

" 22",  
( regex ).

```
bool str_regex_search(string source, string code);
```

```

:
source - ,
code - .

```

## str\_regex\_match

" 22",  
( regex ).

```
bool str_regex_match(string source, string code);
```

```

:
source - ,
code - .

```

```
string code = ".*@.*\\.ru";
//
```

## str\_regex\_replace

```

        :
        regex (
        ).
string sResult = str_regex_replace(string sCodedValue, string sOriginalValue, string
sReplaceFragment);

```

```

:
sCodedValue - (regex, "abc[12]"),
sOriginalValue - ,
sReplaceFragment - ,
sResult - .

```

sOriginalValue

"123 abc 456" 12 6 #.

```
string sres = str_regex_replace("(?:12)|(?:[6])", "123 abc 456", string "#");
coutvar << sres;
```

```
sres = "#3 abc 45#"
```

```
"10001m" "2200k"
```

```
ac_request_special("get_element_value", "UP", "Технические переменные для отверстий для
LabPP\\Маркировка отверстия");
svalue = ac_getstrvalue();
svalue = str_regex_replace("[a-z]", svalue, "#");
svalue = strleft(svalue, "#");
coutvar << svalue;
```

Результат:  
10001 2200 и т.п. в зависимости от значения свойства.

## str\_regex\_substr

```
bool str_regex_substr(string source, string code, string usSubstrRes, string usPrefix,
string usSuffix);
```

```
source -
;
code -
;
usSubstrRes -
;
usPrefix -
;
usSuffix -
;
true -
```

```
string s = "Дверной блок из алюминиевых профилей (ВхН): 1400x2100 мм"; // исходный текст
string usSubstrRes, usPrefix, usSuffix;
string code = "\\(ВхН\\)\\: \\d{1,}x.*\\d{1,}"; // шаблон означает: текст, у которого
(ВхН): "одна или несколько цифр" x "одна или несколько цифр"
bool b = str_regex_substr(s, code, usSubstrRes, usPrefix, usSuffix);
if (b)
{
coutvar << usSubstrRes; // в окно сообщений будет выведено "(ВхН): 1400x2100 мм"
code = "\\d{1,}"; // шаблон означает: текст, у которого "одна или несколько цифр"
b = str_regex_substr(usSubstrRes, code, usSubstrRes, usPrefix, usSuffix);
coutvar << usSubstrRes; // в окно сообщений будет выведено "1400"
if (b)
{
code = "\\d{1,}"; // шаблон означает: текст, у которого "одна или несколько цифр"
// для поиска второго числа используем суффикс от прошлого поиска
b = str_regex_substr(usSuffix, code, usSubstrRes, usPrefix, usSuffix);
```

```
coutvar << usSubstrRes; // будет выведено "2100"
}
```

## straddlashtoregexcodes

```

        '!', '\\', '[', '\\[', '(', '\\(' . . .
    :
string straddslashtoregexcodes(string source);

```

## strremovebaddelimiters

```

:
string strremovebaddelimiters(string originalstr, string delimiter);
:
originalstr -
delimiter -
(
"???!").

" "
,

int main()
{
    string s = ", text,, еще текст ,, , и еще вот текст, ";
    string s2 = strremovebaddelimiters(s,",");
    coutvar << s2;
    // результат: "text еще текст и еще вот текст"
}

```

## str2threekeys

```
int str2threekeys(string sSource, double& k1, double &k2, double &k3);
```

Здесь:  
sSource - исходная строка.  
k1, k2, k3 - значения, которые будут выведены в консоль.

## strCompareLikeABC10\_a

```
-1/0/1          s1 < s2 / s1=s2/ s1 > s2.
```

```
int strCompareLikeABC10 a(string s1, string s2);
```

Здесь:  
s1 и s2 - сравниваемые строки.

Пример:

Сравнить выше или ниже по списку должно размещаться помещение с именем "ПОМЕЩЕНИЕ 102.2b" относительно помещения с именем "Помещение 102.3.b"

```
string s1 = "ПОМЕЩЕНИЕ 102.2b";
string s2 = "Помещение 102.3.b";

int ires = strCompareLikeABC10_a(s1, s2);

coutvar << ires;
```

Результат - в поле сообщений будет выведено значение -1. Это означает, что помещение 102.2b в списке должно располагаться раньше чем помещение 102.3.b.

## Руководство по LABPP\_Automat для ARCHICAD

```
escape-                "\u0420\u0435"
:
string strreplace_escape_u(string sSource);
escape-                .
```

## strisnumber

```

,                1,                0.

bool result = strisnumber(string str);
```

```

:
str -                .
```

## smallnums

(unicode).

```

:
string sresult = smallnums(string str);
```

```

:
str -                .
```

```

                " 2".
```

```
string str = "Объем, м" + smallnums("2");
```

## ts\_text\_mc

```

:
string sresult = ts_text_mc(int itextcode);
```

```

:
itextcode -                .
```

1	Запуск
2	Старт
3	Отменить
4	Продолжить

5	Ошибка
6	Информация
7	Предупреждение
8	Проверка
9	Выполнить все
10	Ок
11	Создать
12	Открыть
13	Закрыть
14	Считать
15	Записать
16	Да
17	Нет
18	Помощь

"Запуск"

```
string str = ts_text_mc(1);
```

## Объект ac\_element\_guid (ts\_guid)

---

## Объект ac\_element

---

## ac\_request

---

### get\_floor\_name\_by\_floor\_index

```

:
int ires = ac_request("get_floor_index_by_floor_name", string sStoryName);

ac_getnumvalue();
ires -1
0.
:
"2- "
```

```

int ifloorindex;
int ires = ac_request("get_floor_index_by_floor_name", "2-й этаж");
if(ires == 0)
{
    ifloorindex = ac_getnumvalue();
    coutvar << ifloorindex;
}

```

### get\_floor\_index\_by\_floor\_name

```

:
int ires = ac_request("get_floor_name_by_floor_index", int iStoryIndex);

                                ac_getstrvalue();
                                ires -1
                                0.
:
                                1

```

```

string sStoryName;
int ires = ac_request("get_floor_name_by_floor_index", "2-й этаж");
if(ires == 0)
{
    sStoryName = ac_getstrvalue();
    coutvar << sStoryName;
}

```

### store\_cur\_element\_to\_descr

```

                                ac_element_guid.
                                ac_request(...)
                                .
                                ac_element_guid.
                                -
                                ac_element_guid
:
ac_request("store_cur_element_to_descr", int &iGuidDescr);

iGuidDescr -
                                ac_element_guid,

```

### set\_current\_element\_from\_descr

```

                                ac_reauest(...)
                                ac_element_guid.
:
ac_request("set_current_element_from_descr", int &iGuidDescr);

iGuidDescr -
                                ac_element_guid,
                                ac_request(...)

```

### set\_element\_by\_guidstr\_as\_current

```

                                ac_reauest(...)
                                GUID
:
ac_request("set_element_by_guidstr_as_current", string sGUID);

sGUID -
                                ,
                                guid

```

### get\_guid\_from\_element

```

                                ARCHICAD
                                ac_element
ac_element_guid.
:
Формат запроса:
ac_request("get_guid_from_element",int iElementDescr, int iGuidDescr);

:
iElementDescr -
                                ac_element,
                                ac_element_guid,
                                iGuidDescr.

```

## load\_element\_from\_guid

```

        ac_element          ARCHICAD,          (guid),
        ac_element_guid.
    ,
    ac_element.
:
ac_request("load_element_from_guid",int iElementDescr, int iGuidDescr);

:
iElementDescr -          ac_element,          ,
        ac_element_guid,          iGuidDescr.

```

## load\_elements\_list

```

        ARCHICAD
:
ac_request("load_elements_list",int iListNum,string sElemTypeName,"MainFilter",int
iMainFilterValue,string filterparametrname, string/double filterparametervalue,...);

:
iListNum -          ( 0...9).
sElemTypeName -          ,          - "ZombieElemType",
        , "ColumnType",          - "WallType" . .
"MainFilter" -          ,          . . iMainFilterValue -
        - 3.
        "ID"," ID" -          ,          ID=" ID" . .
        "MainFilter"          ,          "MainFilter",0.
. .
.
MainFilterValue=j1+2*j2+4*j3+8*j4+16*j5+32*j6+64*j7+128*j8+256*j9+512*j10+1024*j11+2048*j12+4096*j13
+268435456*j14,
        j          0 1.
j1:          .
j2:          .
j3:          .
j4:          3d
j5:          .
j6:          .
j7:          .
j8:          .
j9:          ,          variationID
j10:          .
j11:          .
j12:          ;          ,
j13:          ;
j14:          2d

```

## set\_current\_element\_from\_list

```

:
int ires = ac_request("set_current_element_from_list", int listindex, int
elemindexinlist);

```

Здесь:

`listindex` - номер списка элементов, из которого надо сделать элемент текущим для выполнения операций над ним.

`elemindexinlist` - индекс элемента в этом списке.

`ires` - при успешном выполнении равно 0.

## add\_elements\_list

`load_element_list,`

## load\_elements\_list\_from\_selection и add\_elements\_list\_from\_selection

`, load_elements_list,`  
ARCHICAD.

## load\_elements\_list\_curdb

`, load_elements_list,`

## clear\_list

`ac_request("clear_list",int iListNum);`

`iListNum` -

## store\_current\_element\_to\_list

`ac_request("store_current_element_to_list",int listnum, int index);`

Здесь:

`listnum` - номер списка, в который добавить текущий элемент.

`index = -1`

0.

## get\_loaded\_elements\_list\_count

`ac_request("get_loaded_elements_list_count", int iListNum);`

`iListNum` -

`ac_getnumvalue();`

## select\_elements\_from\_list

ARCHICAD

`ac_request("select_elements_from_list", int iListNum, bool bClear);`

`- iListNum` -

`bClear` -



## Слой - layer

### create

ARCHICAD.

```
int ires = ac_request("layer","create", string slayername);
```

Здесь:

slayername - имя слоя, который необходимо создать.

Если слой успешно создан в ires возвращается 0.

Если произошла ошибка, то возвращается отрицательное значение.

### get\_index

ARCHICAD.

```
int ires = ac_request("layer","get_index", string slayername);
```

```
slayername - "01 10 Несущие стены",
ires 0,
ac_getnumvalue();
"01 10", -
```

```
string slayername = "01 10 Несущие стены";
int ires = ac_request("layer","get_index",slayername);
if(ires < 0)
{
    // layer not exist - create it
    ires = ac_request("layer","create",slayername);
    if(ires == 0)
    {
        ires = ac_request("layer","get_index",slayername);
    }
    else
    {
        cout << "cann't create layer "<<slayername;
        return -1;
    }
}
```

```
int iLayerIndex = ac_getnumvalue();
cout << "Index of the layer=" << iLayerIndex << "\n";
```

### set\_layer\_visible

```
ac_request("layer", string whatdo);
whatdo = "ON"/"OFF"/"SWITCH", (
).
```

### get\_element\_overall\_dimensions

```

:
ac_request("get_element_overall_dimensions", double &lx, double &ly, double &lz);

: lx,ly,sz -

```

## get\_quantity\_value

```

:
int iret = ac_request("get_quantity_value", string svaluenam);

:
svaluenam -
iret - 0,

ac_getnumvalue();

1 ID="s ".

ac_request("load_elements_list",1,"ZombieElemType","ID","s участка","MainFilter",3);
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue();
cout << "Число выбранных элементов=" << icount;
int i; // объявляем переменную цикла.
double value; // для текущих значений
double summa=0; // для суммарной площади
for(i=0;i<icount;i++)
{
    ac_request("set_current_element_from_list", 1, i);
    ac_request("get_quantity_value", "Surface");
    value = ac_getnumvalue();
    cout << "элемент № " << i << ", площадь=" << value << "\n";
    summa += value;
}
cout << "суммарная площадь=" << summa;

```

### - WallType:

```

volume -
length -
volume ond -
volumeaskin -
volumebskin -
volumeaskincond -
volumebskincond -
surfacereflineside -
surfacereflineoppside -
surfaceofedge -
surfacereflinesidecond -
surfacereflineoppsidecond -
surfacewindows -
surfacedoors -
surfacetmptyholes -
columnsvolume -
columnsnnumber -
widthofwindows -

```

widthofdoors - ;  
 minheight - ;  
 maxheight - ;  
 minheightaskin - ;  
 maxheightaskin - ;  
 minheightbskin - ;  
 maxheightbskin - ;  
 centerlength - ;  
 area - ;  
 petimeter - ;  
 grossvolume - ;  
 grosssurfacereflineside - ;  
 grosssurfacereflinoopside - ;  
 emptyholesvolume - ;  
 emptyholessurfreflineside - ;  
 emptyholessurfreflineopside - ;  
 lengthonreflineside - ;  
 lengthonreflineopside - ;  
 lengthonreflinesidecond - ;  
 lengthonreflineopsidecond - ;  
 insulationskinthickness - ;  
 wallairskinthickness - ;  
 skinreflinethickness - ;  
 skinreflineoppthickness - ;  
 reflinlength - .

### **- ColumnType:**

surface - ;  
 coresurface - ;  
 venesurface - ;  
 volume - ;  
 veneervolume - ;  
 minheight - ;  
 maxheight - ;  
 perimeter - ;  
 area - ;  
 grosssurfaceofcore - ;  
 grosssurfaceofveneer - ;  
 coregrossvolume - ;  
 veneergrossvolume - ;  
 coretopsurface - ;  
 corebottomsurface - ;  
 veneertopsurface - ;  
 veneergrosssurface - ;  
 coregrosstopandbotsurface - ;  
 veneergrosstopandbotsurface - .

### **- BeamType:**

rightlength - ;  
 leftlength - ;  
 length - ;  
 bottomsurface - ;  
 topsurface - ;  
 edgesurfaceleft - ;  
 edgesurfaceright - ;  
 edgesurface - ;  
 holessurface - ;  
 holedgesurface - ;

holesnumber - ;  
 volume - ;  
 condvolume - ;  
 holesvolume - ;

### - **WindowType:**

openwidthrevside - ;  
 openwidthrevsideopp - , ;  
 openheightrevside - ;  
 openheightrevsideopp - , ;  
 opensurfacerevside - ;  
 opensurfacerevsideopp - , ;  
 nominalopenwidthrevside - ;  
 nominalopenwidthrevsideopp - , ;  
 ;  
 nominalopenheightrevside - - ;  
 nominalopenheightrevsideopp - , ;  
 ;  
 nominalopensurfacerevside - ;  
 nominalopensurfacerevsideopp - , ;  
 ;  
 volume - ;  
 nominalopensurgace - ;  
 nominalopenvolume - ;  
 surface - ;  
 nominalsillheight - ;  
 nominalsillheightrevside - ;  
 nominalsillheightrevsideopp - , ;  
 nominalheadheight - ;  
 nominalheadheightrevside ;  
 nominalheadheightrevsideopp - , ;  
 ;  
 sillheightaccvertanchor - ;  
 headheightaccvertanchor - .

### - **DoorType:**

openwidthrevside - - ;  
 openwidthrevsideopp - , ;  
 openheightrevside - ;  
 openheightrevsideopp - , ;  
 opensurfacerevside - ;  
 opensurfacerevsideopp - , ;  
 nominalopenwidthrevside - ;  
 nominalopenwidthrevsideopp - , ;  
 ;  
 nominalopenheightrevside - ;  
 nominalopenheightrevsideopp - , ;  
 ;  
 nominalopensurfacerevside - ;  
 nominalopensurfacerevsideopp - - , ;  
 ;  
 volume - ;  
 nominalopensurgace - ;  
 nominalopenvolume - ;  
 surface - ;  
 nominalsillheight - ;  
 nominalsillheightrevside - ;  
 nominalsillheightrevsideopp - , ;  
 nominalheadheight - ;

nominalheadheightreveside - ;  
 nominalheadheightrevesideopp - ,  
 ;  
 sillheightaccvertanchor - ;  
 headheightaccvertanchor - .

### - **ObjectType:**

volume - ;  
 surface - .

### - **LampType:**

volume - ;  
 surface - .

### - **SlabType:**

bottomsurface - ;  
 surface - ;  
 edgesurface - ;  
 condbottomsurface - ;  
 condtopsurface - ;  
 volume - ;  
 condvolume - ;  
 perimeter - ;  
 holessurface - ;  
 holesperimeter - ;  
 grossbottomsurface - ;  
 grosstopsurface - ;  
 grossedgessurface - ;  
 grossvolume - ;  
 grossbottomsurfacewithholes - ;  
 grosstopsurfacewithholes - ;  
 grossedgessurfacewithholes - ;  
 grossvolumewithholes - .

### - **RoofType:**

bottomsurface - ;  
 topsurface - ;  
 edgesurface - ;  
 condbottomsurface - ;  
 condtopsurface - ;  
 volume - ;  
 ondvolume - ;  
 perimeter - ;  
 holessurface - ;  
 holesperimeter - ;  
 grossbottomsurface - ;  
 grosstopsurface - ;  
 grossedgesurface - ;  
 areacontourpoligon - ;  
 grossvolume - ;  
 insskinthickness - ;  
 lengthofridgeedgesdiv2 - , 2;  
 lengthofvalleyedgesdiv2 - , 2;  
 lengthofgableedges - ;  
 lengthofhipedgesdiv2 - , 2;  
 lengthofeaveedges - ;  
 lengthofpeakedges - ;  
 lengthofsidewalldges - ;

```

lengthofendwalledges - ;
lengthofrtdomeedgesdiv2 - rtdom, 2;
lengthofrthollowedgesdiv2 - rthollow, 2;
sumofopeningssurfaces - ;
numofholes - ;
numofskylight - (skylight).

```

### **3d - MeshType:**

```

bottomsurface - ;
topsurface - ;
edgesurface - ;
volume - ;
perimeter - ;
holessurface - ;
holesperimeter - ;
projectedarea - ;

```

### **- ZoneType:**

```

area - ;
perimeter - ;
holesperimeter - ;
wallsperimeter - ;
numberofcornersprojectedarea - ;
numberofconcavecorners - ;
surfaceareaofperimeterwall - ;
widthofdoors - ;
surfaceofdoors - ;
widthofwindows - ;
surfaceofwindows - ;
floorlevel - ;
subfloorthickness - ;
height - ;
netarea - ;
netperimeter - ;
volume - ;
areareducement - ;
calcareas - ;
totalextractedarea - ;
reducedextractedarea - ;
lowareaofzone - ;
extractedwallarea - ;
extractedcolumnarea - ;
extractedfillarea - ;
wallincettopsurface - ;
wallincetbacksidesurface - ;
wallincetsidesurface - ;
tsfloorplintuslength - (
);
tsseilingplintuslength - ;

```

### **- HatchType:**

```

surface - ;
perimeter - ;
holesperimeter - ;
holessurface - .

```

### **- LineType:**

```

length - .

```

**- PolyLineType:**

length -

**- ArcType:**

length -

**- CircleType:**

length -

**- \_SplineType:**

length -

**- ShellType:**

bottom\_surface - ;")

top\_surface - ;

volume - ;

surface - ;

holes\_surface -

**- MorphType:**

surface - ;

volume - ;

floorplanprojarea - ;

elevation - ;

baseheight - -

;

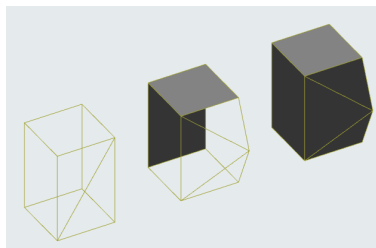
height - ;

floorplanprojperimeter - ;

numberofnodes - ;

numberofedges - ;

numberofhiddenedges - ;



numberofsoftedges - " " ;

numberofvisiblenodes - ;

numberoffacenodes - .

length\_edges - ( ).

length\_edge\_max -

length\_edges\_pgons - ( )

length\_edge\_pgons\_max - , .

**get\_layer\_by\_substring**

, ARCHICAD

" " "01 " "20 "

-

,

```

:
int ires = ac_request("get_layer_by_substring",string slayernamefragment, int from);
int ires = ac_request("get_layer_by_substring",string slayernamefragment, int from, int
range);

:
slayernamefragment - , ( " ").
from - ,
.
range - ,
ires - 0, .
ac_getstrvalue();

.
" " ,
"01 ".

```

```

string smalllayername="Квартиры";
string longlayername;
ires = ac_request("get_layer_by_substring",smalllayername,3); // начать сравнение с 3-й
позиции
if(ires != 0) {
    tsalert(-1,"Ошибка во время выполнения","Не обнаружен слой",smalllayername);
    cout << "Ошибка: Не обнаружен слой - "<< smalllayername << "\n";
}
longlayername = ac_getstrvalue(); // считать полное имя слоя

```

## elem\_user\_property

ARCHICAD

.. C++.

## get

```

:
int ires = ac_request("elem_user_property","get", string sparname);

:
svarname - ,
ires - 0, .
ac_getnumvalue() ac_getstrvalue();

.
" "
string sUP = "Полезная площадь здания";
int ires = ac_request("elem_user_property","get",sUP);
if(ires==0)
{
    istrue = ac_getnumvalue();
    if(istrue==1)
    {
        cout << "Эта зона относится к полезной площади здания";
    }
}

```

" " " "





```

    }
}

```

## Property\_Import

```

:
XML.
:
int iret = ac_request("elem_user_property", "Property_Import", string sXMLsource, int
conflictResolutionPolicy);

```

```

:
sXMLsource - XML
conflictResolutionPolicy -
:
0 -
1 -
2 -
:
XML
:

```

```

int do_iButtonloadProperties()
{
    cout << "load properties\n";
    int iStringDescr;
    object("create", "ts_string", iStringDescr);
    // загрузка свойств
    string spropfname = "Свойства КСИ";
    int iret = ts_string(iStringDescr, "ReadFromURL",
"http://www.labpp.ru/TSDownload/KSI/" + spropfname + ".txt");
    object("read_simple_value", iStringDescr);
    string sXMLsource = ac_getstrvalue();

    ts_dialogcontrol(iProgressBar, "SetValue", 2);
    if (iret == 0)
    {
        coutvar << sXMLsource;
        iret = ac_request("elem_user_property", "Property_Import", sXMLsource, 1);
        if (iret == 0)
        {
            tsalert(-3, "Сообщение", "Свойства успешно загружены", spropfname);
        }
    }
    object("delete", iStringDescr);
}

```

## get\_object\_property\_value

```

:
GDL
:
int iret = ac_request("get_object_property_value", string parametername);

ac_getnumvalue() ac_getstrvalue() -
:
iret 0-
:
"SNAME".

```

```

int iret = ac_request("get_object_property_value", "SNAME");
string svalue;
if(iret == 0)
{

```

```

    svalue = ac_getstrvalue();
    cout << svalue;    // написать полученное значение в окне сообщений
}

```

## set\_object\_property\_value

GDL

```

:
int iret = ac_request("set_object_property_value", string parametername, string/double
valuetowrite);

```

```

:
parametername - ;
valuetowrite - ;
            iret 0-

```

```

                                "SNAME" - "my new name".
int iret = ac_request("set_object_property_value", "SNAME", "my new name");
if(iret == 0)
{
    cout << "Запись прошла успешно";
}

```

## set\_object\_property\_value\_curdb

set\_object\_property\_value,

ARCHICAD.

## get\_element\_value

```

:
int ires = ac_request("get_element_value", string svaluenamename);
:
svaluenamename -
            - 0 -
                                ac_getnumvalue()    ac_getstrvalue();

```

```

int floornum;
int ires = ac_request("get_element_value", "StoryIndex");
if(ires == 0)
{
    floornum = ac_getnumvalue();
    cout << "Номер этажа=" << floornum;
}

```

svaluenamename:

### TypeID

### TypeName

### Layer

**ID**

ID

**StoryIndex****GuidAsText**

(guid)

**ObjectName****level**

Z

**pos.x**

X.

**pos.y**

Y.

**offset.x**

pos.x.

**offset.y**

pos.y.

**set\_element\_value****assign\_element\_values**

ac\_element

```
ac_request("assign_element_values",int iElemDescr, string paramname, string/double/int paramvalue);
```

**load\_element\_default\_values****set\_element\_infoltext**

ID

```
ac_request("set_element_infoltext", string svalue);
```

: svalue - ID

## create\_element\_on\_project

ac\_element.

```
int ires = acrequest("create_element_on_project",int iElemDescr);
```

```
iElemDescr -  
ires - 0.
```

## get\_element\_infoidtext

```
ID  
int ires = ac_request("get_element_infoidtext");  
ac_getstrvalue();  
ires - 0
```

## project\_property

### Руководство по LABPP\_Automat для ARCHICAD

#### CalcUnitsAreaDecimals

Получить заданное в проекте число точек после запятой.

Вызов:

```
ac_request("project_property","get","CalcUnitsAreaDecimals");
```

Результат считывается вызовом функции ac\_getnumvalue().

## autotext

ARCHICAD.

## get

```
int iret = ac_request("autotext","get", string svarname);  
: svarname -  
iret - 0,  
ac_getstrvalue();
```

```
string svaluenamе = "Адрес объекта";  
int iret = ac_request("autotext","get", svarname);  
if( iret == 0)  
{  
    string saddress = ac_getstrvalue();  
    cout << "Адрес = " << saddress;  
}  
else  
{
```

```
    cout << "Считать адрес не удалось. Возможно переменная " << svaluenam << " в проекте
не создана";
}
```

## set

Вызов:

```
int iret = ac_request("autotext","set", string svarname, string svarvalue);
```

```

:
svarname -
svalue -
iret - 0 -

"
"

string svaluenam = "Адрес объекта";
string snwvalue;
snwvalue = "г. Москва, Шаболовка, 37";
int iret = ac_request("autotext","set", svarname, snwvalue);
if( iret == 0)
{
    cout << "Адрес успешно изменен";
}
else
{
    cout << "Записать адрес не удалось. Возможно переменная " << svaluenam << " в проекте
не создана";
}
```

## create

```
int iret = ac_request("autotext","create", string svarname, string svalue);
```

```

:
svarname -
svalue -
iret - 0 -

"
"

string svaluenam = "Адрес объекта";
string snwvalue;
snwvalue = "г. Москва, Шаболовка, 37";
int iret = ac_request("autotext","create", svarname, snwvalue);
if( iret == 0)
{
    cout << "Переменная успешно создана";
}
else
{
    cout << "Создать переменную " << svaluenam << " не удалось";
}
}
```

## get\_as\_table

```
ac_request("autotext", "get_as_table", int iTable, int type);
```

Здесь:

iTable - дескриптор таблицы для заполнения данными из проекта;

формат колонок таблицы на выходе:

"text" (string) - имена переменных;

"key" (string) - внутренние идентификаторы переменных;

"value" (string) - значения.

type - битовый флаг для фильтрации типов переменных из проекта - 1-фиксированные (fixed)/2-собственные(custom)/4 - прочие (other).

Можно суммировать, например 1+2 означает выбрать фиксированные и собственные переменные (см.пример).

iTable

```
int iTable;
object("create", "ts_table", iTable, "Autotext variables list");
int type=1+2;
ac_request("autotext", "get_as_table", iTable, type);
```

## interface\_input2point

```
int iret ac_request("interface_input2dpoint", string firstMessage, double &x, double &y);

sFirstMessage -
x, y -
0,
```

## interface\_input2dline

```
2d
int iret ac_request("interface_input2dline", string sFirstMessage, string
sSecondMessage, double x1, double y1, double x2, double y2, double angleInRad, double
length);
```

```
sFirstMessage, sSecondMessage -
x1, y1, x2, y2 -
angleInRad -
length -
0,
```

## interface\_input3dline

```
3d
int iret ac_request("interface_input3dline", string sFirstMessage, string
sSecondMessage, double x1, double y1, double z1, double x2, double y2, double z2, double
angleInRad, double length);
```

```
sFirstMessage, sSecondMessage -
x1, y1, z1, x2, y2, z2 -
angleInRad -
length -
0,
```

## interface\_input2dpoly

```
2d
int ires = ac_request("interface_input2dpoly", string sMessage, double &square, double
&perimeter);
```

sMessage -  
square, perimeter

## get\_levelfromprojectnull\_by\_floorindexandlevel

```

(
)

:
:
ac_request("get_levelfromprojectnull_by_floorindexandlevel", int floorInd, double
bottomOffset, double &levelfromprojectnull);

:
floorind -
bottomOffset -
levelfromprojectnull -

:

0

1.

// загрузить из проекта в список 1 все элементы, у которых присвоено любое значение
классификатора ARCHICAD (не обязательно)
ac_request_special("add_elements_list", 1, "ZombieType", 2,
    "", "Cls", "Классификация ARCHICAD", "ASSIGNED", "", "");

// запросить количество собранных элементов
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue(); // получить количество в переменную
coutvar << icount; // вывести в окно сообщений

if (icount == 0)
{
    cout << "В списке нет элементов";
    return -1;
}

// объявляем необходимые переменные для работы
double dLevel, dHeight, elemlevel, elemprojectlevel;
int ielemstoryindex;
string sElemTypeName, sID;
int i;
int ires;
// выполнить цикл icount раз
for (i = 0; i < icount; i++)
{
    ires = ac_request("set_current_element_from_list", 1, i); // сделать текущим i-вый
    элемент из списка 1

    ac_request("get_element_value", "TypeName"); // запросить название типа этого
    элемента
    sElemTypeName = ac_getstrvalue();
    coutvar << sElemTypeName; // вывести имя типа элемента в окно сообщений

    // если это элемент типа Морф:
    if (sElemTypeName == "MorphType")
    {
        ires = ac_request("get_element_value", "ID"); // запросить ID элемента
        sID = ac_getstrvalue(); // получить его в переменную
        coutvar << sID; // вывести ID в окно сообщений
        ires = ac_request("get_element_value", "Level"); // запрашиваем свойство
    }
}

```



```

Level (для Морф - высота над уровнем его этажа)
    dLevel = ac_getnumvalue(); // получаем его в переменную
    coutvar << dLevel; // выводим в окно сообщений
    ires = ac_request("get_quantity_value", "max_height"); // запрашиваем
максимальный размер Морф по вертикали
    dHeight = ac_getnumvalue(); // получаем его в переменную
    coutvar << dHeight; // выводим в окно сообщений
    elemlevel = dLevel + dHeight / 2; // рассчитываем уровень центра Морф по
высоте относительно этажа
    ac_request("get_element_value", "StoryIndex"); // запрашиваем индекс этажа
    ielemstoryindex = ac_getnumvalue(); // получаем индекс этажа в переменную
    // получаем высоту центра Морф относительно 0 проекта в переменную
elemprojectlevel
    ac_request("get_levelfromprojectnull_by_floorindexandlevel",
ielemstoryindex, elemlevel, elemprojectlevel);
    coutvar << elemprojectlevel; // выводим в окно сообщений
}
}

```

Вариант 2. Сразу выбираем только элементы типа Морф, которые классифицированы как "Лестница"

```

// загрузить из проекта в список 1 элементы типа Морф, у которых присвоено значение
классификатора ARCHICAD "Лестница"
ac_request_special("add_elements_list", 1, "MorphType", 2,
    "", "Cls", "Классификация ARCHICAD", "=", "Лестница", "");

// запросить количество собранных элементов
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue(); // получить количество в переменную
coutvar << icount; // вывести в окно сообщений

if (icount == 0)
{
    cout << "В списке нет элементов типа Морф, классифицированных как \"Лестница\"";
    return -1;
}

// объявляем необходимые переменные для работы
double dLevel, dHeight, elemlevel, elemprojectlevel;
string sID;
int ielemstoryindex;
int i;
int ires;
// выполнить цикл icount раз
for (i = 0; i < icount; i++)
{
    ires = ac_request("set_current_element_from_list", 1, i); // сделать текущим i-ый
элемент из списка 1
    ires = ac_request("get_element_value", "ID"); // запросить ID элемента
    sID = ac_getstrvalue(); // получить его в переменную
    coutvar << sID; // вывести ID в окно сообщений
    ires = ac_request("get_element_value", "Level"); // запрашиваем свойство Level
(для Морф - высота над уровнем его этажа)
    dLevel = ac_getnumvalue(); // получаем его в переменную
    coutvar << dLevel; // выводим в окно сообщений
    ires = ac_request("get_quantity_value", "max_height"); // запрашиваем максимальный
размер Морф по вертикали
    dHeight = ac_getnumvalue(); // получаем его в переменную
    coutvar << dHeight; // выводим в окно сообщений
    elemlevel = dLevel + dHeight / 2; // рассчитываем уровень центра Морф по высоте
относительно этажа
    ac_request("get_element_value", "StoryIndex"); // запрашиваем индекс этажа
    ielemstoryindex = ac_getnumvalue(); // получаем индекс этажа в переменную

```

```

        // получаем высоту центра Морф относительно 0 проекта в переменную
elemprojectlevel
    ac_request("get_levelfromprojectnull_by_floorindexandlevel", ielemstoryindex,
elemlevel, elemprojectlevel);
    coutvar << elemprojectlevel; // выводим в окно сообщений
}

```

## get\_floorindexandlevel\_by\_levelfromprojectnull

```

:
ac_request("get_floorindexandlevel_by_levelfromprojectnull", double
levelfromprojectnull, int &floorInd, double &bottomOffset);

```

```

:

levelfromprojectnull -
floorInd - - ,
bottomOffset - - ,

```

## get\_storyinfo\_as\_table

```

:
int ierr = ac_request("get_story_info_as_table", int iTableStories);

```

```

:

iTableStories -
ierr - ( 0 - ).

```

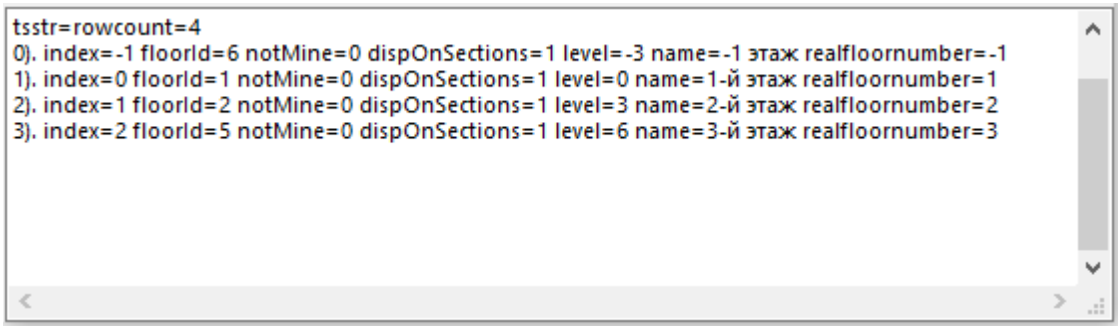
index	int	
floorId	int	Id
notMine	int	( )
dispOnSections	int	
level	double	
name	string	
realfloornumber	int	

```

int iTableStories;
object("create", "ts_table", iTableStories);
ac_request("get_story_info_as_table", iTableStories);
string tsstr;
ts_table(iTableStories, "print_to_str", tsstr);
coutvar << tsstr;
object("delete", iTableStories);

```

Результат:



elem\_classification\_get\_item\_value

```

:
int err = ac_request("elem_classification_get_item_value",string sclassname);

:
err = 0
sclassname -
ac_getstrvalue();
```

elem\_classification\_get\_parent\_item\_value

```

:
int err = ac_request("elem_classification_get_item_value",string sclassname);

:
err = 0
sclassname -
ac_getstrvalue();
```

hotlinks

GetHotlinkNodes

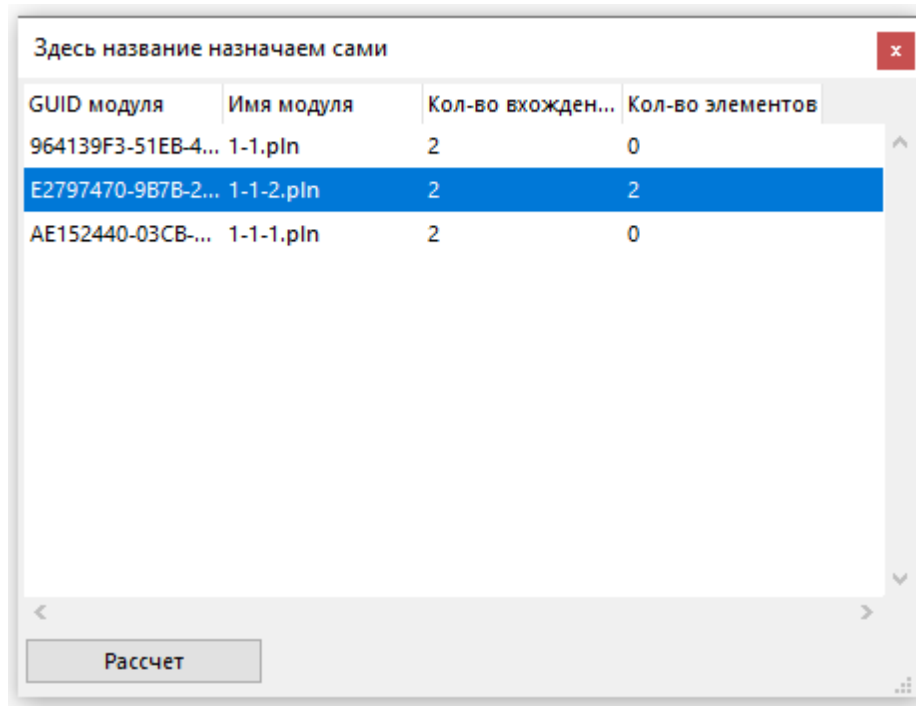
```

:
int err = ac_request("hotlinks","GetHotlinkNodes",int iTable);

:
err -
iTable -
0
(
).
```

guid	string	guid
name	stting	
type	int	0/1/2 - / ARCHICAD/dwg dxf .
refFloorName	string	,
refFloorInd	int	,
instancesCount	int	

guid, name instancesCount iListBox.



```

//*****
// Create dialog and listbox
// Get from project list of the modules
// Print list to message window
// Place 3 columns of the list to listbox
// Calc beams in modules by pressing button
//*****

int iDialogDescr; // Дескриптор диалога
int iListBoxNodes, iTableNodes; // Листбокс элементов hotlink nodes

int iTable;

int iButtonCalcElementsCountInModule;

int main()
{
#pragma region Создаем диалог
    int x, y, w, h;
    object("create", "ts_dialog", iDialogDescr);
    ts_dialog(iDialogDescr, "init_dialog", "palette", 0, 0, 450, 310); // Создаем окно
    диалога как палитку, т.е. немодальное
    ts_dialog(iDialogDescr, "set_as_main_panel"); // Если так сделать, то все
    немодальные окна этого сеанса будут закрываться вместе с этим окном
    ts_dialog(iDialogDescr, "SetTitle", "Здесь название назначаем сами");

    // листбокс
    object("create", "ts_dialogcontrol", iListBoxNodes, "iListBoxNodes");
    ts_dialogcontrol(iListBoxNodes, "init_control", "singlesellistbox", iDialogDescr,
3, 3, 450-3, 300-5-22, 48, 20);
    ts_dialogcontrol(iListBoxNodes, "SetAnchorToPanelResize", 0, 0, 1, 1);

    // кнопка расчета для позиции диалога
    int delta = 3;
    int yy = 282;
    x = 3; y = yy; w = 120; h = 20;

    object("create", "ts_dialogcontrol", iButtonCalcElementsCountInModule,

```

```

"iButtonCalcElementsCountInModule");
    ts_dialogcontrol(iButtonCalcElementsCountInModule, "init_control", "button",
iDialogDescr, x, y, w, h);
    ts_dialogcontrol(iButtonCalcElementsCountInModule, "eventreaction",
"Event_ButtonClicked");
    ts_dialogcontrol(iButtonCalcElementsCountInModule, "settext", "Рассчет");
    ts_dialogcontrol(iButtonCalcElementsCountInModule, "SetAnchorToPanelResize", 0, 1,
0, 0);
    ts_dialogcontrol(iButtonCalcElementsCountInModule, "SetToolTip", "Рассчитать
количество элементов балок в модуле");

#pragma endregion

    object("create", "ts_table", iTable);
    // получить список hotlinks - колонки таблицы сформируются автоматически
    ac_request("hotlinks", "GetHotlinkNodes", iTable);
    // добавить в конец колонку "количество элементов"
    ts_table(iTable, "add_column", -1, "int", "Кол-во элементов");

    // вывести список hotlink nodes в окно сообщений
    string str;
    ts_table(iTable, "print_to_str", str);
    coutvar << str;

    // задать имена колонок перед экспортом в листбокс
    ts_table(iTable, "mapping_columns_to_export",
        "guid", "GUID модуля", "",
        "name", "Имя модуля", "",
        "instancesCount", "Кол-во вхождений в проект", "",
        "Кол-во элементов", "", "" // имя колонки оставляем как есть
    );
    // задать ширину колонок при экспорте
    ts_table(iTable, "set_columns_width_when_export", 100);
    // экспортировать таблицу в листбокс
    ts_table(iTable, "export_to_dialogcontrol", iListBoxNodes, -1, -1);

    //object("delete", iTable); если объект таблицы больше не потребуется, то можно ее
удалить
    //но если потом будем к ней обращаться из обработчиков событий (кнопки и т.п.) то
оставляем в памяти

    bool bres;
    // запускаем диалог
    ts_dialog(iDialogDescr, "invoke", bres);
}

// обработчик событий кнопок на нажатие
int Event_ButtonClicked(int iDescr, string sDescr)
{
    if (sDescr == "iButtonCalcElementsCountInModule") {
        cout << sDescr << "\n";
        do_iButtonCalcElementsCountInModule();
    }
}

int do_iButtonCalcElementsCountInModule()
{
    // получить имя модуля, выбранного в листбоксе
    int item;
    string smodulename;

    ts_dialogcontrol(iListBoxNodes, "GetSelectedItem", item);
    if (item == 0) {

```

```

        return -1;
    }

    ts_table(iTable, "select_row", item - 1);
    ts_table(iTable, "get_value_of", "name", smodulename);
    // выбрать элементы, у которых совпадает имя модуля с выбранным в листбоксе
    ac_request_special("load_elements_list", 2, "BeamType", 2,
        "", "EP", "hotlinkName", "=", smodulename, "");

    ac_request("get_loaded_elements_list_count", 2);
    int icount = ac_getnumvalue();

    // записать результат в текущую строку таблицы
    ts_table(iTable, "set_value_of", "Кол-во элементов", icount);
    coutvar << icount;
    // экспортировать таблицу в листбокс (вид колонок уже был задан при первом
    экспорте в main())
    ts_table(iTable, "export_to_dialogcontrol", iListBoxNodes, -1, -1);
    ts_dialogcontrol(iListBoxNodes, "SelectItem", item);
}

```

## geometry\_calc\_2d

### is\_point\_on\_element\_polygon

```

X,Y
int err = ac_request("geometry_calc_2d","is_point_on_element_polygon",double X, double Y,
int iElemDescr);
или
int err = ac_request("geometry_calc_2d","is_point_on_element_polygon",double X, double Y,
int iElemDescr, bool mainContOnly, double grow_contour, double grow_holes);

```

```

:
X,Y -
iElemDescr -
mainContOnly -
grow_contour -
;
grow_holes -
0
.
ac_getnumvalue() - 1 -
, 0 -
(
, mainContOnly - true)

```

### is\_curelem\_inside\_element\_polygon

```

,
.
:
int ires = ac_request("geometry_calc_2d","is_curelem_inside_element_polygon", int
iElemGuidDescr);
или
int ires = ac_request("geometry_calc_2d","is_curelem_inside_element_polygon", int
iElemGuidDescr, bool mainContOnly, double grow_contour, double grow_holes, int mode);

```

```

:
iElemmmDescr -
mainContOnly - 0/1 - 0,
iElemDescr
guid ("ts_guid").
, 1 -
,
.

```

```

grow_contour -
grow_holes -
mode - iElemDescr : -1 -
, 0 - (pos.x pos.y), 1 - , 2 -

```

## is\_point\_on\_element\_refside

```

:
ac_request("is_point_on_element_refside", int px, int py, int elemguid, bool
&bIsOnRefSide);

```

```

:
px, py - ;
blsOnRefSide - , oppSide -

```

## where\_C\_leftorright

```

:
ac_request("geometry_calc_2d","where_C_leftorright",double X1, double Y1, double X2,
double Y2, double pX, double pY, double &dist);

```

```

:
X1, Y1, X2, Y2 - ,
pX, pY -
dist - 0 - , 0 - , > 0 -
- < 0,
- 0, - >0.

```

## getDistPointFrom2dLine

```

ac_request("geometry_calc_2d","getDistPointFrom2dLine",double X1, double Y1, double X2,
double Y2, double pX, double pY, double &dist);

```

Здесь:

X1, Y1, X2, Y2 - координаты прямой;

pX, pY - координаты тестируемой точки;

dist - результат, расстояние точки от прямой;

## geometry\_calc\_3d

### RotatePoint3dAroundVect3d

```

- 0,0,0.

```

```

ac_request("geometry_calc_3d","RotatePoint3dAroundVect3d",double rx, double ry, double
rz, double alpha, double x, double y, double z, double& resx, double& resy, double&
resz);

```

```

:

```

```

rx,ry,rz - , ;
alpha - ;
x,y,z - ;
resx, resy, resz - .

```

## GetBounds

```

:
err = ac_request("geometry_calc_3d","GetBounds",double xMin, double yMin, double zMin,
double xMax, double yMax, double zMax);

```

## classification

```

:
int err = ac_request("elem_classification_get_item_value",string sclassifname);
:
err = 0
sclassifname - ,
ac_getstrvalue();

```

## find\_item\_by\_name

```

:
int err = ac_request("classification","find_item_by_name", string classifname, string
itemname, int iGuidIdItem);
:
err = 0
sclassifname - ,
itemname - ,
iGuidIdItem - ts_guid, guid
ARCHICAD.

```

## check\_class\_is\_inbranch

```

:
bool bres = check_class_is_inbranch(UniString classification, UniString classname1,
UniString classnametest)
:
bres - true classnametest clsaaname1.
classnametest - ,
classname1 - , classnametest.
classification - ,

```

## CreateClassificationSystem

```

:
int err = ac_request("classification","CreateClassificationSystem", int iguiddescr,
string name, string description, string source, string editionVersion, int year, int
month, int day);
:
err = 0
iguiddescr - ,
name - ,
description - ,
source - ,

```



```

editionVersion -
year -
month -
day -

```

## CreateClassificationItem

```

:
int err = ac_request("classification","CreateClassificationItem", int
iguiddescritem/string sguidresultitem, string id, string name, string description, int
isystemguid/string ssystemguid, int iparentitemguid, int inextitemguid)

:
err = 0
iguiddescritem - ts_guid guid
sguidresultitem - guid
id - id
name -
description -
isystemguid - ts_guid guid'
ssystemguid - guid
iparentitemguid - guid'
-1,
sparentitemguid - guid
inextitemguid - (-1)
snextitemguid -

```

## DeleteClassificationItem

```

guid.
:
int err = ac_request("classification","DeleteClassificationItem", int
iguiddescritem/string sguiditem)

```

```

:
err = 0
iguiddescritem - ts_guid
sguiditem - guid

```

## set\_item\_to\_curelem

```

:
int err = ac_request("classification","set_item_to_curelem", iGuidItem);
:
err = 0
iGuidItem - guid
ts_guid.
"find_item_by_name"
guid

```

## GetClassificationSystems

```

:
int iret = ac_request("classification", "GetClassificationSystems", int
iTableClassifiersInProject);
:
iTableClassifiersInProject -

```

В iret - код ошибки или 0.

В iTableClassifiersInProjectLoc формируется таблица с колонками:

```

"guid" - строка;
"name" - строка;
"description" - строка;
"source" - строка;
"editionVersion" - строка;
"editionDate" - дата.

```

### Пример:

```

int iTableInClassifiersInProjectLoc;
object("create", "ts_table", iTableClassifiersInProjectLoc);
iret = ac_request("classification", "GetClassificationSystems",
iTableClassifiersInProjectLoc);

```

## GetClassificationSystem\_guid\_by\_name

```

guid
:
int ires = ac_request("classification", "GetClassificationSystem_guid_by_name", string
classifsysname, string classifsysver, int isystemguid/string ssystemguid);

```

```

:
classifsysname -
classifsysver -
isystemguid - ts_guid guid
ssystemguid - guid
:
"
"
```

```

string classifsysname = "Структура ЦИМ", classifsysver = "1.0";
string systemguid;
int ires = ac_request("classification", "GetClassificationSystem_guid_by_name",
classifsysname, classifsysver, systemguid);
if (ires == 0)
{
string id, name, description;
id = "new id"; name = "new name"; description = "new descr";
string snewguid;
ac_request("classification", "CreateClassificationItem", snewguid, id, name,
description, systemguid, -1, -1);
}

```

## Classification\_Import

Импортировать в проект классификатор, представленный в виде xml текста.

```
iret = ac_request("classification", "Classification_Import", strlen sXMLsource, int
iclassifpolicy, int ipropertypolicy);
```

Здесь:

sXMLsource - исходный текст классификатора в xml кодировке,

iclassifpolicy - как поступать при конфликтах во время загрузки классификатора:

0 - объединить с существующей классификацией,

1 - заменить,

2 - пропустить.

ipropertypolicy - как поступать при конфликтах во время загрузки параметров у классификации:

0 - объединить,

1 - пропустить.

Пример:

Получить с сайта текст классификатора из файла и загрузить его в проект.

```
int do_iButtonloadClassifier()
{
    string sclassifname = "Классификация КСИ- Объекты капитального строительства";

    int iStringDescr;
    object("create", "ts_string", iStringDescr);
    int iret = ts_string(iStringDescr, "ReadFromURL",
"http://www.labpp.ru/TSDownload/KSI/" + sclassifname + ".txt");
    object("read_simple_value", iStringDescr);
    string sXMLsource = ac_getstrvalue();

    if (iret == 0)
    {
        coutvar << sXMLsource;
        iret = ac_request("classification", "Classification_Import", sXMLsource, 1,
0);

        if (iret == 0)
        {
            tsalert(-3, "Сообщение", "Классификатор успешно загружен",
sclassifname);
        }
    }
    object("delete", iStringDescr);
    refresh_classifiers_list_for_loaded();
}
```

## GetClassifSystem\_By\_ClassItemGuid

guid .

```
int err = ac_request("classification", "GetClassifSystem_By_ClassItemGuid", int/string
guid, int/string classifsystemguid, string name, string description, string source,
string editionVersion, string editionDate);
```

Здесь:

guid - GUID позиции класса, для которого надо узнать данные классификатора. Задается как дескриптор или текст;

classifsystemguid - результат - guid классификатора (текст или дескриптор);

name, description, source, editionVersion, editionDate - результат - данные классификатора.

## do\_move\_rot\_and\_scale\_elemlist

```
ac_request("do_move_rot_and_scale_elemlist", int iElemList, double coordtoX, double
coordtoY, double coordRotCenterX, double coordRotCenterY, double angle, double scale,
bool bDoRotate, bool bWithDelete, int bpt, int mmode, int iElemList2);
```

Здесь:

iElemList - номер списка элементов, которые копировать или перемещать;  
coordtoX, coordtoY - координаты куда переместить;  
coordRotCenterX, coordRotCenterY - координаты центра поворота;  
angle - угол в радианах;  
scale - масштаб (доли 1);  
bDoRotate - поворачивать или нет;  
bWithDelete - удалять оригиналы или нет;  
bpt - что взять за базовую точку  
0 - TopLeft,  
1 - TopMiddle,  
2 - TopRight,  
3 - MiddleLeft,  
4 - MiddleMiddle,  
5 - MiddleRight,  
6 - BottomLeft,  
7 - BottomMiddle,  
8 - BottomRight,  
9 - Default,  
10 - ByParentCoordCenter.  
mmode - режим учета групп  
каждый элемент по отдельности BySingle=0,  
по последней группировке ByGroupsBelow=1,  
по первой группировке ByGroupsTop=2  
все как единое целое ByAll=3  
iElemList2 - номер списка элементов, куда поместить вновь созданные элементы.

Пример

Скопировать элемент кровли на расстояние 1 м вправо, находящийся в списке 2 и записано созданный элемент в список 4,  
и задать номер многослойной конструкции 111.

```
int iElemList = 2;
int iElemList2 = 4;
double coordtoX = 1;
double coordtoY = 0;
double coordRotCenterX = 0;
double coordRotCenterY = 0;
double angle = 0;
double scale = 1;
bool bDoRotate = false;
bool bWithDelete = false;
int bpt = 0;
int mmode = 0;
```

```
coordtoX = coordtoX + offset;
ac_request("do_move_rot_and_scale_elemlist", iElemList, coordtoX, coordtoY,
coordRotCenterX, coordRotCenterY, angle, scale, bDoRotate, bWithDelete, bpt, mmode,
iElemList2);
ac_request("set_current_element_from_list", 4, 0);
int compositindex = 111;
ac_request_special("set_element_value", "EP", "composite", compositindex);
```

## clearmessagewindow

```

:
ac_request("clearmessagewindow");

```

## get\_labpp\_product\_name

```

:
    add-ons LABPP.
    "LabPP_Calc".
:
ac_request("get_labpp_product_name");

    ac_getstrvalue();

:

ac_request("get_labpp_product_name");
string sproductname = ac_getstrvalue();
if (sproductname != "LabPP_Calc")
{
    ts_dialogcontrol(iButton_ToCalc_DS, "disable");
}

```

## ac\_request\_special

---

### get\_pie

```

:
    1,
    2.
:
ac_request2("get_pie", int pieindex, int iTable1, int iTable2);

:
pieindex -
:
    (
    ).
    -1,
:
iTable1 -
    ts_table
:
    (
:
"composite_index" int -
:
"piename" string -
:
"totalThick" double -
:
"nComps" int -
:
iTable2 -
    ts_table
:
:
"composite_index" int -
:
"buildingmaterial" int -
:
"pielayername" string -
:
"ID" string - id
:
"thickness" double -
:
"flagbits" int -
:
"leftright" int -
:
- 0, - 1, 2);
"cutMaterial" int -
:
"cutFill" int -
:

```

### load\_elements\_list

ARCHICAD

```

:
:
ac_request("load_elements_list",int iListNum,string sElemTypeName,"MainFilter",int
iMainFilterValue,string filterparametrname, string/double filterparametervalue,...);

:
iListNum - ( 0...9).
sElemTypeName - - "ZombieElemType",
"ColumnType", - "WallType" . .
"MainFilter" - ,
. . iMainFilterValue -
- 3.
"ID"," ID" - , ID=" ID" . .
"MainFilter" , "MainFilter",0.
. .

MainFilterValue=j1+2*j2+4*j3+8*j4+16*j5+32*j6+64*j7+128*j8+256*j9+512*j10+1024*j11+2048*j12+4096*j13
+268435456*j14,
j 0 1.
j1:
j2:
j3:
j4: 3d
j5:
j6:
j7:
j8:
j9: , variationID
j10:
j11:
j12: ; ,
j13: ;
j14: 2d

:
ac_request_special("load_elements_list", int iListNum, string ElemTypeName, int
iMainFilterValue,
string left_braces1, string genesis1, string parname1, string operation1, string
value1, string right_braces1, string operation1and2,
string left_braces2, string genesis2, string parname2, string operation2, string
value2, string right_braces2, string operation2and3,
...
string left_bracesN, string genesisN, string parnameN, string operationN, string
parvalueN, string right_bracesN);

:
left_braces1 - , "("
right_braces - , ")))".
genesis1 - ,
:
"ANY" - ,
"GDL" - GDL ,
"EP" - ,

```

```

"UP" -
"UPb" -
"UPub" -
"UPa" -
"CIs" -
"IFC" - UFC,
"UTS" -
"Q" -

value1 -

operation1and2 -
    " " (AND)    " " (OR).

```

```

:

ac_request_special("load_elements_list", int iListNum, string ElemTypeName, int
iMainFilterValue, int iTableFilter);

```

Здесь:

iListNum - номер списка, в который нужно загрузить элементы,

ElemTypeName - низвание типа выбираемых элементов,

iMainFilterValue - , . ( . ).

```

iTableFilter -
,
.
:
1 , 1 2.

```

```

int iTableFilter;
object("create", "ts_table", iTableFilter);
ts_table(iTableFilter, "create_structure_for_filter_table");
ts_table(iTableFilter, "add_row_to_filter_table", "", "EP", "ObjectName", "CONTAINS",
"labelLabPP_Flat", "", "AND");
// дополнить таблицу фильтра элементов индексами этажей
ts_table(iTableFilter, "add_row_to_filter_table", "", "EP", "StoryIndex", "!=", 1, "",
"AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "EP", "StoryIndex", "!=", 2, "",
"");
ac_request_special("load_elements_list", 1, "ObjectType", 2, iTableFilter);
object("delete", iTableFilter);

```

## add\_elements\_list

То же что "load\_elements\_list", но перед заполнением список не очищается а дополняется новыми элементами, удовлетворяющими условиям выборки.

## load\_elements\_list\_from\_selection

То же что "load\_elements\_list", но выборка происходит из выбранных элементов.

## add\_elements\_list\_from\_selection

То же что "load\_elements\_list", но выборка происходит из выбранных элементов и перед заполнением список не очищается а дополняется новыми элементами, удовлетворяющими условиям выборки.

## load\_elements\_list\_curdb

```

, "load_elements_list"
.
```

## copy\_elements\_to\_list\_from\_list\_by\_filter

```
, "load_elements_list",
```

## check\_cur\_element\_by\_filter

```
"load_elements_list",
```

## GetMaterialSurfaceSquare

```

:
( Morph
CTRL+SHIFT
-
).
Morph.

:
int ires = ac_request_special("GetMaterialSurfaceSquare", string materialname);

int ires = ac_request_special("GetMaterialSurfaceSquare", int materialindex);
:
materialname - ( , , .. ).
materialindex -
-1
Morph,
"
.

ires = ac_request_special("GetMaterialSurfaceSquare", "Лестница - ступень");
square = ac_getnumvalue();
cout << "Площадь ступеней=" << square << "\n";

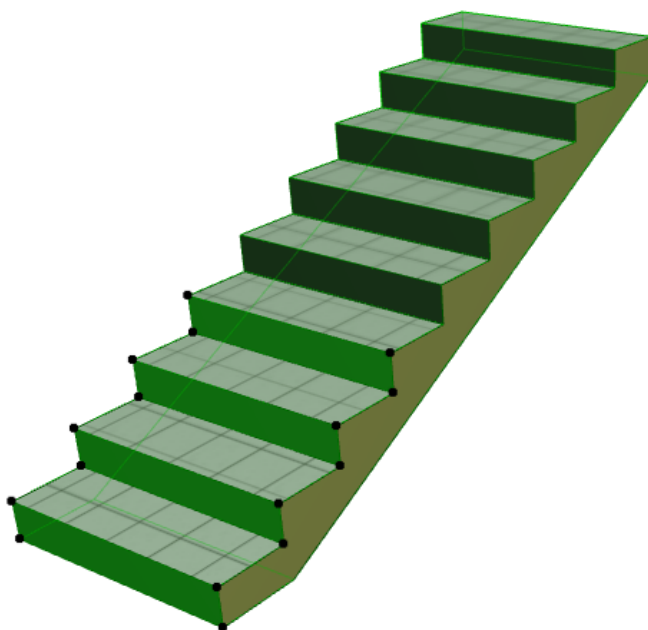
```

## GetMaterialSurfaceSquareTable

```

: Morph
( CTRL+SHIFT
-
).

```





```

:
int ires = ac_request_special("GetMaterialSurfaceSquareTable", int iTableDescr);
:
iTableDescr -

```

```

"materialindex" - ;
"materialname" - ;
"square" -

```

Morph

```

int main()
{
    ac_request_special("load_elements_list_from_selection", 1, "MorphType", 0);
    ac_request("get_loaded_elements_list_count", 1);
    int icount = ac_getnumvalue();
    if (icount == 0)
    {
        cout << "Сначала выберите элемента типа Morph";
        return;
    }

    ac_request("set_current_element_from_list", 1, 0);

    int iTableMat;
    object("create", "ts_table", iTableMat);
    ac_request_special("GetMaterialSurfaceSquareTable", iTableMat);
    string str;
    ts_table(iTableMat, "print_to_str", strrrr);
    coutvar << str;
    object("delete", iTableMat);
    cout << "Программа успешно завершена";
}

```

Результат:

```

str=rowcount=4
0). materialindex=250 materialname=Поверх.-Бетон уложенный square=1.133147
1). materialindex=194 materialname=Плитка керамическая 200x200 square=2.7
2). materialindex=3 materialname=Краска - Темно-Серая square=1.395
3). materialindex=74 materialname=Краска - Желтая Охра square=0.637526

```

## get\_element\_value

Считать значение свойства или параметра у текущего элемента.

### buildingMaterialName\_thickness

Для текущего элемента рассчитать общую толщину стройматериалов многослойки или одного материала, которые использованы в назначении элементу при условии, что имя материала отвечает шаблону.

```

int err = ac_request_special("get_element_value", "EP",
"buildingMaterialName_thickness( \"(?=.*Минерал.*)(?=.*ват.*)(?=.*плит.*)\"),int
iTableResult);
или

```

```
int err = ac_request_special("get_element_value", "EP",
"buildingMaterialName_thickness( \"(?=.*Минерал.*)(?=.*ват.*)(?=.*плит.*)\");
```

Результат получаем через `ac_getnumvalue()`;

Если указан дескриптор таблицы, в нее записываются характеристики материалов, имена которых отвечают шаблону, а толщины одинаковых материалов складываются.

Т.е. если в многослойке есть один слой утеплителя одного типа и два слоя утеплителя другого типа - то в результате будет посчитана общая толщина утеплителя, а в таблицу выпадут две строки - утеплитель первого типа с толщиной, и утеплитель второго типа с суммарной толщиной его двух слоев.

### buildingMaterialName\_used

Для текущего элемента рассчитать общую толщину стройматериалов многослойки или одного материала, которые использованы в назначении элементу при условии, что имя материала отвечает шаблону.

```
int err = ac_request_special("get_element_value", "EP", "buildingMaterialName_used( \"(?=.*Минерал.*)(?=.*ват.*)(?=.*плит.*)\");int iTableResult);
```

или

```
int err = ac_request_special("get_element_value", "EP", "buildingMaterialName_used( \"(?=.*Минерал.*)(?=.*ват.*)(?=.*плит.*)\");
```

Результат получаем через `ac_getnumvalue()`;

Если указан дескриптор таблицы, в нее записываются характеристики материалов, имена которых отвечают шаблону, а толщины одинаковых материалов складываются.

Т.е. если в многослойке есть один слой утеплителя одного типа и два слоя утеплителя другого типа - то в результате будет посчитана общая толщина утеплителя, а в таблицу выпадут две строки - утеплитель первого типа с толщиной, и утеплитель второго типа с суммарной толщиной его двух слоев.

Свойство может указываться в фильтрах для отбора строк по табличному фильтру.

building_material_index	int	ARCHICAD
name	string	ARCHICAD
id	string	ID ARCHICAD
thickness	double	

### set\_element\_value

### set\_element\_value\_curdb

```
, "set_element_value"
```

### RunGDLParScript

```
GDL
```

```
:
int ires = ac_request_special("RunGDLParScript");
```

### get\_building\_material

```
:
int ires = ac_request_special("get_material", int bmatindex, int iTableBMat);
```

```

:
iTableBMat - ,
matindex ,
-1 - .

:
"building_material_index" int - ,
"name" string - ,
"id" string - ID ,
"cutMaterial" string - ,
"strMaterialGuid" - guid .

```

## get\_build\_material\_value

```

:
int ires = ac_request_special("get_build_material_value", int ibuildmatindex, string
genesis, void &variable);

int ires = ac_request_special("get_build_material_value", string sbuildmatname, string
genesis, void &variable);
:
ibuildmatindex - ,
sbuildmatname - .
genesis - , "UP" -
variable - ,
.
ts_table.

:
" "
.

int bmatindex;
string bmatname, bmatid, bmatnameenglish;
// считать у текущего элемента назначенный стройматериал
ac_request("get_element_value", "buildingMaterial");
bmatindex = ac_getnumvalue();
// получить имя этого стройматериала
err = ac_request("get_bmnamebyindex", bmatindex, bmatname, bmatid);
// получить имя этого стройматериала на английском из пользовательского свойства
стройматериала
ac_request_special("get_build_material_value", bmatindex, "UP", "Имя стройматериала на
английском");
bmatnameenglish = ac_getstrvalue();
// записать все эти свойства стройматериала в пользовательские свойства текущего элемента
ac_request("elem_user_property", "set", "Имя материала у элемента", bmatname);
ac_request("elem_user_property", "set", "Имя материала у элемента на английском",
bmatnameenglish);

```

## get\_material

```

( ) ,
( ) .
:
int ires = ac_request_special("get_material", int matindex, int iTableMat);
:
iTableMat - ( ).
matindex ,
-1 - ( ).
Структура таблицы на выходе:
"material_index" int - ( ),

```

```

"name" string - ( ),
"texturepath" string - ,
"surfaceRGB" string - RGB ,
"surfaceRGB255" - string - RGB ( "255,100,20").

```

## load\_zonerelatedwallstolist

```

:
int ires = ac_request_special("load_zonerelatedwallstolist", int iZoneGuid, int
ilistindex);
:
iZoneGuid - ts_guid, guid ,
ilistindex - , .

```

## load\_zonerelatedwallparttable

```

:
int ires = ac_request_special("load_zonerelatedwallstolist", int iZoneGuid, int
iTableDescr);
:
iZoneGuid - ts_guid, guid ,
iTableDescr - ts_table .

"GUID" - - guid , ;
"roomEdge" - - , ;
"tBeg" - - ,
"tEnd" - - ,

```

## Attribute\_GetNum

```

:
ac_request2("Attribute_GetNum", int iattrtype, int &icount);
:
iattrtype - .
Перья - 1,
Слои - 2,
Типы линий - 3,
Типы штриховок - 4,
Пироги материалов - 5,
Материалы - 6,
Города - 7,
Комбинации слоев - 8,
Категории зон - 9,
Шрифты - 10,
Профили - 11,
Таблицы перьев - 12,
Стандарты размеров - 13,
Список модельных видов - 14,
МЕР системы - 15,
Энергетические назначения - 16,
Строительные материалы - 17,
Стили маркировки - 18
icount - возвращается количество атрибутов заданного типа.

```

## clipboard\_set

```

Windows.
Windows (Ctrl+V      Shift+Ins).
:
ac_request_special("clipboard_set", string str);
:
str - , , Windows.

```

## clipboard\_get

```

Windows.
:
ac_request_special("clipboard_get",string &str);
:
str - , Windows.

```

## element\_user\_data

```

.
,
, ..
.
.
"read"
( , - ( ), ,
..
- "write".
- .
ts_table.

```

## read

```

( , , . .)
:
int ires = ac_request_special("element_user_data","read");

```

## write

```

:
int ires = ac_request_special("element_user_data","write");

```

## write\_to\_file

Записать блок данных пользовательских данных в файл.  
 int ires = ac\_request\_special("element\_user\_data", "read\_from\_file", string fullpath);  
 Возвращает в ires 0 в отсутствии ошибок.  
 Если возникла ошибка - возвращается -1.  
 fullpath - путь к файлу, откуда считать блок данных.

Пример.

Упаковать рабочие таблицы в блок пользовательских данных и записать блок в файл и обратная операция - прочитать из файла и распаковать в таблицы.

```

int WriteTablesToFile()
{

```

```

string str, tmppath, filename;
shell_func("get_path", "tmp", tmppath);
filename = "LabPP_SAINt-GOBAIN.ini";
tmppath += "\\\" + filename;
coutvar << tmppath;

ac_request_special("element_user_data", "clear");
ts_table(iTableMaterials, "print_to_str_as_json", str);
coutvar << str;
ac_request_special("element_user_data", "set_variable_value", "iTableMaterials",
str);
ts_table(iTableSolutionsMats, "print_to_str_as_json", str);
ac_request_special("element_user_data", "set_variable_value",
"iTableSolutionsMats", str);

ts_table(iTableSolutions, "print_to_str_as_json", str);
ac_request_special("element_user_data", "set_variable_value", "iTableSolutions",
str);
ts_table(iTableSysProps, "print_to_str_as_json", str);
ac_request_special("element_user_data", "set_variable_value", "iTableSysProps",
str);

int ires = ac_request_special("element_user_data", "write_to_file", tmppath);
if (ires < 0)
{
    tsalert(-2, "Предупреждение", "Не удалось записать данные в файл " +
tmppath, "Этот файл предназначен для быстрой загрузки данных GYPROC. Возможно этот путь
не доступен для записи. Настройте config.cpp", "Ok");
    return -1;
}

int ReadTablesFromFile()
{
    string str, tmppath, filename;
    filename = "LabPP_SAINt-GOBAIN.ini";
    shell_func("get_path", "tmp", tmppath);
    tmppath += "\\\"+filename;
    int ires = ac_request_special("element_user_data", "read_from_file", tmppath);
    if (ires < 0)
    {
        tsalert(-2, "Предупреждение", "Не удалось считать данные из файла " +
tmppath, "Этот файл содержит информацию GYPROC из файла Excel для быстрой загрузки",
"Ok");
        return -1;
    }

    ac_request_special("element_user_data", "get_variable_value", "iTableMaterials",
str);
    cout << "-----\n";
    coutvar << str;
    ts_table(iTableMaterials, "load_from_json_str", str);
    ac_request_special("element_user_data", "get_variable_value",
"iTableSolutionsMats", str);
    ts_table(iTableSolutionsMats, "load_from_json_str", str);
    ac_request_special("element_user_data", "get_variable_value", "iTableSolutions",
str);
    ts_table(iTableSolutions, "load_from_json_str", str);
    ac_request_special("element_user_data", "get_variable_value", "iTableSysProps",
str);
    ts_table(iTableSysProps, "load_from_json_str", str);
}

```

## clear

```

:
int ires = ac_request_special("element_user_data","clear");

```

## delete\_variable

```

:
int ires = ac_request_special("element_user_data","delete_variable",string varname);
:
varname -

```

## get\_variable\_value

```

:
int err = ac_request_special("element_user_data","get_variable_value",string varname, void
&varnametovalue);
:
varname - ;
varnametovalue - ,
err - - 0.

```

## set\_variable\_value

```

:
ac_request_special("element_user_data","set_variable_value",string varname, void
&varvalue);
:
varname - , ;
varvalue - ,

```

## linkingElems

```

:
...
...
...
1 - , 2 -
...
1). , 3 (
...
( - 1 - 2). ,
,

```

## uplinkBiWardByFlags

```

/
/
:
int err = ac_request_special("linkingElems", "uplinkBiWardByFlags", int iGuid, int
bitvalue_from_to, bool bFromToOn, int bitvalue_to_from, bToFromOn, int iTableGuids);
:
iGuid - ts_guid, guid , /
/
bitvalue_from_to - ,

```

```

        ,
        iTableGuids
        ( bToFromOn==true)
( bFromToOn==false).
bitvalue_to_from -
        ,
        iTableGuids
        iGuid
        ( bFromToOn==true)
( bToFromOn==false);
iTableGuids -
        ts_table,
        guid
        /
        .

```

### getLinkedElemsByFlags

```

        guid
        ,
        :
        :
ac_request_special("linkingElems", "getLinkedElemsByFlags", iGuid, int bitvalue_from_to,
int iTable);
        :
iGuid -
        ts_guid,
bitvalue_from_to -
        ,
        :
iTable -
        ts_table,
        guid
        .

```

### set\_TSMAX\_ELEMLIST\_NUMBER

```

        10-
        ,
        :
        :
ac_request_special("set_TSMAX_ELEMLIST_NUMBER", int ineeded_count);
        :
ideeded_count -
        .
        = 100,
        .
int iicount;
ac_request_special("get_TSMAX_ELEMLIST_NUMBER", iicount);
if (iicount < 100)
{
    ac_request_special("set_TSMAX_ELEMLIST_NUMBER", 100);
}
coutvar << iicount;

```

### get\_TSMAX\_ELEMLIST\_NUMBER

```

        10-
        ,
        :
        :
ac_request_special("get_TSMAX_ELEMLIST_NUMBER", int &icurrent_lists_count);
        :
icurrent_lists_count -
        .

```

### set\_elemlistname

```

        .
        -
        ,
        .
        -
        .
        :

```



```
ac_request_special("set_elemlistname", int ilistnum, string slistnewname);

:
ilistnum -
slistnewname -
```

Get\_Elements\_Collisions

```
:
int ires = ac_request_special("Get_Elements_Collisions",int iList1, int iList2, int
iTable_res,
double volumeTolerance, bool performSurfaceCheck, double surfaceTolerance);

:
iList1, iList2 -
iTable_res -
```

GUID1		guid 1
TypeName1		1
ID1		ID 1
hasBodyCollision1	bool	
hasClearenceCollision1	bool	
GUID2		guid 2
TypeName2		2
ID2		ID 2
hasBodyCollision2	bool	
hasClearenceCollision2	bool	

```
double volumeTolerance -
bool performSurfaceCheck -
double surfaceTolerance -
```

delete\_elements\_from\_list

```
:
ac_request_special("delete_elements_from_list", int/string listindex/listname, int
start_index, int count);

:
listindex listname -
start_index -
count -
```

delete\_elements\_from\_project

```

:
int err = ac_request_special("delete_elements_from_project", int/string
listindex/listname);

```

```

:
listindex listname -
start_index -
count -

```

0

## ac\_request\_attributes

### Таблица типов атрибутов

"Pen"	1	
"Layer"	2	
"LineType"	3	
"FillType"	4	
"CompWall"	5	
"Material"	6	
"City"	7	
"LayerComb"	8	
"ZoneCat"	9	
"Font"	10	
"Profile"	11	
"PenTable"	12	
"DimStand"	13	
"ModelViewOptions"	14	
"MepSystem"	15	MEP
"OperationProfile"	16	
"BuildingMaterial"	17	
"MarkUpStyle"	18	

### create\_attr

```

( , , . .)

```

```

:
int err = ac_request_attributes("create_attr", int/string attrtypeid, string name, string
id, int/string baseattr);

```

:

```

attrtypeid - _____;
name - _____;
id - id
baseattr - _____, _____

```

## get\_num

## delete\_attr

```

:
int ires = ac_request_attributes("delete_attr", int/string attrtypeid, int attr_index);

```

```

:
attrtypeid - _____;
attr_index - _____

```

```

0

```

## set\_attr\_value

```

:
int ires = ac_request_attributes("set_attr_value", int/string attrtype, int
attrindex/string attrname, int/string genesis, string/bool/int/double value);

```

```

:
attrtypeid - _____;
attrindex attrname - _____;
genesis - _____;
value - _____

```

```

0

```

## get\_attr\_value

```

:
int ires = ac_request_attributes("get_attr_value", int/string attrtype, int
attrindex/string attrname, string/int genesis, string/bool/int/double value);

```

```

:
attrtypeid - _____;
attrindex attrname - _____;
genesis - _____(____);
ac_getnumvalue() ac_getstrvalue().

```

```

0

```

```

description

```

```

int building_material_index = 12;
ac_request_attributes("get_attr_value", "BuildingMaterial", building_material_index,
"AP", "description");
string description = ac_getstrvalue();
coutvar << description;

```

## ac\_typeidfromstring

---

```

:
int itype = ac_typeidfromstring(string selemtypename);

:
selemtypename - ("ZoneType", "MeshType", "ObjectType" . .)

```

## ac\_getresvaluetype

---

```

:
string stype = ac_getresvaluetype();

:
stype "String"/"Real"/"Integer"/"Boolean" -
1/0.

```

## ac\_getstrvalue

---

```

:
string sresult = ac_getstrvalue();

```

## ac\_getnumvalue

---

```

:
double dresult = ac_getnumvalue();

```

## ac\_getupvaluestatus

---

```

:
Null ( ), Undefined ( ) Normal.
0
,
- 0 -
Null Undefined -
Null Undefined, . . 0 1 Null.
2 - . .
Null.

:
int ires = ac_getupvaluestatus();

:
ires - :
0 - (Null);
1 - (Undefined);
2 - (Normal);

```

Null.

```
ac_request_special("get_element_value", "UP", "Переменные водосточной системы\\Координаты
элементов водосточной системы");
string svalue = ac_getstrvalue();
int ivalue = ac_getupvaluestatus();
if(ivalue == 2) {
    cout << "Свойство имеет нормальное значение";
}
else
{
    cout << "Свойство не имеет присвоенного значения или является Null";
}
```

## Гравитация на поверхность

### do\_elements\_landing - приземление элементов

```
ac_request("do_elements_landing", int iLandList, int iLandingElemsList, double doffset,
int mmode);
```

```

:
iLandList - ,
iLandingElemsList - ,
doffset - ,
mmode - 0 - , 1 - , 2 - , 3 -
```

### do\_surface\_landing - приземление по точкам

```

3d " "
" "
:
ac_request("do_surface_landing", int iLandList, int iLandingElemsList, double doffset);
```

```

:
iLandList - ,
iLandingElemsList - ,
doffset - ,
```

### do\_point\_landing - приземление точки X,Y

```

X Y Z ,
iLandList. " "
:
ac_request("do_point_landing", int iLandList, double doffset, double x1, double y1, double
&z1);
```

```

:
x1,y1 -
doffset -
z1 -
iLandList -

1 "My land" x=1, y=1
0, ..

ac_request("load_elements_list,1,"ZombieElemType","Layer","My land","MainFilter",3);

double doffset=0;
double x=1, y=1, z=0;
ac_request("do_point_landing", 1, doffset, x,y,z);
cout << "Результат - координата z на поверхности = "<< z;

```

### 3d сетки

#### get\_lands\_combined\_contour\_coords

```

2d ARCHICAD 3d
, , , ( ..
).
:
ac_request("get_lands_combined_contour_coords", int iElementsList, int iTablePointsXY_S);

:
iElementsList - ;
iTablePointsXY_S - ;

: "x","y" - 2d ;
"s" - ( ).

3d
.

```

```

int iTable_Coords2DTableContour;
object("create", "ts_table", iTable_Coords2DTableContour);

ac_request_special("load_elements_list_from_selection", 1, "MeshType", 2);

ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue();
cout << "Отобрано элементов для анализа = " << icount << "\n";

ac_request("get_lands_combined_contour_coords", 1, iTable_Coords2DTableContour);
string str;
ts_table(iTable_Coords2DTableContour, "print_to_str", str);
cout << "Контур\n";
coutvar << str;

```

#### do\_create\_contour\_around\_points

Создать контур вокруг набора точек.

Вызов:

```

ac_request("do_create_contour_around_points", int iTablePointsXYZ, int iTablePointsXY_S,
bool bJustSquare, double& dMinZ);

```

Здесь:

iTablePointsXYZ - дескриптор таблицы точек, вокруг которых сделать контур;  
iTablePointsXY\_S - дескриптор таблицы, куда записать результат;  
bJustSquare - если true, то контур - габаритный прямоугольник, если false - контур - наименьший выпуклый многоугольник.  
dMinZ - возвращается минимальная высота среди предоставленных точек;

## get\_lands\_topcoords

```

        :
        :
        :
ac_request("get_lands_topcoords", int iElementsList, int iTablePointsXYZ_s_polynum, int
iMode);

        :
iElementsList -
iTablePointsXYZ_s_polynum -
        :
"x","y","z" - 3d
"s" - (
"polynum" -
iMode - - 0/1/2/3.
0 - "polynum"
1 -
2 -
3 -
        iMode 0,2 3 3d
        1
ac_request_special("load_elements_list_from_selection", 1, "MeshType", 2);
int iTable_Coords3DTable;
object("create", "ts_table", iTable_Coords3DTable);
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue();
cout << "Отобрано элементов для анализа = " << icount << "\n";
string str;
ac_request("get_lands_topcoords", 1, iTable_Coords3DTable, 1);
ts_table(iTable_Coords3DTable, "print_to_str", str);
cout << "\nКоординаты\n";
coutvar << str;

```

## do\_create\_mesh\_by\_points\_and\_lines

```

        3d mesh (3d      )      /
        :
ac_request("do_create_mesh_by_points_and_lines",  int  iTTable_Coords3DPointsTable, int
iTTable_Coords2DTableContour, int  iTTable_Coords3DPointsTable);

```

Здесь:

iTable\_Coords3DPointsTable - дескриптор таблицы, описывающей 3d точки уровня поверхности (может быть -1, если уровни заданы только линиями);

iTable\_Coords2DTableContour - дескриптор таблицы, описывающей 2d точки контура. Если указать -1 - то контуром будут точки будут рассчитаны автоматически по таблицам точек уровня и линий;

iTable\_Coords3DPointsTable - дескриптор таблицы, описывающей 3d линии уровней (может быть -1, если уровни задаются только точками).

Пример.

Построить общую поверхность для набора выделенных элементов в виде 3d сетки, у которой координаты уровней заданы точками.

```
int iTable_Coords3DTable;
object("create", "ts_table", iTable_Coords3DTable);

int iTable_Coords2DTableContour;
object("create", "ts_table", iTable_Coords2DTableContour);

ac_request_special("load_elements_list_from_selection", 1, "MeshType", 2);
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue();
cout << "Отобрано элементов для анализа = " << icount << "\n";
ac_request("get_lands_combined_contour_coords", 1, iTable_Coords2DTableContour);
string str;
ts_table(iTable_Coords2DTableContour, "print_to_str", str);
cout << "Контур\n";
coutvar << str;
ac_request("get_lands_topcoords", 1, iTable_Coords3DTable, 1);
ts_table(iTable_Coords3DTable, "print_to_str", str);
cout << "\nКоординаты\n";
coutvar << str;
ac_request("do_create_mesh_by_points_and_lines", iTable_Coords3DTable,
iTable_Coords2DTableContour, -1);

object("delete", iTable_Coords2DTableContour);
object("delete", iTable_Coords3DTable);
```

## Диалоги

### Сообщения или выбор варианта - tsalert

```
int res = tsalert(int messagecode, string stitle, string smessagebig, string ssmall, string
button1, string button2, string button3);
```

```

:
messagecode -
-1 -
-2 -
-3 -
stitle - , smessagebig - , ssmall -
, button1 - button3 -
res
button1 - 1, button2 - 2, button3 - 3.
- 0.
```

(10, 1 ).



```

int res = tsalert(-3, "Задайте значение", "Сколько выводить элементов?", "Для пробной
выгрузки удобно вывести сначала не все маркеры", "10", "1", "Все");
if (res == 0)
{
    cout << "Пользователь отказался";
    return;
}

```

## Поиск и выбор файла - dialog\_get\_filename

```

ac_request dialog_get_filename
:
int iret = ac_request("dialog_get_filename", string Title, string Filter, string
sStartFolder, string for_what, string &FileNameAndPath);

: Title - , Filter - , sStartFolder -
FileNameAndPath.
for_what - "OpenFile" "Save".
iret -1 - ,
.cpp. C:.

string sFileNameAndPath;
string sStartFolder = "c:\\";
int iret = ac_request("dialog_get_filename", "Выберите файл для запуска", "cpp",
sStartFolder, "OpenFile", sFileNameAndPath);
if (iret == -1)
{
    cout << "Пользователь отказался от выбора файла\n";
    return -1;
}
run_cpp("run_from_file", sFileNameAndPath);
cout << "Файл выбран и выполнен";

```

Можно составлять сложные фильтры и группы фильтров по расширениям файлов.

```

1 - Excel xls,xslx, xslm;
2 - Archicad pln, pla, tpl;
3 - Text txt, sx.
;

string FileNameAndPath;
ac_request("dialog_get_filename", "Выберите файл", "Excel,xls,xlsx,xlsm|
Archicad,pln,pla,tpl|Text,txt,sx", "", "OpenFile", FileNameAndPath);
coutvar << FileNameAndPath;

```

## editdoubledialog - ввод числа

```

ac_request() "editdoubledialog"
:
int res = ac_request("editdoubledialog", string smessage, string sstartvalue);
int res = ac_request("editdoubledialog", string smessage, double dstartvalue);

```

```

:
smessage - , sstartvalue dstartvalue - ,
0 ( " ",
"Ok" "Esc" ).
1.
.
20.

```

```
int res = ac_request("editdoubledialog","Введите дистанцию между элементами (м)","20");
```

## editintdialog - ввод целого числа

```

ac_request() "editdoubledialog"
:
int res = ac_request("editintdialog",string smessage, string sstartvalue);
int res = ac_request("editintdialog",string smessage, double dstartvalue);
:
smessage - , sstartvalue dstartvalue - ,
0 ( " ",
"Ok" "Esc" ).
1.
.
20.

```

### Вариант 1:

```
int res = ac_request("editintdialog","Введите количество элементов (шт)","20");
if(res == 1) {
    ivalue = ac_getnumvalue();
}
```

### Вариант 2:

```
int res = ac_request("editintdialog","Введите количество элементов (шт)", 20);
if(res == 1) {
    ivalue = ac_getnumvalue();
}
```

### Вариант 3:

```
int ivalue = 20;
int res = ac_request("editintdialog","Введите количество элементов (шт)",ivalue);
if(res == 1) {
    ivalue = ac_getnumvalue();
}
```

## editstringdialog - ввод строки

```

ac_request() "editstringdialog"
:
int res = ac_request("editstringdialog",string smessage, string sdefaultstr);
:
smessage - ,
sdefaultstr - ,
0 ( " ",

```

```

        "Ok"          "Esc"          ).
1.

```

```

string svalue = "Иванов В.В."
int res = ac_request("editstringdialog","Введите ФИО архитектора", svalue);
if(res == 1)
{
    svalue = ac_getstrvalue();
}
coutvar << svalue;

```

## editstringlistdialog - выбор строки из списка

```

        ac_request()          "editstringlistdialog"
popup.
:
int res = ac_request("editstringlistdialog",string smessage, int iTable, string sstr);
или
int res = ac_request("editstringlistdialog",string smessage, int iTable, int index);
или
int res = ac_request("editstringlistdialog",string smessage, string savailablelist,
string sselected, string sdelimiter);

:
smessage -
iTable -
,
str -
index -
,
popup
;
popup
- 0
popup. index
- 1.
0-
1-
popup.
ts_table
- 0
popup -
index
- 1.
0-
1-
popup.
savailablelist -
sselected -
sd delimiter -
.
0
"
",
"Esc"
).
"Ok"
1.
ac_getstrvalue() -
ac_getnumvalue()
1.
"
..
int iTable;
object("create","ts_table",iTable);
ts_table(iTable,"add_column",-1,"string","h1");
ts_table(iTable,"add_row",0,"Иванов В.В.");
ts_table(iTable,"add_row",0,"Петров В.В.");
ts_table(iTable,"add_row",0,"Сидоров В.В.");
string svalue = "Петров В.В.";
int index;
int res = ac_request("editstringlistdialog","Введите ФИО архитектора", iTable, svalue);
if(res == 1)
{
    svalue = ac_getstrvalue();
    index = ac_getnumvalue();
}

```

```

}
coutvar << svalue;
coutvar << index;
object("delete",iTable);

```

2

```

string s = "one;two;three";
string ss = "two";
int iires = ac_request("EditStringListDialog", "Тип линии", s, ss,"");
coutvar<<iires;
coutvar<< ss;
return;

```

## editcheckstringsdialog - выбор нескольких строк из списка

```

ac_request() "EditCheckStringsDialog"
.
:
int res = ac_request("EditCheckStringsDialog",string smessage, int iTable);

int res = ac_request("EditCheckStringsDialog", string smessage, string savailablelist,
string sselected, string sdelimiter);

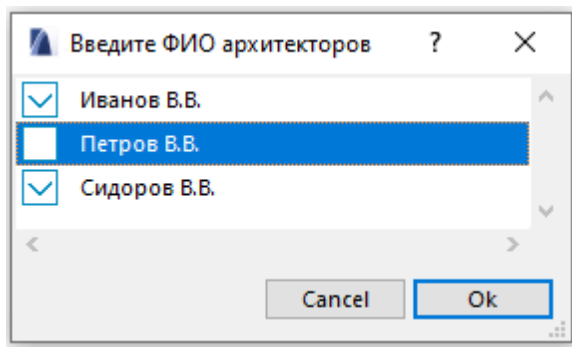
:
smessage -
iTable -
- bool / string -
savailablelist - список возможных вариантов через разделитель;
sselected - список помеченных вариантов через разделитель и возвращаемый набор значений,
так же через разделитель;
sd delimiter - разделитель.

0 ( " " ,
"Esc" ).
"Ok" 1.

.
/ .

object("create", "ts_table", iTable);
ts_table(iTable, "add_column", -1, "bool", "h1");
ts_table(iTable, "add_column", -1, "string", "h2");
ts_table(iTable, "add_row", 0, false, 1, "Иванов В.В.");
ts_table(iTable, "add_row", 0, true , 1, "Петров В.В.");
ts_table(iTable, "add_row", 0, false, 1, "Сидоров В.В.");
int res = ac_request("EditCheckStringsDialog", "Введите ФИО архитекторов", iTable);
if (res == 1)
{
string str;
ts_table(iTable, "print_to_str", str);
coutvar << str;
}
object("delete", iTable);

```



```
str=rowcount=3
0). h1=0 h2=Иванов В.В.
1). h1=1 h2=Петров В.В.
2). h1=0 h2=Сидоров В.В.
```

2.

```
string savailable = "1 значение;2 значение;3 значение";
string sselected = "2 значение;3 значение";
int res = ac_request("EditCheckStringsDialog", "Тип линии", savailable, sselected,"");
if (res == 1)
{
    coutvar << sselected;
}
```

## editbooldialog - выбор Да/Нет

```
ac_request() "editbooldialog"
- / ( / ).
:
int res = ac_request("editbooldialog", string smessage, bool bdefaultvalue);
:
smessage -
bdefaultvalue -
0
"Ok" "Esc" 1.
.
" ..".

bool bRes = false;
int res = ac_request("editbooldialog","Включать заголовки при экспорте данных?", bRes);
if(res == 1)
{
    bRes = ac_getnumvalue();
}
coutvar << bRes;
```

## editstringlistandcustomdialog - ввод строки или выбор из списка

```
ac_request() "EditStringListAndCustomDialog"
:
int res = ac_request("EditStringListAndCustomDialog",string smessage, int iTable, string
str);
или
int res = ac_request("EditStringListAndCustomDialog",string smessage, int iTable, int
index);
```

или

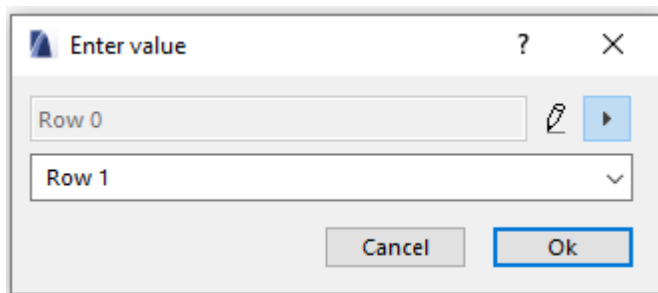
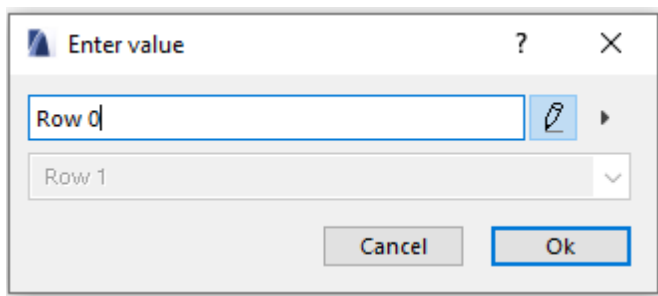
```
int res = ac_request("EditStringListAndCustomDialog", string smessage, string
savailablelist, string sselected, string sdelimiter);
```

```

:
smessage -
iTable -
0
string -
(
"Esc"
).
"Ok"
1.
str -
-
savailablelist -
;
sselected -
;
sd delimiter -
.
index -
;
ac_getstrvalue() -
ac_getnumvalue()
0 -
iTable savailablelist;
```

```

int iTable;
object("create", "ts_table", iTable);
ts_table(iTable, "add_column", -1, "string", "col");
ts_table(iTable, "add_row", 0, "Row 1");
ts_table(iTable, "add_row", 0, "Row 2");
ts_table(iTable, "add_row", 0, "Row 3");
bool bres = ac_request("EditStringListAndCustomDialog", "Enter value", iTable, "Row 0");
if(bres)
{
    string sres = ac_getstrvalue();
    int ires = ac_getnumvalue();
    coutvar << sres;
    coutvar << ires;
    object("delete", iTable);
}
}
```



2

```

string s = "one;two;three";
string ss = "two";
int iires = ac_request("EditStringListAndCustomDialog", "Тип линии", s, ss,"");
coutvar<<iires;
coutvar<< ss;
return;

```

## Работа с Excel

---

### excel\_attach

```

Excel
Excel
Excel.
:
int ires = excel_attach();
0,
Excel
100.1.

int my_variable;
coutvar << my_variable;
int ires = excel_attach();
if (res != 0)
{
    cout << "Нет связи с Excel";
    return -1;
}

double dvalue = 100.1;
excel_putnumvalue(dvalue);
excel_detach();

```

### excel\_detach

```

Excel.

excel_detach();

```

### excel\_putnumvalue

```

Excel
:
excel_putnumvalue(double dvalue);

```

### excel\_putstrvalue

```

Excel
:
excel_putstrvalue(string svalue);

```

### excel\_select\_range

```

Excel
:

```

```
excel_select_range( string srange);
: srange - Excel.
```

```
Excel "A2:C4".
excel_select_range( "A2:C4");
```

## **excel\_visible**

```
Excel .
excel_visible();
```

## **excel\_speedup**

```
/ Excel.
.
:
excel_speedup(int what);
what - 0/1 - / Excel.
```

## **excel\_getnumvalue**

```
Excel .
double dvalue = excel_getnumvalue();
```

## **excel\_getstrvalue**

```
Excel .
string svalue = excel_getstrvalue();
```

## **excel\_request**

### **set\_column\_width**

```

:
excel_request("set_column_width",string sdiapazon, double width);
: sdiapazon - , : "B:B" -
B. "B:D" B,C D.
width - .
B 18.86:
excel_request("set_column_width", "B:B",18.86);
```

### **get\_column\_width**

```

:
excel_request("get_column_width", string sdiapazon, double width);
: sdiapazon - , : "B:B" -
B. "B:D" B,C D.
width -
```

### **set\_row\_height**



```

:
excel_request("set_row_height", string srowdiapazon, double height);

: srowdiapazon -
: "1:1" - , "2:4" -
height -
:
: 2- 8- 15.75.

excel_request("set_row_height", "2:8", 15.75);

```

## get\_row\_height

```

:
excel_request("get_row_height", string srowdiapazon, double height);

: srowdiapazon -
: "1:1" - , "2:4" -
height -
:

```

## set\_borders

```

:
excel_request("set_borders", int border_left, int border_top, int border_right, int
border_bottom);

: border_left, border_top, border_right, border_bottom -
: , , , . 1- , 0-
:

excel_request("set_borders");

```

## put\_selection\_values

```

:
excel_request("put_selection_values", string varname1, string/int/double/bool
varvalue1,..., string varnamen, string/int/double/bool varvaluen);

```

```

:
varname1 - varnamen -
Excel Range Excel VBA.
varvalue1-varvaluen -

```

```

1:
Excel.

excel_request("put_selection_values", "Width", 10, "Height", 20);

```

```

2:
Excel.

excel_request("put_selection_values", "MergeCells", 1);

```

## put\_selection\_fontvalues

Excel ( ).

```

:
excel_request("put_selection_fontvalues", string fontparname1, string/double/int/bool
fontparvalue1, ... , string fontparnamen, string/double/int/bool/ fontparvaluen);

```

```

:
fontparname1-fontparnamen - , Excel (
Excel VBA).
fontparvalue1-fontparvaluen - .

```

```

:
Excel 10.32,
excel_request("put_selection_fontvalues", "Size", 10.32, "Bold", 1, "Italic", 1);

```

## get\_selection\_area

```

Excel.
:
excel_request("get_selection_area", int start_row, int start_column, int end_row, int
end_column);
:
start_row, start_column - Excel,
end_row, end_column - Excel.
.
Excel

```

```

int start_row, start_column, end_row, end_column, rows, columns;
excel_request("get_selection_area", start_row, start_column, end_row, end_column);

```

## merged\_cell\_info

```

:
excel_request("merged_cell_info", int &merged_start_row, int &merged_start_col, int
&merged_rows, int &merged_cols);

```

Здесь:  
merged\_start\_row, merged\_start\_col - номер строки и колонки левого верхнего угла  
объединенной области в которую входит текущая ячейка,  
merged\_rows - число строк, которые объединены в область,  
merged\_cols - число колонок, которые объединены в область.

## is\_merge\_cells

```

Excel.
:
excel_request("is_merge_cells", int &ismerged);
:
ismerged - Excel , ismerged
1.

```

**set\_backcolor****get\_backcolor****set\_interior****get\_interior****selection\_varvalues****selection\_font\_varvalues****sheet\_select****range\_copy****booknamedcell****save\_active\_sheet\_as\_csv**

```

                                CSV
:
int ires = excel_request("save_active_sheet_as_csv", string filename);
:
filename -                               Excel           CSV.
.
                                "
                                .xlsm" (
                                Excel).
,
.
                                "h:\1.xlsx".
Excel.
                                "1_
                                .csv".
                                "h:\1.xlsx",

int main()
{
    excel_attach();
    bool bexist;
    int ires;

    string sWorkbookUprFile = "Управляющий файл.xlsm";
    ires = excel_request("workbook_select", sWorkbookUprFile);
    if (ires != 0)
    {
        tsalert(-1, "Ошибка во время выполнения", "Не получается переключиться в
файл excel", sWorkbookUprFile);
        excel_detach();
        return -1;
    }

    string filename = "h:\\1.tmp";
    ires = shell_func("file", "is_exist", filename, bexist);
    if (bexist)
    {
        cout << "Удаляем файл " << filename << "\n";

```

```

        ires = shell_func("file", "delete", filename);
    }

    excel_request("save_copy_as", filename);
    excel_request("workbook_open", filename);
    ires = save_sheet_as_csv_file("Пироги стен");
    if (ires != 0)
    {
        return;
    }
    ires = save_sheet_as_csv_file("Пироги колонн");
    ires = save_sheet_as_csv_file("Пироги балок");
    ires = save_sheet_as_csv_file("Пироги потолков");
    ires = save_sheet_as_csv_file("Пироги низа и торца лестниц");
    ires = save_sheet_as_csv_file("Пироги полов");
    ires = save_sheet_as_csv_file("Плнтуса и торцы фундам. оборуд");
    ires = save_sheet_as_csv_file("Узлы");
    ires = save_sheet_as_csv_file("Слои пирогов отделки");
    ires = save_sheet_as_csv_file("Слои пирогов полов");
    excel_request("workbook_close");

    excel_detach();
}

int save_sheet_as_csv_file(string sSheetName)
{
    bool bexist;

    int ires = excel_request("sheet_select", sSheetName);
    if (ires != 0) {
        tsalert(-1, "Ошибка во время выполнения", "Не получается переключиться на
страницу", sSheetName);
        excel_detach();
        return -1;
    }

    string filename;

    filename = "h:\\1_" + sSheetName + ".csv";
    ires = shell_func("file", "is_exist", filename, bexist);
    if (bexist)
    {
        cout << "Удаляем файл " << filename << "\n";
        ires = shell_func("file", "delete", filename);
    }
    ires = excel_request("save_active_sheet_as_csv", filename);
    return ires;
}

```

## save\_copy\_as

Сохранить копию текущего файла Excel без его открытия в Excel.

Вызов:

```
int ires = excel_request("save_copy_as", string filename);
```

Здесь:

filename - имя и полный путь к файлу, куда надо сохранить копию текущего файла Excel.

## workbook\_open

Excel.

```

:
int ires = excel_request("workbook_open", string filename);

```

```

:
filename - , Excel.

```

## workbook\_close

Excel.

```

:
int ires = excel_request("workbook_close");

```

## работа с примечаниями

### getcomment

```

:
excel_request("comment", "get", string &scomment);

```

```

: scomment -

```

```

string scomment;
excel_request("comment", "get", scomment);
cout << scomment;

```

### clearcomments

```

:
excel_request("comment", "clear");

```

### setcomment

```

:
excel_request("comment", "set", string scomment);

```

```

: scomment -

```

```

string scomment = "Это комментарий к ячейке\n в две строки";
excel_request("comment", "set", scomment);

```

## Загрузка и выгрузка табличных данных

### excel\_chart - диаграммы в Excel

#### AddChart

Excel.

```

excel_chart(int iChartDescr, "AddChart");

```

Здесь:  
iChartDescr - дескриптор объекта excel\_chart. Сюда возвращается элемент управления созданной диаграммой.

#### SetChartType

Excel.

```
excel_chart(int iChartDescr, "SetChartType", int itype);
```

itype - Excel.

Имя типа диаграммы	Код типа	Описание
xl3DArea	-4098	Трехмерная область.
xl3DAreaStacked	78	Трехмерная область с накоплением.
xl3DAreaStacked100	79	100% с областями с накоплением.
xl3DBarClustered	60	Трехмерная кластерная панель.
xl3DBarStacked	61	Трехмерная линейчатая диаграмма с накоплением.
xl3DBarStacked100	62	Трехмерная линейчатая диаграмма с накоплением 100.
xl3DColumn	-4100	Трехмерный столбец.
xl3DColumnClustered	54	Трехмерный кластеризованный столбец.
xl3DColumnStacked	55	Объемный столбец с накоплением.
xl3DColumnStacked100	56	Трехмерная гистограмма с накоплением 100%.
xl3DLine	-4101	Трехмерная линия.
xl3DPie	-4102	Трехмерная круговая диаграмма.
xl3DPieExploded	70	Развернутая трехмерная круговая диаграмма.
xlArea	1	Область
xlAreaStacked	76	Область с накоплением.
xlAreaStacked100	77	100% с областями с накоплением.
xlBarClustered	57	Панель кластера.
xlBarOfPie	71	Вторичная гистограмма.
xlBarStacked	58	Линейчатая диаграмма с накоплением.
xlBarStacked100	59	100% линейчатой диаграммы с накоплением.
xlBubble	15	Плыв.
xlBubble3DEffect	87	Пузырьковая с трехмерными эффектами.
xlColumnClustered	51	Кластеризованный столбец.
xlColumnStacked	52	Гистограмма с накоплением.
xlColumnStacked100	53	100% гистограммы с накоплением.
xlConeBarClustered	102	Кластеризованная коническая с полосой.
xlConeBarStacked	103	Коническая с накоплением.
xlConeBarStacked100	104	100% нормированной конусной полосы.
xlConeCol	105	Столбец объемных конусов.
xlConeColClustered	99	Кластеризованный конусный столбец.
xlConeColStacked	100	Конусный столбец с накоплением.
xlConeColStacked100	101	100% конусный столбец с накоплением.
xlCylinderBarClustered	95	Кластеризованная цилиндрическая панель.
xlCylinderBarStacked	96	Вертикальная полоса цилиндра.
xlCylinderBarStacked100	97	100% нормированной цилиндрической полосы.
xlCylinderCol	98	Столбец 3D-цилиндра.
xlCylinderColClustered	92	Кластеризованный конусный столбец.
xlCylinderColStacked	93	Конусный столбец с накоплением.
xlCylinderColStacked100	94	100% нормированный столбец цилиндра.
xlDoughnut	-4120	Сектор.
xlDoughnutExploded	80	Разрезанная кольцевая диаграмма.
xlLine	4	Междустрочный.

xlLineMarkers	65	Строка с маркерами.
xlLineMarkersStacked	66	Линия с накоплением с маркерами.
xlLineMarkersStacked100	67	100% графика с накоплением с маркерами.
xlLineStacked	63	График с накоплением.
xlLineStacked100	64	100% графика с накоплением.
xlPie	5	Кругов.
xlPieExploded	69	Разрезанная круговая диаграмма.
xlPieOfPie	68	Вторичная круговая диаграмма.
xlPyramidBarClustered	109	Кластеризованная пирамида.
xlPyramidBarStacked	110	Пирамидальная диаграмма с накоплением.
xlPyramidBarStacked100	111	100% пирамидальной диаграммы с накоплением.
xlPyramidCol	112	Столбец 3D пирамиды.
xlPyramidColClustered	106	Кластеризованный столбец пирамиды.
xlPyramidColStacked	107	Столбец пирамиды с накоплением.
xlPyramidColStacked100	108	100% столбца пирамиды с накоплением.
xlRadar	-4151	Лепестковые.
xlRadarFilled	82	Заполненная лепестковая диаграмма.
xlRadarMarkers	81	Лепестковая диаграмма с маркерами данных.
xlRegionMap	140	Диаграмма карты.
xlStockHLC	88	Высокая — минимум — закрытие.
xlStockOHLC	89	Открытие — высокая — минимум — закрытие.
xlStockVHLC	90	Громкость — высокая — минимальная — закрытие.
xlStockVOHLC	91	Volume — Open — High — минимум — Close.
xlSurface	83	Трехмерная поверхность.
xlSurfaceTopView	85	Поверхность (вид сверху).
xlSurfaceTopViewWireframe	86	Поверхность (каркас верхнего уровня представления).
xlSurfaceWireframe	84	Трехмерная поверхность (каркасная схема).
xlXYScatter	-4169	Рабочие.
xlXYScatterLines	74	Точечная диаграмма со строками.
xlXYScatterLinesNoMarkers	75	Точечная диаграмма со строками без маркеров данных.
xlXYScatterSmooth	72	Точечная диаграмма с сглаженными линиями.
xlXYScatterSmoothNoMarkers	73	Точечная диаграмма со сглаженными линиями без маркеров данных.

## SetChartTitle

```
excel_chart(int iChartDescr, "SetChartTitle", string sTitle);
```

```

:
iChartDescr - excel_chart,
sTitle -

```

## SetSourceDataRange

Excel

```

:
excel_chart(int iChartDescr, "SetSourceDataRange");

```

```

:
iChartDescr - Excel

```

## Пример

```
//-----
// LABPP
// Создать диаграмму в Excel по данным в ячейках C7:F8
//-----
int main()
{
    excel_attach(); // подключить Excel

    int iChartDescr;
    object("create","excel_chart", iChartDescr); // создать объект
    excel_chart(iChartDescr,"AddChart"); // создать диаграмму на текущей странице таблицы
Excel
    excel_chart(iChartDescr,"SetChartType",5); // задать тип диаграммы
    excel_chart(iChartDescr,"SetChartTitle","Квартирография"); // задать заголовок

    excel_select_range("C7:F8"); // выбрать ячейки в таблице Excel
    excel_chart(iChartDescr,"SetSourceDataRange"); // задать эти ячейки как источник
данных диаграммы
    object("delete",iChartDescr); // удалить объект

    excel_detach(); // отключить Excel
}
```

## Работа с Word

### word\_attach

```

Word
:
int ires = word_attach();
0,
```

### word\_detach

```

Word.
:
word_detach();
```

### word\_visible

```

Word. Word
, ...
, -
:
word_visible(int invisible);
: invisible - 0/1 - / Word.
```

### word\_request

```

Word
ARCHICAD, Word
.
```



## НСчитывание и запись полей переменных - docfield

### get

```

                                Word
int ires = word_request("docfield","get", string wordfielsname, string/double/int/bool
value);
:
wordfielsname -                Word,
string/double/int/bool value -
Word.
                                0
                                .

```

### set

```

                                Word.
:
int ires = word_request("docfield","set",string wordfielsname, string/double/int/bool
value);
:
wordfielsname -                Word,
string/double/int/bool value -
Word.
                                0
                                .

```

### update\_all\_docfields

```

                                Word,
:
word_request("update_all_docfields");

```

## Работа через ts\_idispatch

```

:
                                Word
                                ,
                                .

//-----
// LABPP Word пишем текст в Word и меняем шрифт
// LABPP 2023
//-----
int main()
{
    int iIDispatch_Word;
    object("create", "ts_idispatch", iIDispatch_Word);
    int ires = ts_idispatch(iIDispatch_Word, "attach", "Word.Application"); // create
dispatcher object
    if (ires != 0) {
        cout << "\nError whan attach to word";
        return -1;
    }

    int iVariant_Selection;
    object("create", "ts_variant", iVariant_Selection);
    ires = ts_idispatch(iIDispatch_Word, "AutoWrap", iVariant_Selection,
"DISPATCH_PROPERTYGET", "Selection", 0);

    int iIDispatch_Selection;
    object("create", "ts_idispatch", iIDispatch_Selection);
    ts_variant(iVariant_Selection, "get_pdispVal", iIDispatch_Selection);

    // Для информации об объекте-диспетчере, какие у него есть переменные и функции (не

```

```

обязательно) ---->
// получить список методов через перенос строки (\n)
string smethodslist;
ires = ts_idispatch(iIDispatch_Selection, "GetIDispatchMethods", smethodslist);
coutvar << smethodslist;
//<-----

int iVariant_Null;
object("create", "ts_variant", iVariant_Null);

int iVariant_Text;
object("create", "ts_variant", iVariant_Text);
ts_variant(iVariant_Text, "set_value_simple", "Привет");
ires = ts_idispatch(iIDispatch_Selection, "AutoWrap", iVariant_Null,
"DISPATCH_METHOD", "TypeText", 1, iVariant_Text);

int iVariant_Font;
object("create", "ts_variant", iVariant_Font);
ires = ts_idispatch(iIDispatch_Selection, "AutoWrap", iVariant_Font,
"DISPATCH_PROPERTYGET", "Font", 0);

int iIDispatch_Font;
object("create", "ts_idispatch", iIDispatch_Font);
ts_variant(iVariant_Font, "get_pdispVal", iIDispatch_Font);

ts_variant(iVariant_Text, "set_value_simple", "GOST 2.304 type A");
ires = ts_idispatch(iIDispatch_Font, "AutoWrap", iVariant_Null,
"DISPATCH_PROPERTYPUT", "Name", 1, iVariant_Text);

ts_variant(iVariant_Text, "set_value_simple", "Привет другим шрифтом");
ires = ts_idispatch(iIDispatch_Selection, "AutoWrap", iVariant_Null,
"DISPATCH_METHOD", "TypeText", 1, iVariant_Text);

ts_idispatch(iIDispatch_Word, "detach");

object("delete", iIDispatch_Word);
object("delete", iIDispatch_Selection);
object("delete", iVariant_Selection);
object("delete", iVariant_Text);
object("delete", iVariant_Null);
object("delete", iVariant_Font);
object("delete", iIDispatch_Font);
}

```

## Работа с AutoCAD

---

### autocad\_request

#### attach

AutoCAD.

```

:
int ires = autocad_request("attach");

0

:

int ires;
ires = autocad_request("attach");
if(ires != 0)
{

```

```

    cout << "Can't attach AutoCAD\n";
    return -1;
}

```

## detach

```

        AutoCAD.
    :
autocad_request("detach");

```

## specify\_color\_RGB

```

autocad_request("specify_color_RGB", int red, int green, int blue);

    red, green    blue -                RGB    0    255.

    :
autocad_request("specify_color_RGB", 255, 255, 255);

```

## draw\_text

```

autocad_request("draw_text", string stext, int posx, int posy, int textsize);

    stext -
posx posy -
textsize -

    "123".
string stext = "123";
autocad_request("draw_text", stext, 1.2, -2.3, 0.2);

```

## draw\_polyline

```

autocad_request("draw_polyline", int iCoordTableDescr);

    iCoordTableDescr -                (                ).

    :
double x, double y, double b
    x,y -                , b -                (                - 0).

    1 - (0;0),(1;0),(1;1),(0,1),(0;0).

int iTable;
object("create","ts_table",iTable);
ts_table(iTable,"add_column",0,"double","x",0);
ts_table(iTable,"add_column",1,"double","y",0);
ts_table(iTable,"add_column",2,"double","b",0);

ts_table(iTable,"add_row","x",0,"y",0,"b",0);
ts_table(iTable,"add_row","x",1,"y",0,"b",0);
ts_table(iTable,"add_row","x",1,"y",1,"b",0);
ts_table(iTable,"add_row","x",0,"y",1,"b",0);
ts_table(iTable,"add_row","x",0,"y",0,"b",0);

autocad_request("draw_polyline", iTable);
object("delete",iTable);

```

## draw\_line

Нарисовать линию в AutoCAD.

Вызов:

```
ires = autocad_drawline( double x1, double y1, double x2, double y2);
```

```
      :  
x1, y1, x2, y2 - .
```

## draw\_arc

AutoCAD.

```
      :  
int ires = autocad_drawarc( double centerX, double centerY, double radius, double  
startAngle, double endAngle);
```

```
      :  
centerX, centerY - ,  
radius - ;  
startAngle - :  
endAngle - .
```

## Пример - нарисовать контуры зон из ARCHICAD в AutoCAD полилиниями

```
//-----  
// LABPP  
// Нарисовать зоны полилиниями в AutoCAD  
// Draws Zone elements in AutoCAD by PolyLines  
//-----  
int main()  
{  
    int ires;  
    ires = autocad_request("attach");  
    if(ires != 0)  
    {  
        cout << "Can't attach AutoCAD\n";  
        return -1;  
    }  
    int iTable;  
    object("create", "ts_table", iTable);  
    ts_table(iTable, "add_column", 0, "double", "x", 0);  
    ts_table(iTable, "add_column", 1, "double", "y", 0);  
    ts_table(iTable, "add_column", 2, "double", "b", 0);  
  
    ac_request("load_elements_list_from_selection", 0, "ZoneType");  
    ires = ac_request("get_loaded_elements_list_count", 0);  
    int icount = ac_getnumvalue();  
    if(icount == 0)  
    {  
        cout << "You have not selected a Zone element.";  
        return -1;  
    }  
  
    int rowcount;  
    int iCoordTable;  
    object("create", "ts_table", iCoordTable);  
    int i, j;  
    double xvalue, yvalue, si;  
    double pi = 3.1415926535;  
    int itablerowcount;  
    runtimecontrol("workline", "setmax", icount);  
    for(i=0; i<icount; i++)
```

```

{
runtimecontrol("workline", "setpos", i+1);
ac_request("set_current_element_from_list",0,i);

ac_request("get_gdlelem_property_value","ZoneName");
string szonename = ac_getstrvalue();
cout << "zonename=" << szonename << "\n";
ac_request("get_gdlelem_property_value","ROOM_AREA");
double ra = tsround(ac_getnumvalue(),2);
cout << "room_area=" << ra << "\n";
ac_request("get_gdlelem_property_value","pos.x");
double posx = ac_getnumvalue();
ac_request("get_gdlelem_property_value","pos.y");
double posy = ac_getnumvalue();
string stext = szonename;
autocad_request("specify_color_RGB",0,254,0);
autocad_request("draw_text",stext,posx-0.5,posy,0.2);
stext = "(" + sprintf("%g",ra) + ")";
autocad_request("draw_text",stext,posx-0.5,posy-0.3,0.2);
autocad_request("specify_color_RGB",254,254,0);

ts_table(iTable, "clear_rows");
ac_request("get_element_value","CoordTable",iCoordTable);

ts_table(iCoordTable, "get_rows_count", rowcount);
double lastxvalue, lastyvalue, lastsi;
double radius, firstangle;
double startangle, endangle;
double centerx,centery;
double xfornext, yfornext;
double tg_of_quatro_angle;
for(j=0;j<rowcount;j++)
{
ts_table(iCoordTable,"select_row", j);
ts_table(iCoordTable,"get_value_of","x",xvalue);
ts_table(iCoordTable,"get_value_of","y",yvalue);
ts_table(iCoordTable,"get_value_of","si",si);
//cout << "x="<<xvalue<< ", y="<< yvalue << ", si="<< si << "\n";

if(j!=0)
{
if((si!=900)&&(si!=4000))
{
if(si==-1)
{
ts_table(iTable,"add_row",0,xvalue,1,yvalue,2,0);
autocad_request("draw_polyline",iTable);
ts_table(iTable, "clear_rows");
}
else {
ts_table(iTable,"add_row",0,xvalue,1,yvalue,2,0);}
}
else if(si==900)
{
ac_request("geometry_calc_2d","get_length_2point",lastxvalue, lastyvalue,
xvalue, yvalue, radius);
ac_request("geometry_calc_2d","get_line_angle",lastxvalue, lastyvalue,
xvalue, yvalue, firstangle);
centerx = xvalue;
centery = yvalue;
//cout << "radius=" << radius << " , firstangle="<< firstangle << "\n";
}
else if(si==4000)
{

```

```

        //cout << "yvalue=" << yvalue << "\n";
        if(yvalue > 0)
        {
            startangle = PI - (xvalue*pi/180.0 - firstangle);
            endangle = PI - (-yvalue*pi/180.0 - firstangle);
        }
        else
        {
            endangle = PI - (xvalue*pi/180.0 - firstangle);
            startangle = PI - (-yvalue*pi/180.0 - firstangle);
        }
        ac_request("geometry_calc_2d","rotate_point_and_move",centerx, centery,
lastxvalue,lastyvalue, endangle, centerx, centery, 1, xfornext, yfornext);
        tg_of_quatro_angle = tg((yvalue*pi/180.0)/4);
        ts_table(iTable,"get_rows_count", itablerowcount);
        ts_table(iTable,"select_row", itablerowcount-1);
        ts_table(iTable,"set_value_of",2,tg_of_quatro_angle);
        //cout << "tg=" << tg_of_quatro_angle << "\n";

        xvalue = xfornext;
        yvalue = yfornext;
    }
}
else {
    ts_table(iTable,"add_row",0,xvalue,1,yvalue,2,0); }

    lastxvalue = xvalue;
    lastyvalue = yvalue;
    lastsi = si;
}

ts_table(iTable,"get_rows_count", itablerowcount);
if(itablerowcount>0) {
    autocad_request("draw_polyline",iTable);}
}

object("delete",iCoordTable);
object("delete",iTable);

autocad_request("detach");
return 0;
}

```

## Функция object

---

### create

```
int object("create", string objclass, int descr);
```

Создает объект типа objclass и возвращает его дескриптор в переменную descriptor.  
Возвращает 0 при успешном создании объекта.

### delete

```
int object("delete", int descr);
```

descr.

Пример.

```
C          ts_table (          )
int TableDescr1;
object("create","ts_table",TableDescr1);
object("delete",TableDescr1);
```

## Объект ts\_table

---

### Функция ts\_table

ts\_table

#### init (clear)

```

:
ts_table(int iTable,"init");
ts_table(int iTable,"clear");

:
iTable -          ts_table,
```

### Руководство по LABPP\_Automat для ARCHICAD

```

:
ts_table(int iTable, "clear_rows");

:
iTable -
```

#### add\_column

```

:
ts_table(int descriptor, "add_column", int columnnumber, string columntype, string
columnname);
```

```
ts_table(int descriptor, "add_column", int columnnumber, string columntype, string
columnname, bool/int/double/string defalutvalue);
```

```

:
descriptor -
columnnumber -          (          -1          ).
columntype -          ("string", "int", "double", "bool").
columnname -          (          )
```

1. TableDescr1 0 " + . .":

```
ts_table(TableDescr1, "add_column",0,"string","код материала + ед.изм.");
```

2. ( )  
-10.1 .

```
ts_table(TableDescr1, "add_column", -1, "double", szonename, -10.1);
```

## is\_column\_exist

```

:
-1.
:
int iret = ts_table(int descriptor, "is_column_exist", string columnname);
:
descriptor -
columnname -
1.
iTableDescr1
"
"
:
int ires = ts_table(iTableDescr1, "is_column_exist", "Есть в проекте");
if(ires < 0) {
    ts_table(iTableDescr1, "add_column", -1, "bool", "Есть в проекте");
}

```

## set\_first\_key

```

"
"
:
,
0 1,
"
0", "
"
1", " - 1", " 1"
"
1", " - 2".
0",
( ),
"add_row" "add_row_sum"
( . add_row_sum).
:
1
ts_table(TableDescr1, "set_first_key", 1);
"
"
:
ts_table(TableDescr1, "set_first_key", "Наименование объекта");

```

## add\_row

```

:
ts_table(int TableDescr, "add_row", int column1index/string column1name,
bool/int/double/string column1value, ... , int columnNindex/string columnNname,
bool/int/double/string columnNvalue);
:
column1index column1name - ( 0)
column1value;
0 " " ( set_first_key),
:
ts_table(TableDescr1, "add_row", 0, objectname, 1, value_to_column_1);
ts_table(TableDescr1, "add_row", 0, objectname, 2, value_to_column_2);
0.

```



## add\_row\_sum

.  
 .  
 " " " ("100 40","50 40" . .), "  
 " (" . "," . "," . ") " "  
 ,  
 :

	.	-
100 50	.	1000
50 40	. .	2000
50 20	. .	1900

```
ts_table(TableDescr1,"add_column",0,"string","имя объекта + типоразмер + ед.изм");
ts_table(TableDescr1,"set_first_key",0);
ts_table(TableDescr1,"add_column", 1,"string","objectname");
ts_table(TableDescr1,"add_column", 2,"string","tiporazmer");
ts_table(TableDescr1,"add_column", 3,"string","edizm");
ts_table(TableDescr1,"add_column", 4,"double","kolvo");
ts_table(TableDescr1,"set_columns_to_add_sum", "kolvo");
```

:

```
ts_table(TableDescr1,"add_row_sum",0,objectname+tiporazmer+edizm, 1,objectname, 2,
tiporazmer, 3, edizm, 4, kolvo);
```

, " " + + . .  
 ,

## sort

7- ( ).  
 ,

```
ts_table(int iTableDescr, "sort", int icolnum1, int icolnum2, ... ,int icolnum7);
```

```
ts_table(int iTableDescr, "sort", string scolname1, string scolname2, ... ,string
scolname7);
```

:  
 icolumn1... icolumn7 -  
 scolname1... icolname7 -

1.  
 ts\_table(TableDescr1,"sort",0);

0

2. , "set\_first\_key".  
 ts\_table(TableDescr1,"sort");

```

3. " " " " 0,
ts_table(TableDescr1,"sort", 0, "Класс","Подкласс");

```

## sort\_back

```

(
). 7-
,
:
ts_table(int iTableDescr, "sort", int icolnum1, int icolnum2, ... ,int icolnum7);
ts_table(int iTableDescr, "sort", string scolname1, string scolname2, ... ,string
scolname7);

```

```

:
icolumn1... icolumn7 -
scolname1... scolname7 -

```

```

1. 0
ts_table(TableDescr1,"sort",0);

```

```

2. , "set_first_key".
ts_table(TableDescr1,"sort");

```

```

3. " " " " 0,
ts_table(TableDescr1,"sort", 0, "Класс","Подкласс");

```

## set\_columns\_to\_sort\_as\_LikeABC10\_a

```

Вызов:
ts_table(iTable, "set_columns_to_sort_as_LikeABC10_a");
или
ts_table(iTable,"set_columns_to_sort_as_LikeABC10_a",
int colnum1/string colname1,
int colnum2/string colname2,
...
int colnumN/string colnameN
);

```

Здесь:  
colnumN - номер колонки, в которой будет сортироваться значение как ABC10\_a;  
colname1 - то же можно задать названием колонки.

```

ABC10_a
,
:
(" 1001.b1", "
1001.a" . .).
ts_table(iTable, "set_columns_to_sort_as_LikeABC10_a", "Номер помещения");
задать суммирование по колонкам:
ts_table(iTable,"sort","Номер помещения");

```

## search

```

-1,
1 " "
int irow = ts_table(TableDescr1,"search",1,"Доска");

```

## search\_regex

```

REGEX
-1,
1 " "
int irow = ts_table(TableDescr1,"search_regex",1,"Доска", 2, "*");
(см. regex шаблоны)

```

## search\_extended

Найти в таблице строку, используя расширенный фильтр.

На примере показано как найти в таблице iTable строку, у которой в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "b22" (B - или на русском или на английском) и в колонке "Колонка 4" число больше или равно 2.

```

bool stoponfirstnoteq = false;
bool add_sum = false;

int ires = ts_table(iTable, "search_extended",
    "(, "ANY", "Колонка1", "=", "СТЕНА", "", "AND",
    "", "ANY", "Колонка2", "CONTAINS", "3", "", "AND",
    "", "ANY", "Колонка3", "REGEXMATCH", ".*[BB]22.*", "", "AND",
    "", "ANY", "Колонка4", ">=", 2, ")");

```

Здесь

iTable - дескриптор таблицы, в которой надо найти строку;

Возвращает -1 если строка не найдена, или индекс найденной строки (>=0).

## search\_extended\_table

Найти в таблице строку, используя расширенный фильтр.

Фильтр берется из таблицы фильтра.

На примере показано как найти в таблице iTable строку, у которой в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "b22" (B - или на русском или на английском) и в колонке "Колонка 4" число больше или равно 2.

```

bool stoponfirstnoteq = false;
bool add_sum = false;

int iTableFilter;
object("create", "ts_table", iTableFilter);
ts_table(iTableFilter, "create_structure_for_filter_table");
ts_table(iTableFilter, "add_row_to_filter_table", "(, "ANY", "Колонка1", "=",
    "СТЕНА", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка2", "CONTAINS",
    "3", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка3", "REGEXMACH",
    ".*[BB]22.*", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка4", ">=",

```

```
2, ")", "");
```

```
int ires = ts_table(iTable, "search_extended", iTableFilter);
```

Здесь

iTable - дескриптор таблицы, в которой надо найти строку;

iTableFilter - дескриптор таблицы, в которой описаны требования фильтра;

Возвращает -1 если строка не найдена, или индекс найденной строки ( $\geq 0$ ).

## search\_bounds

```
int ires = ts_table(1, "search_bounds", int colnumSX, int columnSY, int columnEX, int
columnEY, double posX, double posY, int startrow, int endrow);
```

ires - -1  
columnSX, columnSY, columnEX, columnEY -

posX, posY -

startrow -

endrow -

LABPP\_Table\_unicode.cpp -

## select\_row

```
int i=0;
ts_table(TableDescr1, "select_row", i);
```

## seek\_from\_top, seek\_from\_bottom, seek\_from\_current

```
int ires = ts_table(int iTable, "seek_from_top", int ioffset); // смещение от начала
int ires = ts_table(int iTable, "seek_from_bottom", int ioffset); // смещение от конца
таблицы
int ires = ts_table(int iTable, "seek_from_current", int ioffset); // смещение от текущей
позиции
```

iTable -

ioffset -

ires -

0

## add\_row\_from\_eq\_table

```
ts_table(int iTableDescr, "add_row_from_eq_table", int tiTableFrom, int indexrowfrom, bool
bsum);
```

iTableDescr -

iTableFrom -

indexrowfrom -

```

bsum - true,
add_row_sum ( . .
);

```

### add\_rows\_from\_eq\_table\_by\_filter

```

:
ts_table(int iTableDescr, "add_rows_from_eq_table_by_filter", int iTableFrom, bool
bstoponfirstnoteq, bool bsum, int icolumnindex1, string value1, ... string scolumnnameN,
double valueN);

:
iTableDescr - ;
iTableFrom - ;
bstoponfirstnoteq -
( . . );
bsum - true, add_row_sum ( . . );
icolumnindex1...N - ;
scolumnname1...N - ;
svalue1... N - ;
dvalue2... N - .

```

### add\_rows\_from\_eq\_table\_by\_regex\_filter

```

:
REGEX.
ts_table(int iTableDescr, "add_rows_from_eq_table_by_regex_filter", int iTableFrom, bool
bstoponfirstnoteq, bool bsum, int icolumnindex1, string value1, ... string scolumnnameN,
double valueN);

:
iTableDescr - ;
iTableFrom - ;
bstoponfirstnoteq -
( . . );
bsum - true, add_row_sum ( . . );
icolumnindex1...N - ;
scolumnname1...N - ;
svalue1... N - ;
dvalue2... N - .

```

### add\_rows\_from\_eq\_table\_by\_extended\_filter

Скопировать строки из одной таблицы в другую, используя расширенный фильтр.

На примере показано как скопировать из таблицы iTable в iTableFiltered все записи без суммирования числовых переменных одинаковых строк у которых в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "B22" (B - или на русском или на английском) и в колонке "Колонка 4" число больше или равное 2.

```

bool stoponfirstnoteq = false;
bool add_sum = false;

```

```

ts_table(iTableFiltered, "add_rows_from_eq_table_by_extended_filter", iTable, bool
stoponfirstnoteq, bool add_sum,
"(", "ANY", "Колонка1", "=", "СТЕНА", "", "AND",

```

```

"" , "ANY", "Колонка2", "CONTAINS", "3", "", "AND",
"" , "ANY", "Колонка3", "REGEXMATCH", ".*[BB]22.*", "", "AND",
"" , "ANY", "Колонка4", ">=", 2, ")" );

```

Здесь

iTableFiltered - дескриптор таблицы, в которую будут копироваться строки из таблицы iTable;

stoponfirstnoteq - остановить копирование при первом столкновении с неподходящей строкой;

add\_sum - при совпадении данных у строк по ключевой колонке - суммировать числовые значения;

## Руководство по LABPP\_Automat для ARCHICAD

Скопировать строки из одной таблицы в другую, используя расширенный фильтр, передаваемый в виде таблицы.

На примере показано как скопировать из таблицы iTable в iTableFiltered все записи без суммирования числовых переменных одинаковых строк у которых в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "B22" (B - или на русском или на английском) и в колонке "Колонка 4" число больше или равное 2.

```
bool stoponfirstnoteq = false;
```

```
bool add_sum = false;
```

```

int iTableFilter;
object("create", "ts_table", iTableFilter);
ts_table(iTableFilter, "create_structure_for_filter_table");
ts_table(iTableFilter, "add_row_to_filter_table", "(", "ANY", "Колонка1", "=",
"СТЕНА", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка2", "CONTAINS",
"3", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка3", "REGEXMATCH",
".*[BB]22.*", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка4", ">=",
2, ")", "");

```

```
ts_table(iTableFiltered, "add_rows_from_eq_table_by_extended_filter", iTable, bool
stoponfirstnoteq, bool add_sum);
```

Здесь

iTableFiltered - дескриптор таблицы, в которую будут копироваться строки из таблицы iTable;

stoponfirstnoteq - остановить копирование при первом столкновении с неподходящей строкой;

add\_sum - при совпадении данных у строк по ключевой колонке - суммировать числовые значения;

## get\_value\_of

```

.                                0                                objectname:
string objectname;
ts_table(TableDescr1, "get_value_of", 0, objectname);

```

## get\_rows\_count

```

int rowcount;
ts_table(TableDescr1, "get_rows_count", rowcount);

```

## get\_columns\_count

```
int colcount;
```

```
ts_table(TableDescr1,"get_columns_count", colcount);
```

## set\_columnstypes

```

:
ts_table(iTableDescr,"set_columnstypes",string col1name, string stype1, string col2name,
string stype2,...);

```

```
ts_table(iTableDescr,"set_columnstypes",int col1, string stype1, int col2, string
stype2,...);
```

```

:
col1name, col2name -
stype1, stype2 - ("string", "int", "double", "bool");
col1, col2 -

```

## get\_heading\_of

```

:
ts_table(iTableDescr,"get_heading_of", int col, string &colheading);

```

```
col - ( 0).
```

## print\_to\_str

```

:
text;
ts_table(iTableDescr,"print_to_str", string &text);

```

```

:
text -

```

## set\_columns\_to\_add\_sum

```
( "add_row_sum").
```

```

:
"add_row_sum":

```

```
ts_table(iTable,"set_columns_to_add_sum");
```

задать суммирование по колонкам:

```
ts_table(iTable,"set_columns_to_add_sum",
int colnum1/string colname1,
int colnum2/string colname2,
...
int colnumN/string colnameN
);
```

Здесь:

colnumN - номер колонки, в которой будет суммироваться значение, если значение первичного ключа (набор колонок, заданных в команде "set\_first\_key") совпадает с новой записью;  
colname1 - то же можно задать названием колонки.

Пример.

Суммировать количество из таблицы iTableMatRes по GUID'у в таблицу iTableMatResTmp и

выгрузить ее в Excel, изменив названия и последовательность колонок.

```
int ires = excel_attach();
if (ires != 0)
{
    tsalert(-1, "Ошибка во время выполнения", "Не получается подключиться к файлу excel", sWorkbook);
    return -1;
}
ires = excel_request("workbook_select", sWorkbook);
if (ires != 0)
{
    tsalert(-1, "Ошибка во время выполнения", "Не получается переключиться в файл excel", sWorkbook);
    excel_detach();
    return -1;
}

string sSheetName;
sSheetName = "Результат материалы";
ires = excel_request("sheet_select", sSheetName);
if (ires != 0) {
    tsalert(-1, "Ошибка во время выполнения", "Не получается переключиться на страницу", sSheetName);
    excel_detach();
    return -1;
}

int iTableMatResTmp;
object("create", "ts_table", iTableMatResTmp);
ts_table(iTableMatResTmp, "import_columns_from_table", iTableMatRes);
ts_table(iTableMatResTmp, "set_first_key", "GUID материала");
ts_table(iTableMatResTmp, "set_columns_to_add_sum", "Кол-во материала");
ts_table(iTableMatResTmp, "add_rows_from_eq_table_by_filter", iTableMatRes, false, true);

ts_table(iTableMatResTmp, "mapping_columns_to_export",
-1, "Поз.", "",
"Наименование материала (Rus)", "Наименование и техническая характеристика", "",
-1, "Код продукции", "",
-1, "Поставщик", "",
"Ед.изм. материала", "Ед.изм.", "",
"Кол-во материала", "Кол-во", "",
-1, "Масса 1 ед., кг", "",
-1, "Примечание", ""
);

ts_table(iTableMatResTmp, "export_to_excel", 1, "A", 1, 0, -1);
breakpoint(101);
object("delete", iTableMatResTmp);
excel_detach();
```

## Функции для выгрузки и загрузки таблиц

### export\_to\_excel

Excel.

```
ts_table(iTableDescr, "export_to_excel", int withheader, string startcolletter, int
```



```
startexcelrow, int starttablerow, int maxrows);
```

```

:
withheader = 0/1 -
startcolletter - "A" -
( "", - Excel)
starttablerow - (0 - )
maxrows - . (-1 - )

```

### import\_from\_excel

```

Excel.
ts_table(iTableDescr, "import_from_excel", string startcolletter, int startrow, int
maxrows, bool add_sum, bool deleteoldrows);

```

```

:
startcolletter - ( "" - ),
startrow - ( 1),
maxrows - ( -1 - ),
add_sum - "add_row_sum" ( ) - 0/1,
deleteoldrows - .

```

### import\_columns\_from\_excel

```

Excel.
"set_columntypes".
:
ts_table(iTableDescr, "import_columns_from_excel", string startcolletter, int startrow,
int maxcols);

```

```

:
startcolletter - Excel, "",
startrow - Excel (-1 - ),
maxcols - (-1 - );

```

### import\_columns\_from\_table

```

:
ts_table(iTableDescr, "import_columns_from_table", int iTableDescrFrom);

```

```

:
iTableDescrFrom - ,

```

### import\_columns\_from\_dialogcontrol

```

(ListBox . .)
:
ts_table(iTableDescr, "init_columns_from_dialogcontrol", int iDialogControlDescr);

```

```

:
iDialogControlDescr - ,

```

### import\_from\_dialogcontrol

```

:
ts_table(iTableDescr, "import_from_dialogcontrol", int iDialogControlDescr);

```

```

:
iDialogControlDescr -
export_to_dialogcontrol

```

```

ts_table(iTableDescr, "export_to_dialogcontrol", int iDialogControlDescr, int startrow,
int maxrows);

```

```

:
iDialogControlDescr -
startrow - ( 0),
maxrows - (-1 ).

```

export\_to\_dialogcontrol\_by\_filter

```

(getuservalue).
ts_table(iTableDescr, "export_to_dialogcontrol_by_filter", int iDialogControlDescr, int
startrow, int maxrows, bool bstoponfirstnoteq, string scolname1, void colvalue1, ...);

```

```

:
iDialogControlDescr -
startrow - ( 0),
maxrows - (-1 ).
scolname1... - ;
colvalue1... -

```

export\_to\_dialogcontrol\_by\_regex\_filter

```

REGEX.
(getuservalue).
ts_table(iTableDescr, "export_to_dialogcontrol_by_regex_filter", int iDialogControlDescr,
int startrow, int maxrows, bool bstoponfirstnoteq, string scolname1, void
colvalue1, ...);

```

```

:
iDialogControlDescr -
startrow - ( 0),
maxrows - (-1 ).
scolname1... - (
colvalue1... - 0);
( . regex ).

```

export\_to\_dialogcontrol\_by\_extended\_filter

Записать содержимое таблицы в табличный элемент диалога с использованием расширенного фильтра.

На примере показано как скопировать из таблицы iTable в табличный элемент диалога (iListBox) все записи от начала до конца таблицы с учетом фильтра - в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "b22" (B - или на русском или на

английском) и в колонке "Колонка 4" число больше или равное 2.

```
int startrow = 0;
bool stoponfirstnoteq = false;
int maxrows = -1;
```

```
ts_table(iTable, "add_rows_from_eq_table_by_extended_filter", iListBox, startrow,
stoponfirstnoteq, maxrows,
    "(", "ANY", "Колонка1", "=", "СТЕНА", "", "AND",
    "", "ANY", "Колонка2", "CONTAINS", "3", "", "AND",
    "", "ANY", "Колонка3", "REGEXMATCH", ".*[BB]22.*", "", "AND",
    "", "ANY", "Колонка4", ">=", 2, ")");
```

Здесь

iTable - дескриптор таблицы, из которой будут копироваться строки в табличный элемент диалога;

int startrow - , ;

int maxrows - (-1 - );

bool bstoponfirstnoteq - .

### export\_to\_dialogcontrol\_by\_extended\_filter\_table

Записать содержимое таблицы в табличный элемент диалога с использованием расширенного фильтра, переданного в виде таблицы.

Удобно когда необходимо задавать условия фильтров по ходу выполнения программы.

На примере показано как скопировать из таблицы iTable в табличный элемент диалога (iListBox) все записи от начала до конца таблицы с учетом фильтра - в колонке с названием "Колонка1" значение "СТЕНА", значение в колонке "Колонка2" - содержит "3", в колонке "Колонка3" содержится в любом месте "B22" или "B22" (B - или на русском или на английском) и в колонке "Колонка 4" число больше или равное 2.

```
int startrow = 0;
bool stoponfirstnoteq = false;
int maxrows = -1;
```

```
int iTableFilter;
object("create", "ts_table", iTableFilter);
ts_table(iTableFilter, "create_structure_for_filter_table");
ts_table(iTableFilter, "add_row_to_filter_table", "(", "ANY", "Колонка1", "=",
    "СТЕНА", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка2", "CONTAINS",
    "3", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка3", "REGEXMATCH",
    ".*[BB]22.*", "", "AND");
ts_table(iTableFilter, "add_row_to_filter_table", "", "ANY", "Колонка4", ">=",
    2, ")", "");
```

```
ts_table(iTable, "add_rows_from_eq_table_by_extended_filter", iListBox, startrow,
stoponfirstnoteq, maxrows, iTableFilter);
```

Здесь

iTable - дескриптор таблицы, из которой будут копироваться строки в табличный элемент диалога;

int startrow - , ;

int maxrows - (-1 - );

bool bstoponfirstnoteq - .

### resetoftfromexport

```
ts_table(iTableDescr, "resetoftfromexport", int col1, int col2);
ts_table(iTableDescr, "resetoftfromexport", string col1name, string col2name, ...);
```

```

:
col1,col2 - ( 0).
col1name, col2name - .

```

### export\_to\_LABPP\_Table

LABPP\_Table.gsm.

```

ts_table(iTable, "export_to_LABPP_Table", bool withheader, int startcolindex, int
startObjectrow, int starttablerow, int maxrows, string digit_delimiter);

```

или

```

ts_table(iTable, "export_to_LABPP_Table", bool withheader, int startcolindex, int
startObjectrow, int starttablerow, int maxrows, string digit_delimiter, int
iLABPP_TableGuidDescr);

```

```

:
withheader = 0/1 -
startcolindex - , ( 1);
startObjectrow - , ( 1);
starttablerow - ( 0- )
digit_delimiter - ("." ",")
maxrows - . (-1 - ).

```

LABPP\_Table

```

int main()
{
    int iTable;
    object("create", "ts_table", iTable); // создать таблицу

    // Создаем колонки таблицы
    ts_table(iTable, "add_column", 0, "string", "Моя первая колонка");
    ts_table(iTable, "add_column", 1, "double", "Моя вторая колонка с числом");
    ts_table(iTable, "add_column", 2, "string", "Примечания");

    // Добавляем строки
    ts_table(iTable, "add_row", 0, "Строка 1", 1, 10000.1111, 2, "Примечание 1");
    ts_table(iTable, "add_row", 0, "Строка 2", 1, 20000.2222);
    ts_table(iTable, "add_row", 0, "Строка 3", 1, 30000.335, 2, "Примечание 3");

    // Загрузить из выбранных элементов объекты LABPP_Table
    ac_request("load_elements_list_from_selection", 1, "ObjectType", "ObjectName",
"LABPP_Table", "MainFilter", 2);
    ac_request("get_loaded_elements_list_count", 1);
    int icount = ac_getnumvalue();
    int i;
    bool withheader = true;
    int startcolindex = 1;
    int startObjectrow = 1;
    int starttablerow = 0;
    int maxrows = -1;
    int maxcolumns;
    // получить число колонок и строк
    ts_table(iTable, "get_columns_count", maxcolumns);
    ts_table(iTable, "get_rows_count", maxrows);

    // Перебрать все отобранные объекты LABPP_Table. Если нужен только единственный -
можно цикл не делать.
    for (i = 0; i < icount; i++)
    {

```

```

// Установить текущим элементом объект из списка 1 № i
ac_request("set_current_element_from_list", 1, i);
// Записать в него содержимое таблицы
ts_table(iTable, "export_to_LABPP_Table", withheader, startcolindex,
startObjectrow, starttablerow, maxrows, "");
// Задать в объекте количество использованных строк
ac_request("set_object_property_value", "TS_RowsUsedCount", maxrows+1);
// Задать в объекте количество использованных колонок
ac_request("set_object_property_value", "TS_ColumnsUsedCount", maxcolumns);
// Дать команду объекту выполнить внутренний перерасчет
ac_request("set_object_property_value", "TS_DoRecalcAll", 1);
}
// Удалить из памяти таблицу
object("delete", iTable);
}

```

## Наименование таблицы

Моя первая колонка	Моя вторая колонка с числом	Примечания
Строка 1	10000,11	Примечание 1
Строка 2	20000,22	
Строка 3	30000,34	Примечание 3

### set\_columns\_width\_when\_export

```

:
ts_table(iTable, "set_columns_width_when_export", 40);
Здесь 40 - ширина колонок в пикселях.

```

### mapping\_columns\_to\_export

```

:
:
ts_table(iTable, "mapping_columns_to_export");
задать маппирование:
ts_table(iTable, "mapping_columns_to_export",
int basecolnum1/string basecolname1, string altcolname1, bool/int/double/string
default1,
int basecolnum2/string basecolname2, string altcolname2, bool/int/double/string
default2,
...
int basecolnumN/string basecolnameN, string altcolnameN, bool/int/double/string
defaultN
);
Здесь:
basecolnumN - номер базовой колонки, из которой брать данные. Если -1, то будет пустая
колонка, заполненная значением defaultN;

```

basecolnameN - здесь может быть указано имя базовой колонки;  
 altcolnameN - альтернативное название колонки, если значение "", то берется название базовой колонки;  
 defaultN - значение, которое будет выводиться в случае, если задана пустая колонка (-1).  
 Если задать "", то заполнения не будет.

Пример.

```
ts_table(iTableMatRes, "mapping_columns_to_export",
  -1, "Поз.", "",
  "Наименование материала (Rus)", "Наименование и техническая характеристика", "",
  -1, "Тип, марка, обозначение документа, опросного листа", "",
  -1, "Поставщик", "",
  "Ед.изм. материала", "Ед.измерения", "",
  "Кол-во материала", "Кол.", "",
  -1, "Масса 1 ед.,кг", "",
  -1, "Примечание", ""
);
```

### get\_base\_colindex\_by\_mapped\_column

```
.
:
int irealcolnum = ts_table(int iTable, "get_base_colindex_by_mapped_column", string
sMappingColName);
int irealcolnum = ts_table(int iTable, "get_base_colindex_by_mapped_column", int
iMappingColIndex);
```

```
.
:
irealcolnum - ,
sMappingColName - ( 0 N)
( 0 N) -1.
```

```
" " " " 3
(" ")
```

```
ts_table(iTableMatRes, "mapping_columns_to_export",
  -1, "Поз.", "",
  "Наименование материала (Rus)", "Наименование и техническая характеристика", "",
  -1, "Тип, марка, обозначение документа, опросного листа", "",
  -1, "Поставщик", "",
  "Ед.изм. материала", "Ед.измерения", "",
  "Кол-во материала", "Кол.", "",
  -1, "Масса 1 ед.,кг", "",
  -1, "Примечание", ""
);
```

```
int irealcolnum;
irealcolnum = ts_table(iTableMatRes, "get_base_colindex_by_mapped_column", "Ед.изм.
материала");
// результат - в irealcolnum теперь номер базовой колонки ("Ед.измерения") в таблице
iTableMatRes.
```

```
irealcolnum = ts_table(iTableMatRes, "get_base_colindex_by_mapped_column", 3);
// результат - в irealcolnum теперь -1, т.к. в таблице iTableMatRes при маппировании
базовая колонка не указана (-1).
```

## get\_count\_of\_mapped\_columns

```

:
int mappedcolumnscout = ts_table(int iTable, "get_count_of_mapped_columns");

```

```

:
iTable -
mappedcolumnscout =

```

## export\_to\_csv

Выгрузить содержимое таблицы в текстовую переменную.

```

ts_table(iTable, "export_to_csv", bool bWithHeadings, string column_separator, string
digit_separator, int start_row, int numRows, string csv_result);

```

Здесь:

iTable - дескриптор таблицы,  
bWithHeadings - выводить заголовки таблицы или нет,  
column\_separator - символ разделителя колонок - обычно ";",  
digit\_separator - разделитель целой и дробной частей чисел - обычно ",",  
start\_row - с какой строки выводить таблицу,  
num\_rows - сколько строк выводить (если -1 - то выводить все);

## Пример

```

//*****
// Импорт ts_table из текста в формате CSV
// и экспорт в этот формат
// LABPP 2021
//*****
int main()
{
    int iTable;
    object("create", "ts_table", iTable);

    string csvstr;
    csvstr = "\"Заголовок 1\";\"Заголовок 2\";\"Заголовок 3\"\\n\\
\"Значение 11 две кавычки \\\"\\\" текст\";123.3;\"Значение 31\"\\n\\
\"Значение 12\";321.21;\"Значение 32\"";

    ts_table(iTable, "import_from_csv", csvstr, ";");
    string str;
    ts_table(iTable, "print_to_str", str);
    coutvar << str;

    string csv_result;
    ts_table(iTable, "export_to_csv", true, ";", ",", 0, -1, csv_result);
    coutvar << csv_result;

    string FileNameAndPath = "c:\\\\LabPP\\\\test_07.csv";
    int iret = 0;
    int iFileDescr;
    object("create", "ts_file", iFileDescr);
    iret = ts_file(iFileDescr, "open", FileNameAndPath, "create", "we");
    if (iret == 0)
    {
        ts_file(iFileDescr, "write", csv_result);
        ts_file(iFileDescr, "close");
    }
    else

```

```

{
    tsalert(-3, "Ошибка записи", "Проверьте, пожалуйста доступность файла для
записи", FileNameAndPath);
}
object("delete", iFileDescr);
object("delete", iTable);
}

```

### import\_from\_csv

CSV.

```
ts_table(iTable, "import_from_csv", string csv_source, string column_separator);
```

```

:
iTable - ,
csv_source - CSV,
column_separator - ",".

```

### export\_to\_ac\_object\_parameter

GDL

```

:
int ires = ts_table(int iTable, "export_to_ac_object_parameter", string sParameterName);

```

```

:
iTable - ,
sParameterName - GDL ( ).

```

1 GDL		sMyParameters.
1	8	
2	2	

0 1 :

1. 1

2. , " 3"

```

int iTable;
object("create", "ts_table", iTable);
int ires = ac_request("set_current_element_from_list", 1, 0);
ac_request_special("get_element_value", "GDL", "sMyParameters", iTable); // считать
содержимое и структуру GDL параметра в таблицу iTable

```

```

string str;
ts_table(iTable, "print_to_str", str);
coutvar << str;

```

```

//считаем значение Мой параметр 1 в svalue;
ires = ts_table(iTable,"search",0,"Мой параметр 1");
string svalue;
if(ires >=0) {
    ts_table(iTable, "select_row", ires);
    ts_table(iTable, "get_value_of", 0, svalue);
}

```

```
//добавим, если такого еще нет нужно, считаем значение Мой параметр 1 в svalue;
```



```

ires = ts_table(iTable,"search",0,"Мой новый текстовый параметр");
if(ires < 0) {
    ts_table(Table,"add_row", 0, "Мой новый текстовый параметр", 1, "значение 3", 2,
"строка");
}
else
{
    ts_table(iTable, "select_row", ires);
    ts_table(iTable, "set_value_of", 1, "значение 3");
}
// Записать таблицу назад в параметр текущего элемента. Размер таблицы-параметра GDL -
изменить с учетом размера таблицы iTable.
ires = ts_table(iTable, "export_to_ac_object_parameter", "sMyParameters");

```

### set\_column\_user\_value

```

int ires = ts_table(int iTable, "set_column_user_value", int icol, string/double/int/bool
valuetoset);

```

```

:
iTable - ;
icol - ;
vsluetoset - ,

```

### get\_column\_user\_value

```

int ires = ts_table(int iTable, "get_column_user_value", int icol, string/double/int/bool
vartogetvalue);

```

```

:
iTable - ;
icol - ;
vartogetvalue - ,

```

### set\_column\_user\_genesis

```

int ires = ts_table(int iTable, "set_column_user_genesis", int icol, string sgenesis);

```

```

:
iTable - ;
icol - ;
sgenesis -

```

### get\_column\_user\_genesis

```

"get_column_user_genesis".
int ires = ts_table(int iTable, "get_column_user_genesis", int icol, string sgenesis);

```

```

iTable - ;
icol - ;
sgenesis - .

```

## delete\_rows

```

:
ts_table(int iTable, "delete_rows", int index, int count);

```

```

:
iTable - ;
index - ( 0 n-1);
count - .

```

"clear\_rows".

## set\_value\_of\_toallrows

```

:
:
ts_table(int iTable, "set_value_of_toallrows", index/string column_index_or_name,
bool/int/double/string newvalue, int irowfrom, int irowto);

```

Здесь:

iTable - дескриптор таблицы;

column\_index\_or\_name - индекс или имя колонки;

newvalue - значение, которое требуется установить ячейкам;

irowfrom - индекс строки, с которой необходимо задать значение (от 0 до n-1);

irowto - индекс последней строки для изменения (если -1 - то до конца таблицы).

## Функции стандартной библиотеки

---

### Преобразование

#### atoi

```

:
string ivalue = atoi(string svalue);

```

#### itoa

```

string svalue = itoa(int ivalue);

```

#### atof

```

double dresult = atof(string svalue);

```

```

: svalue - , , - .

```

#### ctos

```

Excel . . .
:

```

```

string ssymbol = ctos(int isymbolcode);

:
icymbolcode -
ssymbol -
:
Excel

int chcolumn = 'A'+1;
string ssymbol = ctos(chcolumn);
string address = ssymbol+itoa(2)+":"+ssymbol+itoa(2);
cout << address;

"B2:B2"

```

## grad\_to\_radian

```

:
double grad_to_radian(double grad);

: grad -

180
double result = grad_to_radian(180);

result (3.14....).

```

## radian\_to\_grad

```

:
double radian_to_grad(double rad);

3.14
double result = radian_to_grad(3.14);

result 180.

```

## sprintf

```

printf.

:
string sprintf( string format, value);

string sresult = sprintf("%.2f", 2.024);

result "2.02".

```

001	"%03i"	int	int m = 10; string s = sprintf("%03i",m);

## ecvt\_french

```

:
string svalue = ecvt_french(double dvalue);

111222333.11.

```

```
double dvalue = 111222333.11;
string sresvalue = ecvt_french(svalue);
cout << sresvalue;

- "111 222 333,11"
```

### ecvt

```
double dvalue = 111222333.11;
string svalue = ecvt(double dvalue);

111222333.11.
```

```
double dvalue = 111222333.11;
string sresvalue = ecvt(svalue);
cout << sresvalue;

- "111222333.11"
```

## Математические функции

### abs

```
double dres = abs(double dvalue);

dres - dvalue
```

### max

```
double dres = max(double dvalue1, double dvalue2);

dres - dvalue1 dvalue2
```

### min

```
double dres = min(double dvalue1, double dvalue2);

dres - dvalue1 dvalue2.
```

### rand

```
double dres = rand();

0 1;
```

### ln

### log

sqrt

sqr

pow

percent

tsround

```
double tsround(double dvalue, int numpos);
```

dvalue - ,  
numpos -

tsround\_best

```
double tsround_best(dvalue, int numpos);
```

dvalue - ,  
numpos -

floor

```
double dresult = floor(double dvalue);
```

ceil

```
double dresult = ceil(double dvalue);
```

math\_const

```
double math_const(string sConstName);
```

sConstName -

	/
Pi	3.14159265359

MinDouble	double
MaxDouble	double
e	2.71828182846

## inrange

```

:
bool inrange(double value, string left_parenthesis, double left_margin, double
right_margin, string right_parenthesis);

```

```

:
value -
left_parenthesis = ("], "(" "[" - " "left_margin
", "left_margin left_margin");
left_margin -
right_parenthesis = ("[, ")" "]" - " "right_margin
", "right_margin right_margin");
right_margin -
;

```

## Тригонометрические функции

### cos

```

:
double dresult = cos(double cornerinrad);

```

### sin

Вызов:

```

double dresult = sin(double cornerinrad);

```

### arcsin

### arccos

### tg

### arctg

### ctg

### arcctg

## Функции геометрического преобразования

ac\_request() geometry\_calc\_2d

### is\_curelem\_inside\_element\_polygon

( , , 3d- . ).

```
bool res = ac_request("geometry_calc_2d"," is_curelem_inside_element_polygon",double
x1,double y1, int iElemDescr, bool mainContOnly, double grow_contour, double grow_holes,
int mode);
```

### is\_point\_on\_element\_polygon

3d- . .) ( , , ,

```
bool res = ac_request("geometry_calc_2d","is_point_on_element_polygon",double x1,double
y1, int iElemDescr, bool mainContOnly, double grow_contour, double, grow_holes);
```

x1 y1 - , iElemDescr - ,

mainContOnly – ,  
grow\_contour grow\_holes –

3d- ID=" " ac\_element.  
ac\_element\_guid,

```
// считать таблицу координат элемент рельефа 3d mesh с ID="Рельеф"
int iMeshCoordTable; // дескриптор таблицы координат полигона
// считать элементы типа 3d-сетка с ID="Рельеф" в список № 2
ac_request("load_elements_list",2,"MeshType","ID","Рельеф","MainFilter",2);
// определить сколько элементов содержит список № 2
ac_request("get_loaded_elements_list_count",2);
int iicount = ac_getnumvalue();
if(iicount == 0)
{
    cout << "В проекте отсутствует 3d сетка с ID=Рельеф.\nПрограмма остановлена";
    return -1;
}
// создать объект динамической таблицы для координат полигона
object("create","ts_table",MeshCoordTable);
// установить фокус на первом элементе (индекс 0) списка № 2
ac_request("set_current_element_from_list",2, 0);
// создать объект для guid элемента
int iMeshGuidDescr; // дескриптор объекта guid
object("create","ac_element_guid",iMeshGuidDescr);
// записать из текущего элемента в списке № 2 guid в объект iMeshGuidDescr
ac_request("store_cur_element_to_descr", iMeshGuidDescr);
int iMeshElemDescr; // дескриптор объекта элемента
// создать объект для работы с элементом
object("create","ac_element",iMeshElemDescr);
// загрузить объект элемент из guid
ac_request("load_element_from_guid",iMeshElemDescr,iMeshGuidDescr);
// считать таблицу координат полигона
ac_request("get_element_value","SimpleCoordTable",iMeshCoordTable);
// проверить сколько точек содержит полученная таблица координат полигона
```

```

int irowcount;
ts_table(iMeshCoordTable, "get_rows_count", irowcount);
cout << "Количество точек в контуре рельефа=" << irowcount << "\n";
object("delete",MeshCoordTable);
double x1=1.1, y1=2.2; // координаты исследуемой точки
// проверить лежит ли точка с координатами x1, y1
int res = ac_request("geometry_calc_2d","is_point_on_element_polygon",x1,y1,
iMeshElemDedcr);
if(res == 1)
{
    cout << "Точка находится внутри контура полигона элемента";
}

```

### rotate\_point\_and\_move

```

ac_request("geometry_calc_2d","rotate_point_and_move",double centerXfrom, double
centerYfrom,double pXfrom,double pYfrom, double alpha_rad, double centerXto, double
centerYto,double scale,double &pXres, double& pYres);

```

centerXfrom, centerYfrom -  
 pXfrom, pYfrom -  
 alpha\_rad -  
 centerXto, centerYto -  
 scale - ]0,1],  
 - pXres, PYres -

### get\_cross\_point\_of\_2lines

Обращение  
 ac\_request("geometry\_calc\_2d","get\_cross\_point\_of\_2lines",double l1sX, double l1sY,double
 l1eX,double l1eY, double l2sX, double l2sY, double l2eX,double l2eY,double& pXres,
 double& pYres);

:  
 l1sX,l1sY,l1eX, l1eY -  
 l2sX,l2sY,l2eX, l2eY -  
 pXres pYres.

### get\_rot\_and\_move\_point

```

ac_request("geometry_calc_2d","get_rot_and_move_point",double P1X, double P1Y, double dL,
double dAngleRad, double & pXres, double& pYres);

```

:  
 P1X, P1Y - , dL - ,  
 dAngleRad - X.  
 pXres, pYres -

### get\_length\_2point

```

ac_request("geometry_calc_2d","get_length_2point",double l1sX, double l1sY,double
l1eX,double l1eY, double &resLength);

```

:  
 l1sX, l1sY, l1eX, l1eY -



resLength.

## is\_point\_on\_line

```

:
bool bres = ac_request("geometry_calc_2d","is_point_on_line",double P1X, double P1Y,
double P2X, double P2Y, double PointX, double PointY);

```

P1X,P1Y,P2X,P2Y -

PointX, PointY -

bres - 1 0

## get\_line\_angle

```

ac_request("geometry_calc_2d","get_line_angle",double l1sX, double l1sY,double
l1eX,double l1eY, double &resAngleInRad);

```

:

l1sX,l1sY,l1eX, l1eY -

resAngleInRad.

## Специальные функции

---

### codemeter - измерение времени выполнения фрагмента кода

```

:
double dvalue = codemeter(int what);
what-0 1.
0,
1 -
2 -

10000

codemeter(0);
int i;
for(i=0;i<10000;i++)
{
cout << i << "\n";
}
cout<< "Цикл в 10 000 оборотов выполнен за " << codemeter(1) << " секунд";

```

### ac\_save\_messages\_to\_file - сохранить текст из окна сообщений в файл

```

:
ac_save_messages_to_file(string filepath);
filepath

c:\my_file.txt:
ac_save_messages_to_file("c:\my_file.txt");

```

C++

## Связь с квартирोगрафией

---

### solaris\_test

```

                                ARCHICAD
                                .
                                solaris_test.
                                ac_request
                                .

int ires = ac_request("solaris test");

                                0.

int res;
res = ac_request("solaris_test");
if(res != 0) {
    cout << "Нет связи с LabPP_Solaris\n";
    return -1;
}

```

### get\_flat\_rooms

```

                                - ac_request("get_flat_rooms"... )
                                .
                                .

int ires = ac_request("get_flat_rooms",int iFlatGuidObjDescr, int iListNum);
:
iFlatGuidObjDescr -                                guid                                /                                .
iListNum -                                ,                                /                                .
                                - 0 -                                .

```

## runtimecontrol

---

### workline

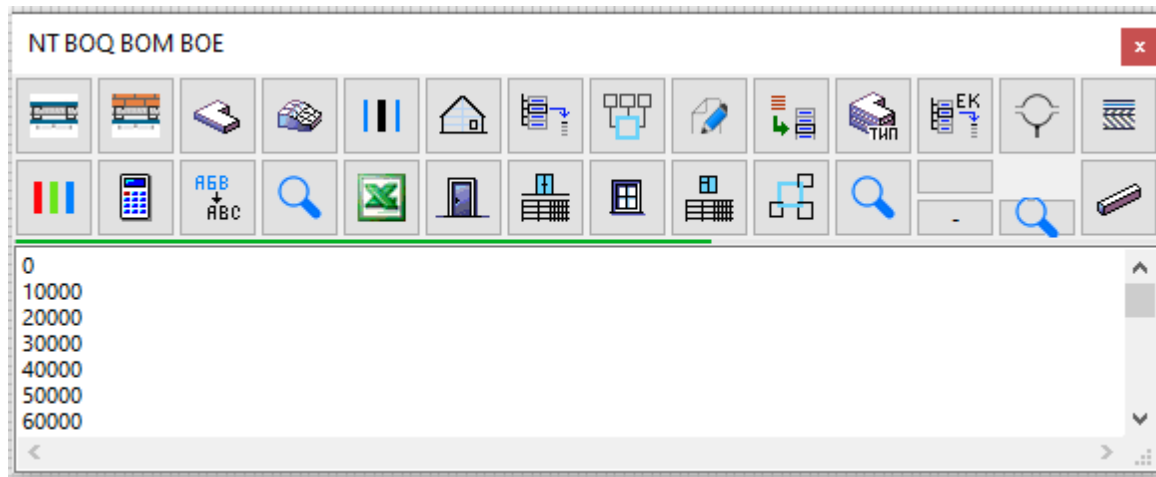
```

                                :
runtimecontrol("workline", string sdirective, int ivalue);
или назначить текущее значение дорожки процентов и одновременно написать это значение в
окно сообщений через каждые igap итераций
runtimecontrol("workline", "setpos_and_print_through_gap", int ivalue, int igap);

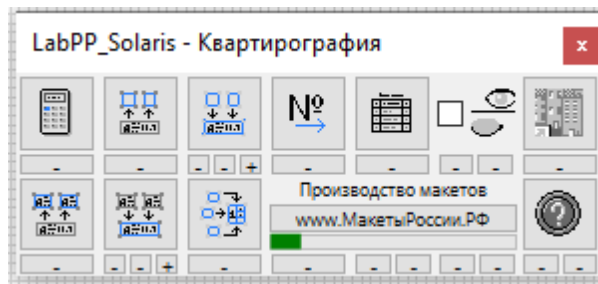
:
directive -                                ,                                "setmin"/"setmax" /"setpos" -                                ,
                                ,
                                "setpos_and_print_through_gap" - то одновременно с назначением
позиции
ivalue -                                ,                                .
igap -                                ,                                (
                                100                                1000).

LabPP_Automat                                :

```

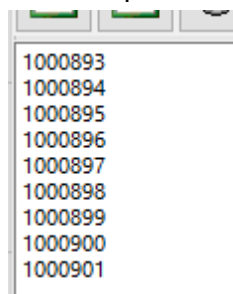


LabPP\_Solaris



1.

```
int main()
{
    int imax = 1000901;
    runtimecontrol("workline", "setmax", imax);
    int i;
    for (i = 0; i < imax; i++)
    {
        cout << i << "\n";
        runtimecontrol("workline", "setpos", i);
    }
    runtimecontrol("workline", "setpos", 0);
}
```



2.

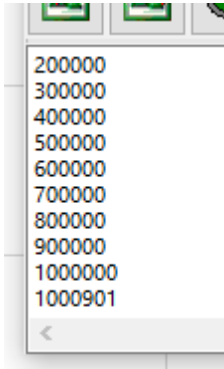
```
100 000                                1000901.
int main()
```

```

{
    int imax = 1000901;
    runtimecontrol("workline", "setmax",imax);
    int i;
    for (i = 0; i < imax; i++)
    {
        cout << i << "\n";
        runtimecontrol("workline", "setpos_and_print_through_gap", i+1, 100000);
    }
    runtimecontrol("workline", "setpos", 0);
}

:

```



## Команды для управления в приложении LabPP\_Calc

### interface

#### calc\_field

```

ARCHICAD.
:
ac_request("interface", "calc_field", string what, string fieldname, int dochangedot,
string svalue,string scomment);
:
what - "set" "get" -
fieldname - "a"/"b"/"c"/"main" -
dochangedot - (1/0).
svalue -
scomment - "set"
:
"a" 111.11

```

```
ac_request("interface","calc_field","set","a",1,"111.11","s");
```

#### get\_active

```

:
ac_getstrvalue() - "a","b","c" "A","B","C" "D".
"main".
:
ac_request("interface", "calc_field", "get_active");
string sletter = ac_getstrvalue();

```

## Внешние переменные

---

var\_extern\_get var\_extern\_set.

### var\_extern\_set

```

:
var_extern_get(string varname, string varvalue);
var_extern_get(string varname, double varvalue);

:
varname - , varvalue

```

### var\_extern\_get

```

:
var_extern_set(string varname, string varnameforvalue);
var_extern_set(string varname, double varnameforvalue);

:
varname - , varnameforvalue -

```

### var\_extern\_delete

```

:
var_extern_delete( string sVarName);

:
sVarName - ,

```

```

int main()
{
#pragma region Создание диалога
    object("create","ts_dialog",iDialogDescr);
    ts_dialog(iDialogDescr, "init_dialog","palette",0,0,450,400);
    ts_dialog(iDialogDescr, "set_as_main_panel");
    ts_dialog(iDialogDescr, "eventreaction", "Event_PanelCloseRequested"); // задание
реакции на событие
    ts_dialog(iDialogDescr, "SetTitle","Выгрузка BOQ и BOM");
    ...
#pragma endregion

    ts_dialog(iDialogDescr, "invoke", bres);
}

// Обработчик событий
int Event_PanelCloseRequested(int iDescr, string sDescr)

```

```

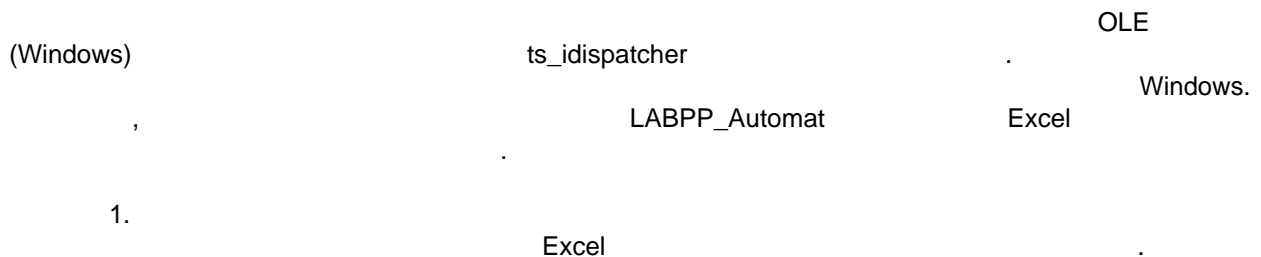
{
    int toret = 0;

    int iTableReestrEdinichnyhRascenok; // реестр единичных расценок
    int ires = 0;
    ires = var_extern_get("iTableReestrEdinichnyhRascenok",
iTableReestrEdinichnyhRascenok, 0);
    if (ires >= 0)
    {
        object("delete", iTableReestrEdinichnyhRascenok);
        var_extern_delete("iTableReestrEdinichnyhRascenok");
    }
    return toret;
}

// Создать таблицу, если она еще не создана
int CreateTableIfNeeded()
{
    int ires = 0;
    ires = var_extern_get("iTableReestrEdinichnyhRascenok",
iTableReestrEdinichnyhRascenok, 0);
    if (ires != 0)
    {
        //
        object("create", "ts_table", iTableReestrEdinichnyhRascenok);
        object("KeepInMemory", iTableReestrEdinichnyhRascenok);
        ...
    }
    else
    {
        ts_table(iTableReestrEdinichnyhRascenok, "reset_cpp_to_current");
    }
    return 0;
}

```

## Объект ts\_idispatcher



```

//-----
// Example
// Read number current cell in Excell table
// LABPP 2022
//-----
int main()
{
    int iIDispatchExcel;
    object("create","ts_idispatch",iIDispatchExcel);
    ts_idispatch(iIDispatchExcel,"attach","Excel.Application"); // create dispatcher
object

    int iVariantRange;
    object("create","ts_variant",iVariantRange); // create variant object to obtain
dispatcher object of the current active cell

```

```

    int ires = ts_idispatch(iIDispatchExcel, "AutoWrap", iVariantRange,
"DISPATCH_PROPERTYGET", "ActiveCell", 0);

    if(ires != 0) {
        return -1;
    }

    int iIDispatchActiveCell;
    object("create","ts_idispatch",iIDispatchActiveCell);
    ts_variant(iVariantRange,"get_pdispVal", iIDispatchActiveCell);

    int iVariantParm;
    object("create","ts_variant",iVariantParm);
    // ts_variant(iVariantParm,"set_dblVal",111.10);
    // ts_variant(iVariantParm,"set_bstrVal","Stop War");

    ires = ts_idispatch(iIDispatchActiveCell,"AutoWrap",
iVariantParm,"DISPATCH_PROPERTYGET", "FormulaR1C1", 0);
    // ires = ts_idispatch(iIDispatchActiveCell,"AutoWrap",
iVariantParm,"DISPATCH_PROPERTYGET", "Value", 0);

    if (ires == 0)
    {
        string s;
        ts_variant(iVariantParm, "get_value_simple",s);
        coutvar << s;
        int i;
        ts_variant(iVariantParm, "get_value_simple",i);
        coutvar << i;
        double d;
        ts_variant(iVariantParm, "get_value_simple",d);
        coutvar << d;
    }
    object("delete",iVariantParm);
    object("delete",iVariantRange);
    object("delete",iIDispatchActiveCell);

    ts_idispatch(iIDispatchExcel,"detach");
    object("delete",iIDispatchExcel);
}

```

2.

Excel

```

//-----
// Example
// Write number 111.10 to current cell in Excell table
// LABPP 2022
//-----
int main()
{
    int iIDispatchExcel;
    object("create","ts_idispatch",iIDispatchExcel);
    ts_idispatch(iIDispatchExcel,"attach","Excel.Application"); // create dispatcher
object

    int iVariantRange;
    object("create","ts_variant",iVariantRange); // create variant object to obtain
dispatcher object of the current active cell
    int ires = ts_idispatch(iIDispatchExcel, "AutoWrap", iVariantRange,
"DISPATCH_PROPERTYGET", "ActiveCell", 0);

    if(ires != 0) {
        return -1;
    }
}

```

```

}

int iIDispatchActiveCell;
object("create","ts_idispatch",iIDispatchActiveCell);
ts_variant(iVariantRange,"get_pdispVal", iIDispatchActiveCell);

int iVariantParm;
object("create","ts_variant",iVariantParm);
// ts_variant(iVariantParm,"set_dblVal",111.10);
ts_variant(iVariantParm,"set_value_simple","Stop War");

int iVariantNull;
object("create","ts_variant",iVariantNull);

ires = ts_idispatch(iIDispatchActiveCell,"AutoWrap",
iVariantNull,"DISPATCH_PROPERTYPUT", "FormulaR1C1", 1, iVariantParm);

object("delete",iVariantParm);
object("delete",iVariantRange);
object("delete",iIDispatchActiveCell);
object("delete",iVariantNull);

ts_idispatch(iIDispatchExcel,"detach");
object("delete",iIDispatchExcel);
}

```

## CreateInstance

Windows OLE CreateInstance.

```

:
int iret = ts_idispatch(int iIDispatchApp,"CreateInstance",string appclassname, string
runcontext);
:
iIDispatchApp - ts_idispatch ;
appclassname - ("Excel.Application" - Excel);
runcontext - .

```

"REMOTE_SERVER"	
"LOCAL_SERVER"	
"ALL"	

## attach

Windows OLE GetActiveObject.

```

Excel).
:
int iret = ts_idispatch(int iIDispatchApp,"attach",string appclassname);
:
iIDispatchApp - ts_idispatch ;
appclassname - ("Excel.Application" - Excel);

```

## detach

```

:
ts_idispatch(int iIDispatch,"detach");

```



:
iIDispatch - , .

AutoWrap

Windows

```

:
int ires = ts_idispatch(iIDispatch,"AutoWrap", int iVarientRes, string commandtypecall,
string commandname, 0);
или
int ires = ts_idispatch(iIDispatch,"AutoWrap", int iVarientRes, string commandtypecall,
string commandname, int par_count, int iVarientArg1, ...);

```

:
ires - 0
iIDispatch - ts\_idispatch,
iVarientRes - ts\_variant
commandtypecall -

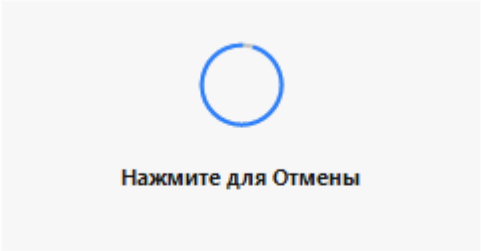
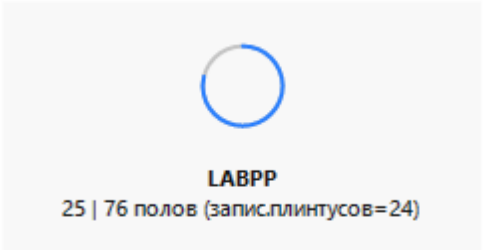
"DISPATCH_PROPERTYGET"	
"DISPATCH_PROPERTYPUT"	
"DISPATCH_METHOD"	
"DISPATCH_PROPERTYPUTREF"	

commandname - ( "ActiveCell" Excel . ).

Управление выполнением и поведением программы

Прерывание выполнения длительных процессов ac\_process\_window

- .
(25), (76),
(24).



```

bool bStop = false;
ac_process_window("Init", "LABPP"); // иницииновать окно процесса
string sPhase = "полов (запис.плинтусов=";
int iAssigned = 0;

for (i = 0; i < icount; i++)
{
    bStop = ac_process_window("SetPhase", sPhase+itoa(iAssigned)+")", icount-i);
    if (bStop == true)
    {
        cout << "\nПроцесс остановлен пользователем\n";
        ac_process_window("Close");
        return -1;
    }
    ac_request("set_current_element_from_list", 1, i);
    ac_request("elem_user_property", "set", "Тип отделки пола", "");
}
ac_process_window("Close");

```

## Init

```

:
ac_process_window("Init", string sTitle);

```

Здесь:  
sTitle - название процесса.

## SetPhase

```

:
bool bStop = ac_process_window("SetPhase", string sPhase, int iCounter2);
:
sPhase - ,
iCounter2 - ,
sPhase.
iCounter2 = -1,
true

```

## IsCanceled

Проверить не поступал ли запрос от пользователя остановить процесс.  
Если нажималась кнопка мыши на окне процесса, то вернется true.  
Вызов:

```

bool bStop = ac_process_window("IsCanceled");

```

## Close

```

:
ac_process_window("Close");

```

## throw\_process

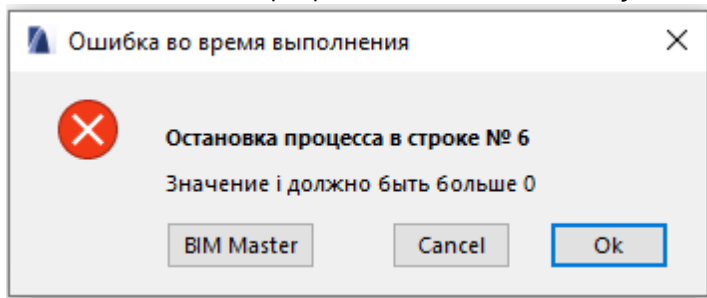
BIM Master

```
void throw_process(string smessage);

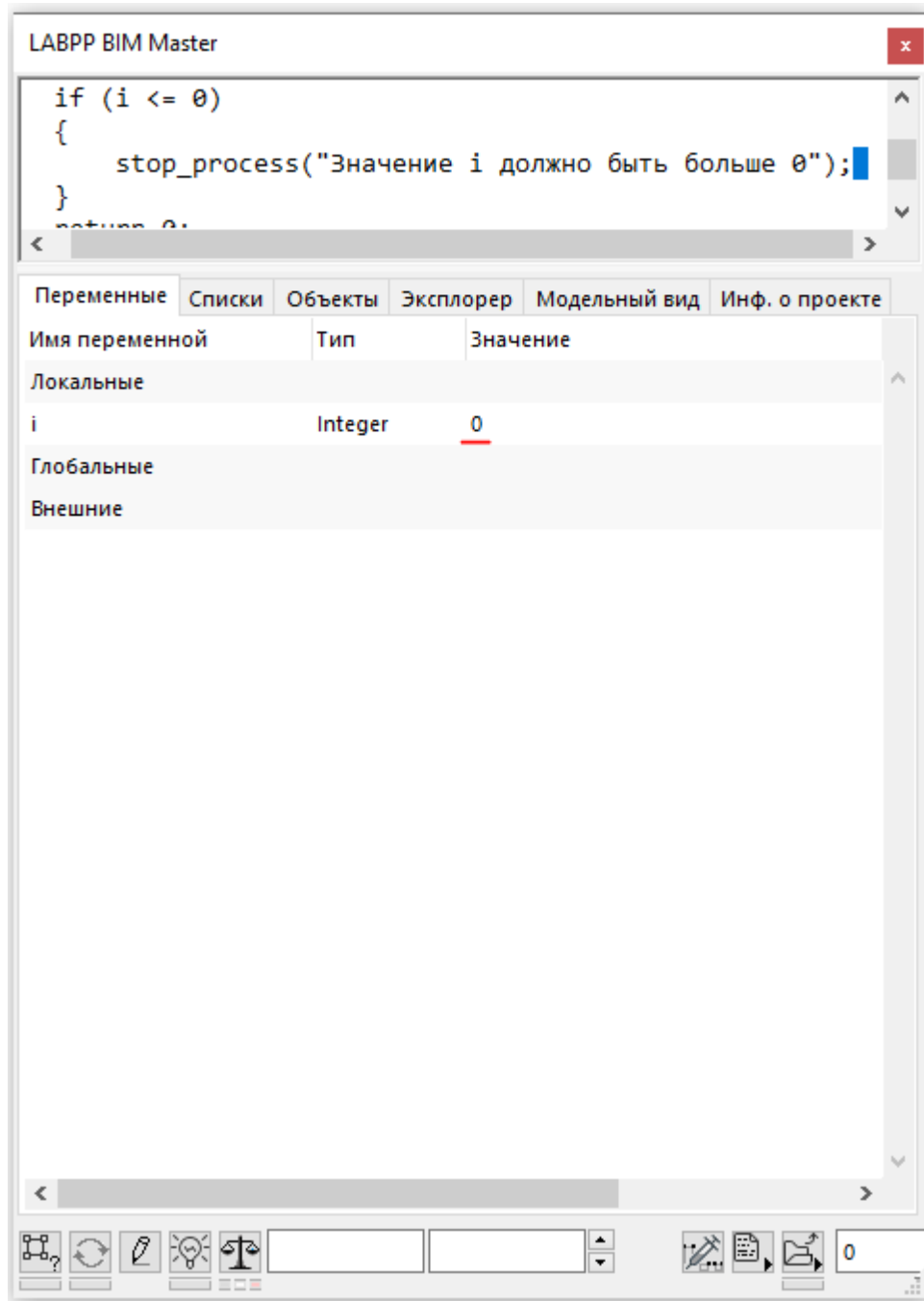
smessage - , i 0.

int main()
{
    int i=0;
    if (i <= 0)
    {
        throw_process("Значение i должно быть больше 0");
    }
    return 0;
}
```

В ходе выполнения программы пользователь получит такое сообщение:



Он сможет проанализировать, при необходимости, ситуацию при помощи панели BIM Master.



## Поведение программы при ошибках считывания/записи переменных элементов

Функции чтения/записи данных в переменные элементов проекта возвращается код ошибки.

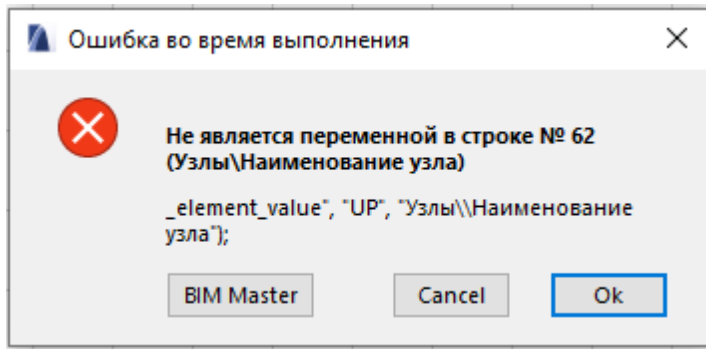
Если 0 - ошибки нет. Если < 0, то в ходе выполнения операции возникла ошибка.

В случае, если ошибки считывания/записи являются критическими, т.е. если нельзя допустить чтобы у каких-то элементов выборки отсутствовали переменные или записались с ошибкой - можно сократить код используя соответствующие конструкции.

```
SETCFG("stop_on_err_elem_value_get", bool bStopOnError);
SETCFG("stop_on_err_elem_value_set", bool bStopOnError);
```

Данные конструкции влияют на текущий сеанс интерпретатора (запуск программы);

Если случится ситуация, когда переменная элемента (например пользовательское свойство или GDL параметр) будет отсутствовать или записано с ошибкой, то пользователь увидит информацию об ошибке в окне и программа предложит вызвать BIM Master для выяснений обстоятельств и устранения проблемы на месте.



## Остановка процесса помещением файла в каталоге

100  
c:\labpp\stop.txt.  
-  
.bat

```
int main()
{
    int i = 0, j=0;
    int ires;
    for (i = 0; i < 100000; i++, j++)
    {
        if (j >= 10000)
        {
            j = 0;
            coutvar << i;
        }
        if (check_user_break()) {
            break;
        }
    }
}

bool check_user_break()
{
    string stopfilename = "c:\\labpp\\stop.txt";
    bool bexist = false;
    shell_func("file", "is_exist", stopfilename, bexist);
    if (bexist)
    {
        shell_func("file", "delete", stopfilename);
        if (tsalert(-3, "Запрос", "Остановить выполнение?", "", "Да", "Нет") == 1)
        {
            return true;
        }
    }
    return false;
}
```

## Объект ts\_variant

ts\_variant Windows.  
Windows IDispatch

### get\_pdispVal

ts\_variant ts\_idispatch  
OLE Windows.  
:  
`int` iret = ts\_variant(`int` iVarientRange, "get\_pdispVal", `int` IIDispatchActiveCell);  
Здесь:  
iVarientRange - дескриптор объекта типа ts\_variant, который содержит объект управления IDispatch.  
IIDispatchActiveCell - дескриптор объекта типа ts\_idispatch, в который надо записать этот элемент управления IDispatch.

Пример.  
Получить внутренний элемент управления текущей ячейкой таблицы Excel.

```
int IIDispatchExcel;
object("create", "ts_idispatch", IIDispatchExcel);
ts_idispatch(IIDispatchExcel, "attach", "Excel.Application"); // create dispatcher object

int iVarientRange;
object("create", "ts_variant", iVarientRange); // create variant object to obtain
dispatcher object of the current active cell
int ires = ts_idispatch(IIDispatchExcel, "AutoWrap", iVarientRange,
"DISPATCH_PROPERTYGET", "ActiveCell", 0);
```

### get\_value\_simple

ts\_variant  
:  
ts\_variant(`int` iVarient, "get\_value\_simple", `bool/int/double/string` value\_ret);  
:  
iVarient - ts\_variant;  
valut\_ret - , iVarient.  
.  
iVariantParm  
.  
`string` s;  
ts\_variant(iVariantParm, "get\_value\_simple", s);  
coutvar << s;

### set\_value\_simple

ts\_variant.  
:  
ts\_variant(`int` iVarient, "set\_value\_simple", `bool/int/double/string` value);  
:  
iVarient - ts\_variant;  
value - , iVarient.

## run\_cpp - запуск другой программы

run\_cpp

### run\_from\_file

```
double dret = run_cpp("run_from_file",string sFileNameAndPath,int arg1,double arg2,string
arg3);
:
sFileNameAndPath -
arg1, arg2 arg3 -
get_args)
dret
return
"c:\my_program.cpp"
100.
```

**int myintvar = 100;**

**double dret = run\_cpp("run\_from\_file","c:\my\_program.cpp",myintvar,0,"");**

### run\_from\_variable

```
double iret = run_cpp("run_from_variable",strint programtext,int arg1,double arg2,string
arg3);
:
programtext -
arg1, arg2 arg3 -
get_args)
100, 111.1 "
```

```
string programtext = "int main(){ cout << \"Моя программа \"<<\"\\n\\n\"; int iarg1; double
darg2; string sarg3; run_cpp(\"get_args\",iarg1, darg2, sarg3);";
programtext += "cout << iarg1 << \" ,\" << darg2 << \" ,\" << sarg3; return -1;}";
int iret = run_cpp("run_from_variable", programtext, 100, 111.1, "аргумент текстовый");
cout << "iret = " << iret << "\\n";
```

```
return -1.
iret -1.
```

## Получение аргументов внутри программы

run\_cpp get\_args.

run\_cpp("get\_args",int iarg1, double darg2, string sarg3);

```
:
iarg1, darg2, sarg3 -
```

## Обработка ошибок

### Коды ошибок

0, - 0.

	C++	
0		
-1		TSErrCommonError
-1100		TSErrLogicError
-1222		TSErrVarNotExist
-1223		TSErrBadIndex
-1224		TSErrVarValueNotAvailable
-2222		TSErrIOSourceNotFound
-2223		TSErrIOTagetExists
-2224		TSErrIODeviceIsFull
-2225		TSErrIOAccessDenied
	JSON	
0		tsjsonerr_no_err
2000		tojsonerr_empty_json
2001		tsjsonerr_not_a_string
2002		tsjsonerr_not_a_colon
2003		tsjsonerr_type_mismatch
2004		tsjsonerr_not_exist

### Функции интерфейса LabPP\_Automat

```
LabPP_Automat

) tsprg (
, LabPP_Automat
config.cpp
LabPP_Automat
( . ).

tsimages (
```

### create\_iconbutton

```
:
ac_request("create_iconbutton",string sPictureName,int sx,int sy, int ex, int ey, string
sToolTip, string sPrgCppFileName);
:
sPictureName - ( tsimages).
sx,sy,ex,ey - LabPP_Automat,
sToolTip - ,
```



```
sPrgCppFileName - ( tsprg),
```

## create\_button

```
:
ac_request("create_button",string sButtonText, int sx, int sy, int ex, int ey, string
sToolTip, string sPrgCppFileName);
```

```
:
sButtonText -
sx,sy,ex,ey - LabPP_Automat,
sToolTip - ',
sPrgCppFileName - ( tsprg),
```

## set\_palette\_size\_and\_message\_place

```
LabPP_Automat
ac_request("set_palette_size_and_message_place", int x, int y, int ex, int ey, int xx,
int yy, int exx, int eyy);
```

```
:
x, y, ex, ey - ',
xx, yy, exx, eyy - .
```

## Диалоги на основе окон

```
( ( ) ).
```

## Класс ts\_dialog

```
ts_dialog object ts_dialog.
object("create", "ts_dialog", iDialogDescr); "create".
object("delete", iDialogDescr);
, .. main()
.
"set_as_main_panel".
```

## init\_dialog

```
:
int ts_dialog(int iDialogDescr,"init_dialog", string modaltype, int x, int y, int width,
int height);
:
smodaltype - "modaldialog" - ,
"palette" - ;
x, y, width, height - , .
```

## set\_as\_main\_panel

```
:
,
,
-
:
```

```
ts_dialog(int iDialogDescr, "set_as_main_panel");
```

## SetClientWH

```
ts_dialog(int iDialogDescr, "SetClientWH", int width, int height);
width, height -
```

## SetTitle

```
ts_dialog(int iDialogDescr,"SetTitle", string sTitle);
```

## SendCloseRequest

```
ts_dialog("iDialogDescr, "SendCloseRequest");
```

## PostCloseRequest

```

:
ts_dialog(iDialogDescr,"PostCloseRequest",string okorcancel);
:
okorcancel -          "ok"      "cancel" -
      "ok"      invoke      1,      0.

```

## eventreaction

```

    "Event_PanelCloseRequested" -
    -
    "eventreaction".

```

```

:
ts_dialog(iDialogDescr, "eventreaction", string event_name);

ts_dialog(iDialogDescr, "eventreaction", string event_name, string eventfunctionname );
:
event_name - . - ,
.
eventfunctionname - - .
-1 - .

```

## Invoke

```
int ts_dialog(iDialogDescr,"invoke");
```

## Руководство по LABPP\_Automat для ARCHICAD

```
ts_dialog( int iDialogDescr, string snew_sDescr);
```

```
:
```

```
iDialogDescr - , ;
```

```
snew_sDescr - ( )
```

### Класс ts\_dialogcontrol

```
ts_dialogcontrol ,
```

```
, . .
```

```
, , , .
```

```
, , , .
```

```
- ,
```

```
.
```

```
,
```

```
,
```

```
"tabpage" "normaltab".
```

### Button

```
:
```

```
ts_dialogcontrol(int iButtonDescr, "init_control", "button",int x, int y, int w, int h);
```

```
:
```

```
x,y,w,h - ,
```

### SetText

```
:
```

```
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
```

```
:
```

```
text -
```

### GetText

```
:
```

```
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
```

```
:
```

```
text - ,
```

### SetFontSizecodeStyle

```

- / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.
```

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);
```

```
:
```

```
iDialogcontrolDescr - /
```

```
.
```

```
iFontSizeCode - ,
```

```
iFontStyleCode - .
```

## IconButton

```

:
ts_dialogcontrol(int iButtonDescr, "init_control", "iconbutton",int x, int y, int w, int h);
:
x,y,w,h -

```

## TextEdit

Создает элемент панель редактирования текстовой строки

```

:
ts_dialogcontrol(int iButtonDescr, "init_control", "iconbutton",int x, int y, int w, int h);
:
x,y,w,h -

```

## SetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
:
text -

```

## GetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
:
text -

```

## SetFontSizecodeStyle

- / / / / -1/0/4/8.  
 - / / / / -1/0/16/32/48.  
 = 32+48.

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
iFontSizeCode - ,
iFontStyleCode -

```

## RealEdit

Создает элемент панель редактирования чисел

```

:
ts_dialogcontrol(int iButtonDescr, "init_control", "realedit",int x, int y, int w, int h);
:
x,y,w,h -

```

## SetFontSizecodeStyle

- / / / / -1/0/4/8.  
 - / / / / -1/0/16/32/48.

= 32+48.

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode - .

```

## CheckBox

Создает чекбокс.

```

:
ts_dialogcontrol(int iButtonDescr, "init_control", "iconbutton",int x, int y, int w, int h);
:
x,y,w,h - , .

```

## SetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
:
text - .

```

## GetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
:
text - , .

```

## SetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetCheck", int onoff);
:
iDialogcontrolDescr - .
onoff - 0/1.

```

## GetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "GetCheck", int &onoff);
:
iDialogcontrolDescr - дескриптор элемента CheckBox.
onoff - - 0/1.

```

## SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
:
= 32+48.

```

```

:
ts_dialogcontrol(int iDialogcontrolDescr, "SetFontsizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
.
iFontSizeCode - ,
iFontStyleCode - .

```

## LeftText, CenterText, RightText

```

:
ts_dialogcontrol(int iTextDescr, "init_control", "lefttext",int x, int y, int w, int h);
ts_dialogcontrol(int iTextDescr, "init_control", "centertext",int x, int y, int w, int
h);
ts_dialogcontrol(int iTextDescr, "init_control", "righttext",int x, int y, int w, int h);
:
x,y,w,h - , .

```

## SetText

```

:
ts_dialogcontrol(iDialogcontrolDescr, "SetText", string text);
:
text - .

```

## GetText

```

:
ts_dialogcontrol(iDialogcontrolDescr, "GetText", string text);
:
text - , .

```

## SetFontsizecodeStyle

```

:
- - / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.

```

```

:
ts_dialogcontrol(int iDialogcontrolDescr, "SetFontsizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
.
iFontSizeCode - ,
iFontStyleCode - .

```

## PopUp



## Создать элемент управления PopUp во всплывающем окне ARCHICAD

### SelectItem

```

:
ts_dialogcontrol(iDialogControlDescr,"SelectItem", int itemindex);
:
Itemindex =

```

### DisableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DisableItem", int itemindex);
:
itemindex - , ( 1).

```

### DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DisableItem", int itemindex);
:
itemindex - , ( 1).

```

### EnableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "EnableItem", int itemindex);
:
itemindex - , ( 1).

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - ,

```

### AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

### SetFontSizecodeStyle

- / / / / -1/0/4/8.  
 - / / / / -1/0/16/32/48.  
 = 32+48.

```
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontStyle", int iFontSizeCode, int iFontStyleCode);
```

iDialogcontrolDescr -

iFontSizeCode -

iFontStyleCode -

## GetItemIndexByText

1-n 0,

```
int ires = ts_dialogcontrol(int iPopUp, "GetItemIndexByText", string stext, int index);
```

Здесь:

iPopUp - дескриптор элемента управления;

ires - -1 если текст не найден, или 0;

index - индекс текста в списке элемента управления, или 0, если текст не найден;

stext - искомый текст в списке элемента управления, индекс которого надо получить.

## GetItemIndexByText

Для рорипу задаем текст и получаем индекс 1-базед (если не нашли - 0)

## RadioButton

### IsSelected

```
ts_dialogcontrol(iDialogcontrolDescr, "IsSelected", int ret);
```

ret - 0/1 - /

### Select

```
ts_dialogcontrol(iDialogcontrolDescr, "Select");
```

### SetText

```
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
```

text -

### GetText

```
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
```

text -





guid	Инд...	№	Этаж	Пло...	Тип ...	GUI...	Номер помещения	Н...	Наи...	ID эл...	Располо...
CDE5850E-4...	0	2	1-й э...	1.8232	Отко...	2409...	2	Ко...	Corri...	5	со сторон...
CDE5850E-4...	0	2	1-й э...	0.4386	Отко...	2409...	2	Ко...	Corri...	5	верх со ст...
C842A68A-8...	0	3	1-й э...	0.4386	Отко...	2409...	2	Ко...	Corri...	6	верх со ст...
C842A68A-8...	0	3	1-й э...	1.8232	Отко...	2409...	2	Ко...	Corri...	6	со сторон...
C842A68A-8...	0	3	1-й э...	0	Отко...	BBF7...	3	П...	Room	6	со сторон...
C842A68A-8...	0	3	1-й э...	0	Отко...	BBF7...	3	П...	Room	6	верх со ст...
6F9F6564-BC...	0	1	1-й э...	0	Отко...	BBF7...	3	П...	Room	4	верх со ст...
6F9F6564-BC...	0	1	1-й э...	0	Отко...	BBF7...	3	П...	Room	4	со сторон...
6F9F6564-BC...	0	1	1-й э...	1.8232	Отко...	2B06...	4	П...	Room	4	со сторон...
6F9F6564-BC...	0	1	1-й э...	0.4386	Отко...	2B06...	4	П...	Room	4	верх со ст...
CDE5850E-4...	0	2	1-й э...	0	Отко...	2B06...	4	П...	Room	5	верх со ст...
CDE5850E-4...	0	2	1-й э...	0	Отко...	2B06...	4	П...	Room	5	со сторон...
OEB8A882-B4...	0	4	1-й э...	0.51	Отко...	2B06...	4	П...	Room	Ок-6	верх со ст...
OEB8A882-B4...	0	4	1-й э...	1.224	Отко...	2B06...	4	П...	Room	Ок-6	со сторон...

В результате помещения визуально выделены полосками.

### SetItemBackgroundColor

Установить цвет подложки текста для заданной строки в табличном элементе диалога.

```
ts_dialogcontrol(int iListBox, "SetItemBackgroundColor", int row, int red, int green, int blue);
```

Здесь:

iListBox - дескриптор табличного элемента диалога,

row - номер строки,

red, green, blue - значения от 0 до 255 для составляющих цветов красного, зеленого и синего.

### SetItemColor

Установить цвет текста для заданной строки в табличном элементе диалога.

```
ts_dialogcontrol(int iListBox, "SetItemColor", int row, int red, int green, int blue);
```

Здесь:

iListBox - дескриптор табличного элемента диалога,

row - номер строки,

red, green, blue - значения от 0 до 255 для составляющих цветов красного, зеленого и синего.

### SelectItem

```
ts_dialogcontrol(iDialogcontrolDescr, "SelectItem", int itemindex);
```

Itemindex =

### GetTabItemText

ListBox.

```
ts_diaolgcontrol(iDialogcontrolDescr, "GetTabItemText", int item, int tabpos, string restext);
```

item - ( 1), tabpos - ( 1), restext -

## GetMouseClickedPosXY

```

                                ListBox                               Event_ListBoxClicked.
:
ts_dialogcontrol(iDialogcontrolDescr,"GetMouseClickedPosXY", int &x, int &y);
:
x,y -                        ,                        .

```

## GetTabFieldPosition

```

:
ts_dialogcontrol(iDialogcontrolDescr," GetTabFieldPosition ", int tabpos, int &begx, int
&endx);
:
tabpos - (1),
begx, endx - (
).
```

## SetTabItemIcon

```
ts_diaolgcontrol(iDialogcontrolDescr, "SetTabItemIcon", int item, int tabpos, string
path);
item - (1), tabpos - (1), path -
```

## SetTabItemText

```

:
ts_dialogcontrol(iDialogcontrolDescr, "SetTabItemText", int item, int tabpos, string
text);
:
item - (1), tabpos - (1), text -

```

## GetTabItemIconId

```

        ,
        CheckBox.
        SetTableItemIconId
        -
        :
ts_diaolgcontrol(iDialogcontrolDescr, "GetTabItemIconId", int item, int tabpos, int
iconid);
        :
item - (1), tabpos - (1), iconid -

```

## SetTabItemIconId

```

:
:
ts_diaolgcontrol(iDialogcontrolDescr, "GetTabItemIconId", int item, int tabpos, int
iconid);
:
item - (1), tabpos - (1), iconid -

```

## SetOnTabItem

```

:
:
ts_dialogcontrol(iListBoxDescr, "SetOnTabItem", int iConrtolDescr);
:
iControlDescr -

```

## SetTabFieldCount

```

:
ts_dialogcontrol(iListBoxDescr, "SetOnTabItem", int count);
:
count -

```

## SetHeaderItemSize

```

:
ts_dialogcontrol(iListBoxDescr, "SetHeaderItemSize", int height);
:
height -

```

## SetTabFieldProperties

```

:
ts_dialogcontrol(iListBox, "SetTabFieldProperties", int col, int startpix, int endpix,
int justification, int truncation);
:
col - ( 1),
startpix, endpix -
justification - ( 0, - 256, - 512),
truncation - , ( 4096).

```

## SetItemHeight

```

:
ts_dialogcontrol(iListBox, "SetItemHeight", int itemheight);
:
itemheight -

```

## SetHeaderItemSizeableFlag

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSizeableFlag", int headerindex, int sizable, int
minwidth);
:
headerindex - ( 1),
sizable - 0/1 ,
minwidth -

```

## SetHeadersSizeableFlag

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSizeableFlag", int sizable, int minwidth);
:
sizable - 0/1 ,

```

minwidth -

### SetHeaderItemText

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemText", int headerindex, string text);
:
headerindex - ( 1),
text -

```

### SetHeaderItemSize

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSize", int headerindex, int width);
:
headerindex - ( 1),
width -

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - ,

```

### AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

### DisableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DisableItem", int itemindex);
:
itemindex - , ( 1)

```

### EnableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "EnableItem", int itemindex);
:
itemindex - , ( 1)

```

### DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
:
itemindex - , ( 1). 0,

```

### SetFontSzecodeStyle

```

- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.

```

```

:
ts_diallgcontrol(int iDialogcontrolDescr, "SetFontsizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode - .

```

## MultiSellListBox

### SelectItem

```

:
ts_dialogcontrol(iDialogcontrolDescr,"SelectItem", int itemindex);
:
Itemindex = .

```

### GetTabItemText

```

:
ListBox.
:
ts_diallgcontrol(iDialogcontrolDescr, "GetTabItemText", int item, int tabpos, string
restext);
:
item - ( 1), tabpos - ( 1), restext - .

```

### GetMouseClickedPosXY

```

:
ListBox Event_ListBoxClicked.
:
ts_dialogcontrol(iDialogcontrolDescr,"GetMouseClickedPosXY", int &x, int &y);
:
x,y - , , .

```

### GetTabFieldPosition

```

:
ts_dialogcontrol(iDialogcontrolDescr," GetTabFieldPosition ", int tabpos, int &begx, int
&endx);
:
tabpos - ( 1),
begx, endx - (
).

```

### SetTabItemIcon

```

:
ts_diallgcontrol(iDialogcontrolDescr, "SetTabItemIcon", int item, int tabpos, string
path);
:
item - ( 1), tabpos - ( 1), path - .

```

### SetTabItemText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "SetTabItemText", int item, int tabpos, string
text);
:
item - ( 1), tabpos - ( 1), text -

```

### GetTabItemIconId

```

,
CheckBox.
SetTableItemIconId
-
:
ts_diaolgcontrol(iDialogcontrolDescr, "GetTabItemIconId", int item, int tabpos, int
iconid);
item - ( 1), tabpos - ( 1), iconid -

```

### SetTabItemIconId

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "GetTabItemIconId", int item, int tabpos, int
iconid);
:
item - ( 1), tabpos - ( 1), iconid -

```

### SetOnTabItem

```

:
ts_dialogcontrol(ilistBoxDescr, "SetOnTabItem", int iConrtolDescr);
:
iControlDescr -

```

### SetTabFieldCount

```

:
ts_dialogcontrol(ilistBoxDescr, "SetOnTabItem", int count);
:
count -

```

### SetTabFieldCount

```

:
ts_dialogcontrol(ilistBoxDescr, "SetOnTabItem", int count);
:
count -

```

### SetHeaderItemSize

```

:
ts_dialogcontrol(ilistBoxDescr, "SetHeaderItemSize", int height);
:
height -

```

### SetTabFieldProperties

```

:
ts_dialogcontrol(iListBox, "SetTabFieldProperties", int col, int startpix, int endpix,
int justification, int truncation);
:
col - ( 1),
startpix, endpix -
justification - ( 0, - 256, - 512),
truncation - , ( 4096).

```

### SetItemHeight

```

:
ts_dialogcontrol(iListBox, "SetItemHeight", int itemheight);
:
itemheight - .

```

### SetHeaderItemSizeableFlag

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSizeableFlag", int headerindex, int sizable, int
minwidth);
:
headerindex - ( 1),
sizable - 0/1 ,
minwidth -

```

### SetHeadersSizeableFlag

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSizeableFlag", int sizable, int minwidth);
:
sizable - 0/1 ,
minwidth -

```

### SetHeaderItemText

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemText", int headerindex, string text);
:
headerindex - ( 1),
text - .

```

### SetHeaderItemSize

```

:
ts_dialogcontrol(iListBox, "SetHeaderItemSize", int headerindex, int width);
:
headerindex - ( 1),
width - .

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - , .

```



## AppendItem

```
ts_dialogcontrol(iDialogControlDescr, "AppendItem");
```

## DisableItem

```
ts_dialogcontrol(iDialogControlDescr, "DisableItem", int itemindex);
itemindex - ( 1)
```

## EnableItem

```
ts_dialogcontrol(iDialogControlDescr, "EnableItem", int itemindex);
itemindex - ( 1)
```

## DeleteItem

```
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
itemindex - ( 1). 0,
```

## SetFontSizecodeStyle

- / / / / -1/0/4/8.  
- / / / / -1/0/16/32/48.  
= 32+48.

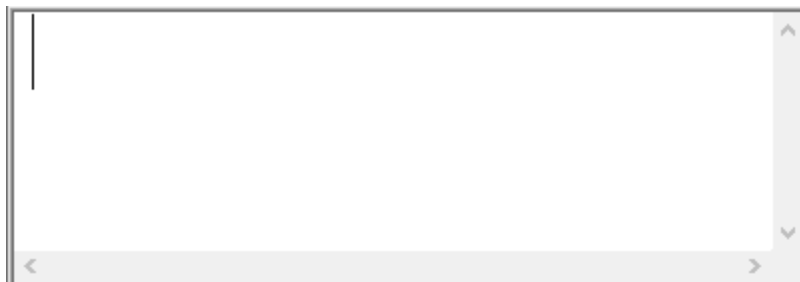
```
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int iFontStyleCode);
```

iDialogcontrolDescr - /

iFontSizeCode - ,

iFontStyleCode - .

## MultiLineEdit



## SetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
:
text -

```

## GetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
:
text -

```

## SetFontSizecodeStyle

- / / / / -1/0/4/8.  
 - / / / / -1/0/16/32/48.  
 = 32+48.

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode - .

```

## IconItem

## IconCheckBox

## SetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetCheck", int onoff);

```

```

:
iDialogcontrolDescr -
onoff - 0/1.

```

## GetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "GetCheck", int &onoff);

```

```

:
iDialogcontrolDescr - дескриптор элемента CheckBox.
onoff - - 0/1.

```

## PushCheck

### SetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "SetText", string text);
:
text -

```

### GetText

```

:
ts_diaolgcontrol(iDialogcontrolDescr, "GetText", string text);
:
text -

```

### SetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetCheck", int onoff);
:
iDialogcontrolDescr -
onoff - 0/1.

```

### GetCheck

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "GetCheck", int &onoff);
:
iDialogcontrolDescr - дескриптор элемента CheckBox.
onoff - - 0/1.

```

### SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.

```

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
iFontSizeCode - ,
iFontStyleCode -

```

## IconPushCheck

### SetCheck

```

:

```

```
ts_diaolgcontrol(int iDialogcontrolDescr, "SetCheck", int onoff);
```

```
:
iDialogcontrolDescr -
onoff - 0/1.
```

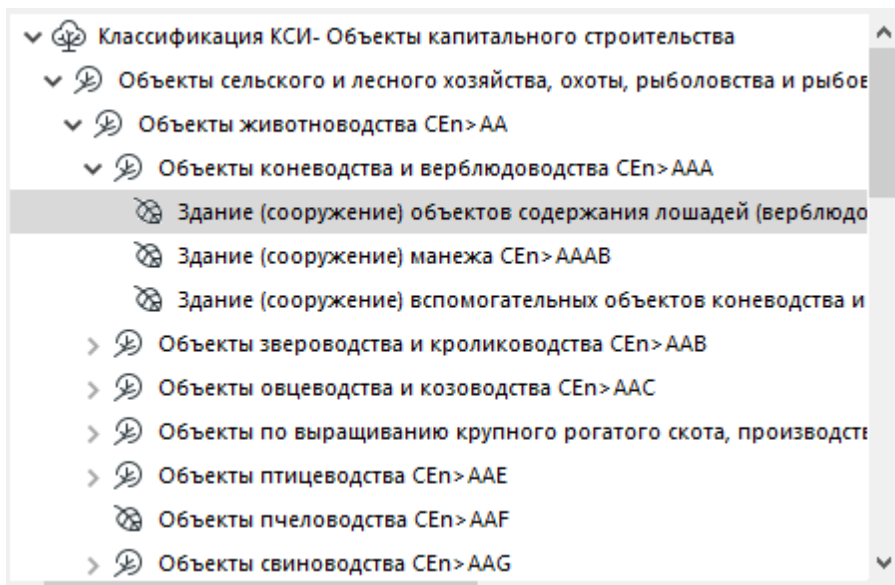
### GetCheck

```
:
ts_diaolgcontrol(int iDialogcontrolDescr, "GetCheck", int &onoff);
```

```
:
iDialogcontrolDescr - дескриптор элемента CheckBox.
onoff - - 0/1.
```

## IconPushRadio

### SingleSelTreeView



### SelectItem

```
:
ts_dialogcontrol(iDialogcontrolDescr,"SelectItem", int itemindex);
```

```
:
Itemindex =
```

### TreeViewInsertItem

#### TreeView

```
:
ts_dialogcontrol(int iDescr,"TreeViewInsertItem",int parent, int tvitem, int &resitem);
```

```
:
parent -
tvitem - RootItem (0), BotItem (-65534), TopItem (-65535).
resitem -
```

### DeleteItem

```
:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
```

```

:
itemindex - , ( 1). 0,
.

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem");

```

### SetFontSizecodeStyle

```

.
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
. = 32+48.

```

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
.
iFontSizeCode - ,
iFontStyleCode - .

```

### GetTVItemIconId

TreeView.

```

:
ts_dialogcontrol(int iTreeViewElement, "GetTVItemIconId", int tvindex, int iwhat, int
iconid);

```

```

:
iTreeViewElement - TreeView;
tvindex - TreeView;
iwhat - 0/1 - / ;
iconid - .

```

### SetTVItemIconId

TreeView.

```

:
ts_dialogcontrol(int iTreeViewElement, "SetTVItemIconId", int tvindex, int iwhat, int
iconid);

```

```

:
iTreeViewElement - TreeView;
tvindex - TreeView;
iwhat - 0/1 - / ;
iconid - .

```

### GetItemCount

```

:
ts_dialogcontrol(int iTreeView, "GetItemCount", int icount);

```

Здесь:  
iTreeView - дескриптор элемента управления TreeView;  
icount - результат - количество элементов в TreeView.

ID 32000 ( " ") TreeView.

```
int i, icount;
ts_dialogcontrol(iTreeView_DCClassSel, "GetItemCount", icount);
coutvar << icount;
for (i = 1; i <= icount; i++)
{
    ts_dialogcontrol(iTreeView_DCClassSel, "SetTVItemIconId", i, 1, 32000);
}
```

## import\_classificator

TreeView.

```
:
int err = ts_dialogcontrol(int iTreeView, "import_classificator", string
classificator_name, int idoobnul, int iTableClassGuids);
```

```
:
iTreeView - TreeView, ;
classificator_name - ;
idoobnul - 0/1 / TreeView ;
iTableClassGuids - guid- ;
Value TreeView.
TreeView "getitemvalue".
```

itemguid	string	GUID
level	int	TreeView (0- , )

TreeView

TreeView iTreeView\_DCClassSel.

```
ts_dialogcontrol(iTreeView_DCClassSel, "import_classificator", "Классификация ЛАБПП", 0,
iTableClassifGuids);
ts_dialogcontrol(iTreeView_DCClassSel, "import_classificator", "ЛАБПП-ДВЕРИ", 0,
iTableClassifGuids);
ts_dialogcontrol(iTreeView_DCClassSel, "import_classificator", "ЛАБПП-ОКНА", 0,
iTableClassifGuids);
```

## MultiSelTreeView

### SelectItem

```
:
ts_dialogcontrol(iDialogcontrolDescr, "SelectItem", int itemindex);
:
Itemindex =
```

## TreeViewInsertItem

```

TreeView
:
ts_dialogcontrol(int iDescr,"TreeViewInsertItem",int parent, int tvitem, int &resitem);
:
parent -
tvitem - RootItem (0), BotItem (-65534), TopItem (-65535).
resitem -

```

## DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
:
itemindex - , ( 1). 0,

```

## InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - ,

```

## AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

## SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.
:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);
:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode -

```

## GetTVItemIconId

```

TreeView.
:
ts_dialogcontrol(int iTreeViewElement, "GetTVItemIconId", int tvindex, int iwhat, int
iconid);
:
iTreeViewElement - TreeView;
tvindex - TreeView;
iwhat - 0/1 - / ;
iconid -

```

## SetTVItemIconId

```

TreeView.
:
ts_dialogcontrol(int iTreeViewElement, "SetTVItemIconId", int tvindex, int iwhat, int
iconid);

```

```

:
iTreeViewElement - TreeView;
tvindex - TreeView;
iwhat - 0/1 - / ;
iconid - .

```

## GetItemCount

## SingleSelListView

### DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
:
itemindex - , ( 1). 0,
.

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - , .

```

### AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

### SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
. = 32+48.

```

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
.
iFontSizeCode - ,
iFontStyleCode - .

```



## GetItemCount

## MultiSelListView

## DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
:
itemindex - , ( 1). 0,

```

## InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - ,

```

## AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

## SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.

```

```

:
ts_diaolgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode - .

```

## GetItemCount

## Splitter

## Attach\_controlled\_element

```

(
).
:
ts_dialogcontrol(int iSplitter, "attach_element_to_resize", int iControlDescr, int
leftanchor, int topanchor, int rightanchor, int bottomanchor);
:
iControlDescr - ,
leftanchor, topanchor, rightanchor, bottomanchor - -1/0/1 - . 1 -

```

/ , 0 - , -1

/ -

## ProgressBar

-



### SetMin

```

:
ts_dialogcontrol(iProgressBar, "SetMin", int minvalue);
:
minvalue -
,

```

### SetMax

```

:
ts_dialogcontrol(iProgressBar, "SetMax", int maxvalue);
:
maxvalue -
,

```

### SetValue

### GetMin

```

e
:
ts_dialogcontrol(iProgressBar, "GetMin", int &value);
:
value -

```

### GetMax

```

e
:
ts_dialogcontrol(iProgressBar, "GetMax", int &value);
:
value -

```

### GetValue

```

e
:
ts_dialogcontrol(iProgressBar, "GetValue", int &value);
:
value -

```

## NormalTab

### SelectItem

```

:
ts_dialogcontrol(iDialogcontrolDescr,"SelectItem", int itemindex);
:
Itemindex =

```

### DisableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DisableItem", int itemindex);
:
itemindex - , ( 1)

```

### EnableItem

```

:
ts_dialogcontrol(iDialogControlDescr, "EnableItem", int itemindex);
:
itemindex - , ( 1).

```

### DeleteItem

```

:
ts_dialogcontrol(iDialogControlDescr, "DeleteItem", int itemindex);
:
itemindex - , ( 1). 0,

```

### InsertItem

```

:
ts_dialogcontrol(iDialogControlDescr, "InsertItem", int itemindex);
:
itemindex - ,

```

### AppendItem

```

:
ts_dialogcontrol(iDialogControlDescr, "AppendItem");

```

### SetFontStyle

- / / / / -1/0/4/8.  
 - / / / / -1/0/16/32/48.  
 = 32+48.

```

:
ts_dialogcontrol(int iDialogcontrolDescr, "SetFontStyle", int iFontSizeCode, int
iFontStyleCode);

```

```

:
iDialogcontrolDescr - /
:
iFontSizeCode - ,
iFontStyleCode - .

```

## TabPage

### SetToolTip

```

:
:
ts_dialogcontrol(iDialogcontrolDescr, "SetToolTip", string tooltipText);
:
tooltipText -

```

## GroupBox

### SetFontSizecodeStyle

```

:
- / / / / -1/0/4/8.
- / / / / -1/0/16/32/48.
= 32+48.

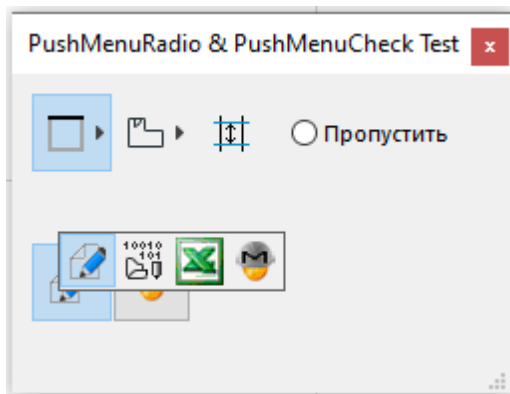
:
ts_diallgcontrol(int iDialogcontrolDescr, "SetFontSizecodeStyle", int iFontSizeCode, int
iFontStyleCode);

:
iDialogcontrolDescr - /

iFontSizeCode - ,
iFontStyleCode - .

```

## PushMenuCheck



```

//-----
// PushMenuRadio & PushMenuCheck example
// LABPP 2023
//-----

int iMenu1, iMenu11, iIconPushRadio, iMenu2, iIconPushCheck;

int main()
{
    int ires;
    ac_request("clearmessagewindow");

    // Create Dialog
    var_by_txt("init", "iDialog_DS", "int", "global", 0);
    int x, y, w, h, offset;

    object("create", "ts_dialog", iDialog_DS);

```

```

x = 0; y = 0; w = 250; h = 160;
ts_dialog(iDialog_DS, "init_dialog", "palette", x, y, w, h);
ts_dialog(iDialog_DS, "eventreaction", "Event_PanelCloseRequested",
"Event_PanelCloseRequested_DS");
ts_dialog(iDialog_DS, "set_as_main_panel");
ts_dialog(iDialog_DS, "SetGrowType", 3);

ts_dialog(iDialog_DS, "SetTitle", "PushMenuRadio & PushMenuCheck Test");

x = 10;      w = 40;      y = 10; h = 40;
object("create", "ts_dialogcontrol", iMenu1);
ts_dialogcontrol(iMenu1, "init_control", "PushMenuRadio", iDialog_DS, x, y, w, h,
1);
ts_dialogcontrol(iMenu1, "SetItemCount", 3);

string iconpath = "c:\\";
ts_dialogcontrol(iMenu1, "SetItemIcon", 1, iconpath + "files\\outlineonleft.png");
ts_dialogcontrol(iMenu1, "settooltip", "Задание линии ячейки:");
ts_dialogcontrol(iMenu1, "SetToolTipItem", 1, "Левый край");
ts_dialogcontrol(iMenu1, "SetItemIcon", 2, iconpath + "files\\outlineontop.png");
ts_dialogcontrol(iMenu1, "SetToolTipItem", 2, "Верхний край");
ts_dialogcontrol(iMenu1, "SetItemIcon", 3, iconpath + "files\\
\\outlineonright.png");
ts_dialogcontrol(iMenu1, "SetToolTipItem", 3, "Правый край");
ts_dialogcontrol(iMenu1, "eventreaction",
"Event_PushMenuChanged", "Event_PushMenuChanged_DCName_DS");
ts_dialogcontrol(iMenu1, "SetAnchorToPanelResize", 0, 1, 0, 0);

x = 50;      w = 40;      y = 10; h = 40;
object("create", "ts_dialogcontrol", iMenu11);
ts_dialogcontrol(iMenu11, "init_control", "PushMenuRadio", iDialog_DS, x, y, w, h,
1);
ts_dialogcontrol(iMenu11, "SetItemCount", 3);

ts_dialogcontrol(iMenu11, "settooltip", "Выборка по цвету пола:");
ts_dialogcontrol(iMenu11, "SetItemIcon", 1, iconpath + "files\\floor.png");
ts_dialogcontrol(iMenu11, "SetToolTipItem", 1, "Светлый пол");
ts_dialogcontrol(iMenu11, "SetItemIcon", 2, iconpath + "files\\floors.png");
ts_dialogcontrol(iMenu11, "SetToolTipItem", 2, "Средние тона");
ts_dialogcontrol(iMenu11, "SetItemIcon", 3, iconpath + "files\\floors_group.png");
ts_dialogcontrol(iMenu11, "SetToolTipItem", 3, "Темный пол");
ts_dialogcontrol(iMenu11, "SetAnchorToPanelResize", 0, 1, 0, 0);

x = 90;      w = 40;      y = 10; h = 40;
object("create", "ts_dialogcontrol", iIconPushRadio);
ts_dialogcontrol(iIconPushRadio, "init_control", "IconPushRadio", iDialog_DS, x,
y, w, h, 1);

ts_dialogcontrol(iIconPushRadio, "settooltip", "Задать высоту ячейки");
ts_dialogcontrol(iIconPushRadio, "SetIcon", iconpath + "files\\cell_height.png");
ts_dialogcontrol(iIconPushRadio, "SetAnchorToPanelResize", 0, 1, 0, 0);

x = 140; y = 18; w = 100; h = 24;
var_by_txt("init", "iRadioButton_DCName_DS", "int", "global", 0);
object("create", "ts_dialogcontrol", iRadioButton_DCName_DS);
int igrup = 1;
ts_dialogcontrol(iRadioButton_DCName_DS, "init_control", "RadioButton",
iDialog_DS, x, y, w, h, 1);
ts_dialogcontrol(iRadioButton_DCName_DS, "settooltip", "Пропустить выполнение
команды");
ts_dialogcontrol(iRadioButton_DCName_DS, "SetText", "Пропустить");
ts_dialogcontrol(iRadioButton_DCName_DS, "Select");

x = 10;      w = 40;      y = 85; h = 40;

```

```

object("create", "ts_dialogcontrol", iMenu2);
ts_dialogcontrol(iMenu2, "init_control", "PushMenuCheck", iDialog_DS, x, y, w, h);
ts_dialogcontrol(iMenu2, "SetItemCount", 4);

ts_dialogcontrol(iMenu2, "settooltip", "Показать отчет в:");
ts_dialogcontrol(iMenu2, "SetItemIcon", 1, iconpath + "files\\edit.png");
ts_dialogcontrol(iMenu2, "SetToolTipItem", 1, "Редактор текста");
ts_dialogcontrol(iMenu2, "SetItemIcon", 2, iconpath + "files\\EditPRGitem.png");
ts_dialogcontrol(iMenu2, "SetToolTipItem", 2, "Редактор программ");
ts_dialogcontrol(iMenu2, "SetItemIcon", 3, iconpath + "files\\пути_excel.png");
ts_dialogcontrol(iMenu2, "SetToolTipItem", 3, "Excel");
ts_dialogcontrol(iMenu2, "SetItemIcon", 4, iconpath + "files\\
\\LABPP_BIM_Master.png");
ts_dialogcontrol(iMenu2, "SetToolTipItem", 4, "BIM Master");
ts_dialogcontrol(iMenu2, "SetAnchorToPanelResize", 0, 1, 0, 0);
ts_dialogcontrol(iMenu2, "eventreaction", "Event_PushMenuChanged",
"Event_PushMenuChanged_DCName_DS");

x = 50;      w = 40;      y = 84; h = 41;
object("create", "ts_dialogcontrol", iIconPushCheck);
ts_dialogcontrol(iIconPushCheck, "init_control", "IconPushCheck", iDialog_DS, x,
y, w, h);

ts_dialogcontrol(iIconPushCheck, "settooltip", "Показать отчет в BIM Master");
ts_dialogcontrol(iIconPushCheck, "SetIcon", iconpath + "files\\
\\LABPP_BIM_Master.png");
ts_dialogcontrol(iIconPushCheck, "SetAnchorToPanelResize", 0, 1, 0, 0);

bool bres;
bres = ts_dialog(iDialog_DS, "invoke", bres);

return 0;
}

int Event_PanelCloseRequested_DS(int iDescr, string sDescr)
{
    int ires = tsalert(-3, "Message", "Are you sure you want to close the panel?", "",
"Ok", "Cancel");
    if (ires == 1)
    {
        return 0;    // Allow panel to close;
    }
    else
    {
        return -1;   // No close panel
    }
}

// Events
int Event_PushMenuChanged_DCName_DS(int iDescr, string sDescr)
{
    if (sDescr == "iMenu1"){
        bool bSelected;
        ts_dialogcontrol(iMenu1, "IsSelected", bSelected);
        coutvar << bSelected;
        bool bSelectionChanged;
        ts_dialogcontrol(iMenu1, "GetNamedEventValue", "WasSelectionChange",
bSelectionChanged);
        coutvar << bSelectionChanged;
        int iPreviousMenuItem;
        ts_dialogcontrol(iMenu1, "GetNamedEventValue", "PreviousMenuItem",
iPreviousMenuItem);
        coutvar << iPreviousMenuItem;
    }
}

```

```

    if (sDescr == "iMenu2") {
        bool bChecked;
        ts_dialogcontrol(iMenu2, "GetCheck", bChecked);
        coutvar << bChecked;
        bool bStateChanged;
        ts_dialogcontrol(iMenu2, "GetNamedEventValue", "WasStateChange",
bStateChanged);
        coutvar << bStateChanged;
        int iPreviousMenuItem;
        ts_dialogcontrol(iMenu2, "GetNamedEventValue", "PreviousMenuItem",
iPreviousMenuItem);
        coutvar << iPreviousMenuItem;
    }
}

```

### SetItemIcon

```

:
ts_dialogcontrol(int iPushMenu, "SetItemIcon", int index, string sIconFilePath);

```

### GetCheck

PushMenu

```

:
ts_dialogcontrol(int iPushMenuCheck, "GetCheck", bool bChecked);

```

### SetCheck

" / " PushMenuCheck.

```

:
ts_dialogcontrol(int iPushMenu, "SetCheck", bool bChecked);

```

### SetItemCount

```

:
ts_dialogcontrol(int iPushMenu, "SetItemCount", int icount);

```

### SelectItem

( - 1 n).

```

:
ts_dialogcontrol(int iPushMenuCheck, "SelectItem", int index);

```

### GetSelectedItem

```

:
ts_dialogcontrol(int iPushMenu, "GetSelectedItem", int iSelectedItem);

```

### SetToolTip

PushMenu.

```

:
ts_dialogcontrol(int iMenu1, "settooltip", string sText);

```

## SetToolTipItem

```

:
ts_dialogcontrol( int iPushMenu, "SetToolTipItem", int index, string sText);

:
iPushMenu - PushMenu,
index - , ( 1
n).
sText -

```

## WasStateChange

```

:
ts_dialogcontrol( int iPushMenu, "GetNamedEventValue", "WasStateChange", bool
bStateChanged);

```

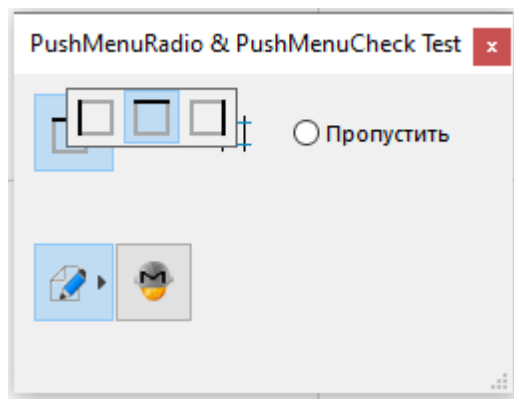
## PreviousMenuItem

```

ts_dialogcontrol( int iPushMenu, "GetNamedEventValue", "PreviousMenuItem", int
iPreviousMenuItem);

```

## PushMenuRadio




---

## SetItemIcon

```

:
ts_dialogcontrol(int iPushMenu, "SetItemIcon", int index, string sIconFilePath);

```



## Select

## IsSelected

## SetItemCount

```

:
ts_dialogcontrol(int iPushMenu, "SetItemCount", int icount);

```

## SelectItem

```

:
( - 1 n).
ts_dialogcontrol(int iPushMenu, "SelectItem", int index);

```

## GetSelectedItem

```

:
ts_dialogcontrol(int iPushMenu, "GetSelectedItem", int iSelectedItem);

```

## SetToolTip

```

:
PushMenu.
ts_dialogcontrol(int iMenu1, "settooltip", string sText);

```

## SetToolTipItem

```

:
ts_dialogcontrol( int iPushMenu, "SetToolTipItem", int index, string sText);

```

```

:
iPushMenu - PushMenu,
index - , ( 1
n).
sText -

```

## PreviousMenuItem

```

ts_dialogcontrol( int iPushMenu, "GetNamedEventValue", "PreviousMenuItem", int
iPreviousMenuItem);

```

## WasSelectionChange

```

:
ts_dialogcontrol( int iPushMenu, "GetNamedEventValue", "WasStateChange", bool
bStateChanged);

```

## Eventreaction

```

ts_dialogcontrol(int iButtonDescr, "eventreaction", string eventfunctionname [, string
eventfunctionname_more]);
:

```

```
eventfunctionname - , .
eventfunctionname_more - ,
eventfunctionname ( , ).
```

```

.
:

int iDialogDescr, iButtonZoom;
int main()
{
...// создать окно диалога
    int x, y, w, h;
    x=1,y=1, w=200; h = 200;
    object("create","ts_dialog",iDialogDescr);
    ts_dialog(iDialogDescr, "init_dialog", "palette",x,y,w,h);
    ts_dialog(iDialogDescr, "eventreaction", "Event_PanelCloseRequested");
    ts_dialog(iDialogDescr, "SetTitle","Расчет ведомости объемов работ");
    bool bres;
    // создать элемент управления
    x=1; y=1; w=50; h=20;
    object("create","ts_dialogcontrol",iButtonZoom,"iButtonZoom");
    ts_dialogcontrol(iButtonZoom, "init_control", "button",iDialogDescr, x, y, w, h);
    ts_dialogcontrol(iButtonZoom, "eventreaction", "Event_ButtonClicked");
    ts_dialogcontrol(iButtonZoom, "settext", "Показать");
...
    ts_dialog(iDialogDescr, "invoke",bres);
    cout << bres;
}

int Event_ButtonClicked(int iDescr, string sDescr)
{
    if(sDescr == "ButtonCancel")
    {
        ts_dialog(iDialogDescr,"PostCloseRequest","cancel"); // закрыть диалог с
результатом отмены
    }
    else if(sDescr == "ButtonOK")
    {
        ts_dialog(iDialogDescr,"PostCloseRequest","ok"); // закрыть диалог с результатом
"Ok"
    }
    else if(sDescr == "ButtonCalc")
    {
        Calc();
    }
    else if(sDescr == "ButtonZoom")
    {
        ZoomElementInProject();
    }
}
}

```

```

iDescr - ,
sDescr - ( ).

```

```

if(iDescr == iDescrZoom)
{
    //
}
Либо
if(sDescr == "iButtonZoom")
{

```

```
//
}
```

Event_TreeViewContextMenuRequested	MultiselTreeView, SingleSelTreeView	
Event_TreeViewItemCollapsed		
Event_TreeViewItemDoubleClicked		
Event_TreeViewItemExpanded		
Event_TreeViewLabelEditFinished		
Event_TreeViewLabelEditStarted		
Event_TreeViewSelectionChanged		
Event_TreeViewStateIconClicked		
Event_ListViewContextMenuRequested	MultiselListView, SingleSelListView	
Event_ListViewDoubleClicked		
Event_ListViewItemUpdate		
Event_ListViewSelectionChanged		
Event_RealEditChanged	RealEdit	
Event_TextEditChanged	TextEdit	
Event_ItemFocusGained		
Event_ItemFocusLost		
Event_ImageClicked	IconItem	
Event_ListBoxClicked	MultiselListBox, SingleSelListBox	
Event_ListBoxDoubleClicked		
Event_ListBoxSelectionChanged		
Event_PopUpChanged	PopUp	PopUp
Event_StaticTextClicked	LeftText, CenterText, RightText	
Event_StaticTextDoubleClicked		
Event_CheckItemChanged	CheckBox, IconCheckBox, IconPushCheck, PushCheck	
Event_CheckItemDoubleClicked		



```

:
iJSONObjectDescr - JSON, .

```

## DecodeFromURL

```

JSON , web .
:
int ires = ts_json(int iJSONObjectDescr, "DecodeFromURL", string sURL);

```

```

:
iJSONObjectDescr - JSON,
sURL - .
0 .
JSON config.json www.appcarrier.com

```

```

int iJSONConfig; // дескриптор объекта
object("create", "ts_json", iJSONConfig); // создать объект в памяти
int ires = ts_json(iJSONConfig, "DecodeFromURL",
"http://www.appcarrier.com/config.json");
if (ires != 0)
{
    cout << "JSON Decode from URL error";
}
else
{
    cout << "File JSON decoded successfully\n";
}
object("delete", iJSONConfig); // удалить объект

```

## DecodeFromString

```

JSON .
:
int ires = ts_json(int iJSONObjectDescr, "DecodeFromString", string sJSON);

```

```

:
iJSONObjectDescr - JSON, ,
sJSON - , JSON.
0 .
JSON .

```

```

int iJSONConfig; // дескриптор для объекта ts_json
object("create", "ts_json", iJSONConfig); // создать объект в памяти

string sJSON = "{ \"My class in JSON code\":{\
    \"StringVariable\": \"my string\", \
    \"DoubleVariable\" : 1.23\
}}";

int ires = ts_json(iJSONConfig, "DecodeFromString", sJSON);
if (ires != 0)
{
    cout << "JSON Decode from URL error";
}
else
{
    cout << "File JSON decoded successfully\n";
}
object("delete", iJSONConfig); // удалить объект

```

## EncodeToStr

```

        JSON
        :
int ires = ts_json(int iJSONObjectDescr, "EncodeToStr", string &sJSON);

        :
iJSONObjectDescr - JSON,
sJSON - JSON
        0
        JSON

string sJSON;

int ires = ts_json(iJSONConfig, "EncodeToStr", sJSON);
coutvar << sJSON; // вывести содержимое переменной sJSON в окно сообщений

```

## Класс ts\_array

## Класс ts\_class

## Класс ts\_num

## Класс ts\_bool

## Класс ts\_string

## Руководство по LABPP\_Automat для ARCHICAD

```

int iStringDescr;
object("create", "ts_string", iStringDescr);
int iret = ts_string(iStringDescr, "ReadFromURL", "http://www.labpp.ru/TSDownload/KSI/" +
sclassifname + ".txt");
object("read_simple_value", iStringDescr);
string sXMLsource = ac_getstrvalue();

```

## TSVariator

```

int ires = ts_json(int iJSON, "tsvariator", "apply", int what_answer_if_notexist);
Здесь:

```

ires - результат команды, если 0 - значит без ошибок и значение искомого свойства задано. В ином случае - либо ошибка, либо искомого свойства нет в тексте.  
 what\_answer\_if\_notexist - что вернуть при отсутствии запрашиваемого свойства. Если 0 - то вернуть содержимое переменной запроса. Если 1 - то вернуть значение из свойства "По умолчанию".

TSVariator:

```

{ "TSVariator":
[
{"Result": "Нет",

```

```

"SourceVarValue" : "По умолчанию",
"Comment" : "По умолчанию, если нет параметра"
},
{
"SourceVarValue": "Ширина профиля",
"Result" : "50",
"Comment" : ""
},
{
"SourceVarValue": "Толщина утеплителя",
"Result" : "50",
"Comment" : ""
},
{
"SourceVarValue": "Огнестойкость",
"Result" : "Нет",
"Comment" : "для отметки огнестойкости пп у элемента"
},
{
"SourceVarValue": "Влагостойкость",
"Result" : "Нет",
"Comment" : "для отметки влагостойкости пп у элемента"
}],
"SourceVarName": "Свойство"
}

```

В квадратных скобках заключен массив классов переменных.

Каждая переменная имеет имя ("SourceVarValue"), примечание ("Comment") и значение ("Result").

"SourceVarName" - имя внешней переменной в коде скрипта, которая используется для получения данных.

В примере имя этой переменной - "Свойство";

Весь текст можно поместить в одну ячейку электронной таблицы и получать из нее значения различных свойств по их названиям ("Ширина профиля", "Толщина профиля" и т.д.). Если запрашиваемого свойства в тексте нет, то выдается значение из первой записи (в примере это значение - "Нет").

Пример.

Получить значение свойства "Огнестойкость" из текста в примере выше.

```

string sexample; // переменная в которой находится текст из примера выше
var_extern_set("Свойство", "Огнестойкость"); // записать во внешнюю переменную с именем
"Свойство" имя свойства, которое хотим считать из текста.
int iJSON;
object("create", "ts_json", iJSON); // создадим объект типа ts_gson
string sresult;
ts_json(iJSON, "DecodeFromString", sexample);
ires = ts_json(iJSON, "tsvariator", "apply", 1); // если не найдена переменная - вернуть
значение по умолчанию                                sresult = ac_getstrvalue();
cout << "Значение огнестойкости = " << sresult << "\n";
object("delete", iJSON); // удалить объект

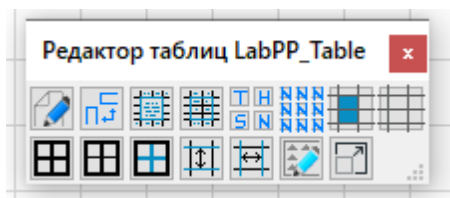
```

## Редактор таблиц LABPP\_Table для ARCHICAD

Номер	Наименование	Отделка стен	Площадь, м²	
		1. Дизлектрическое покрытие 10 <sup>6</sup> ом; 2. Грунтовка бесцветная; 3. Выравнивающий слой - шпатлевка на цементном вяжущем; 4. Грунтовка бесцветная	53,5	
10	Электро-щитовая	1. Водно-дисперсионная; 2. Финишный слой грун 3. Шпатлевка на цеме 4. Грунтовка; 5. Шпатлевка на цеме 33699-2015 фракцией 6. Грунтовка АК-0 бесцветная ГОСТ 9825-73; 7. Стена из кирпича силикатного		

GDL      [LABPP\\_Table.gsm](#).  
LABPP\_Table\_unicode.cpp

([LABPP Table unicode.zip](#)).



```

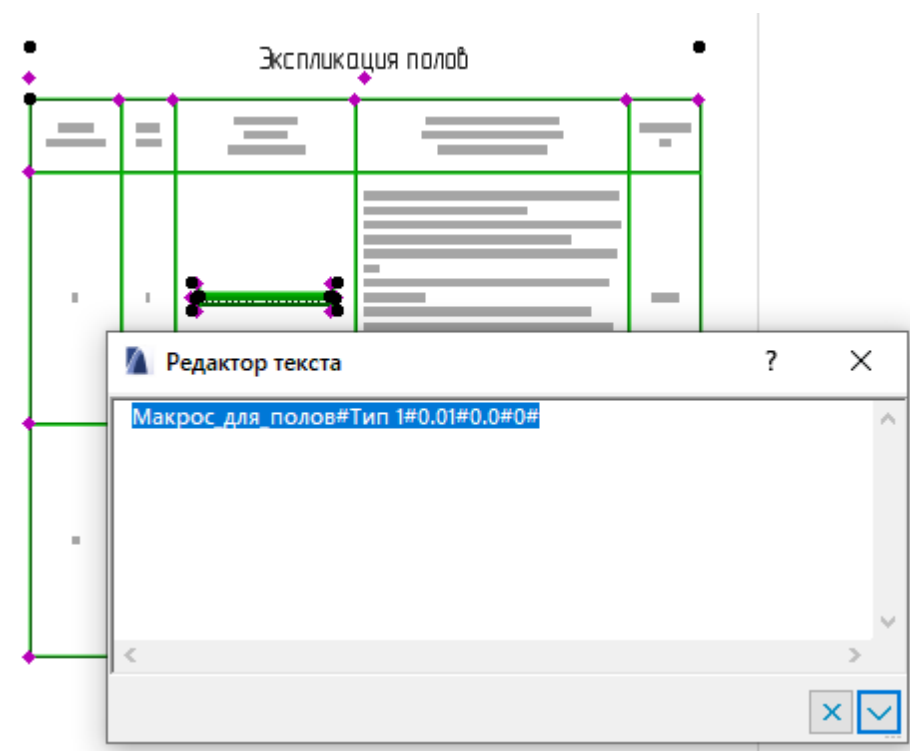
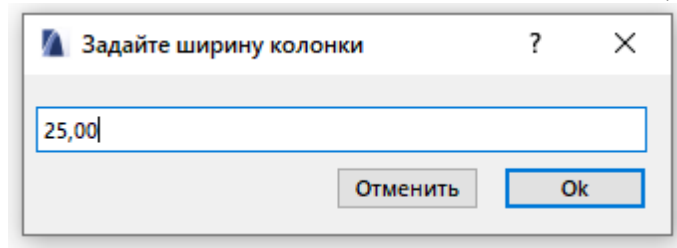
- 100;
- 500;
- ;
- 2 ( , );
- ;
- ;
- ;
- 4;
- , , , ;
;
UNICODE - ( 2 2);
- (9 );
- ;
- ( , );

```

LabPP\_Table\_unicode.cpp.







# # #  
# # #

2d GDL

TSScaleK.

```

!-----
IF LABPP_PictureName = " 1" THEN
  IF LABPP_Command = "GET SIZES" THEN
    ! x,y,w,h
    EXIT 0.951, -16.940, 3.9937, 0.751
  ENDIF

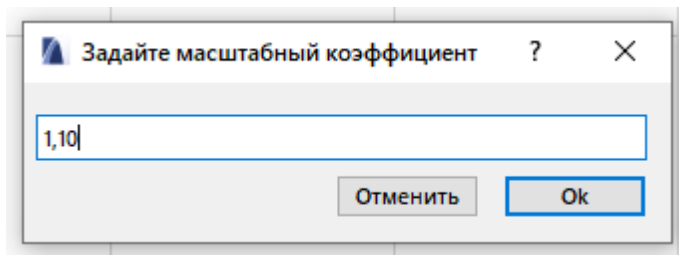
  fillAttribute_1 = 214
  fillAttribute_2 = 330

  lineTypeAttribute_1 = 1
  lineTypeAttribute_2 = 0

  pen penAttribute_1
  fill fillAttribute_1
  poly2_b{5} 5, 2, 1, 8, penAttribute_1, penAttribute_2,
    2.947696588428, -16.52173742916, TSScaleK, 0, 0,
TSScaleK, 0,
    1.09620462548, -16.47923742916, 33,
    1.09620462548, -16.56423742916, 33,
    4.799188551377, -16.56423742916, 33,
    4.799188551377, -16.47923742916, 33,
    1.09620462548, -16.47923742916, 33
  set line_type lineTypeAttribute_1
  pen penAttribute_4
  poly2_b{5} 7, 1, 0, 3, 1, 0,
    0, 0, TSScaleK, 0, 0,
TSScaleK, 0,
    1.22479041471, -16.51429212963, 1,
    1.202131508217, -16.51429212963, 900,
    0, 180, 4001,
    1.179472601724, -16.51429212963, 1,
    1.202131508217, -16.51429212963, 900,
    0, 180, 4001,
    1.22479041471, -16.51429212963, 1
ENDIF

```

/ ( 1 - , 1 - ).



Структура данных объекта LABPP\_Table (для программистов)

LABPP\_Table\_unicode.cpp.

**TS\_Cells\_poscode** -  
500 100

4 4-  
LTRB - ,  
- - ,  
[2][1] [3][1] ( 1- 2 3)  
LTR\_ : L\_RB.  
5  
'a' - 90  
6  
't', 'h', 's', 'n' -  
7- - 0- , 1 - , 2 - 1 , 3 - 2  
8- - l/m/f/r / / /  
9- - t/m/b - / /  
10,11,12,13 - - / / / -> 0/1/2 - / /

LTRB n0mm1111

**TS\_Cells\_value** -  
500 100

**TS\_Cells\_macro** -  
500 100

**TS\_ColumnsUsedCount** -  
**TS\_RowsUsedCount** -

**TS\_RowsH** -  
**TS\_ColumnsW** -

**TS\_ShowSelection** -

**TS\_DoRecalcAll** 1.

LABPP\_Table

**TS\_RealCells** -  
N x 17.

.. 17 - , 17 - ..

1 - SX

2 - SY

3 - EX

4 - EY

5 - 1 -

6 - J -

7 -

8 -

9 -

10 -

11 -

12 -

13 - -

14 - -

15 -

16 -

17 - selected

TS\_RealCellsSelected -

17-

TS\_RealCells

17-

TS RealCells.

9

## Непереносимые пробелы и дефисы

,

;

•

LABPP\_Table

—

Alt

•

0160

- Alt+8211.

		1	2	3					
Alt+	8304	0185	0178	0179	8308	8309	8310	8311	
Unicode	2070	00B9	00B2	00B3	2074	2075	2076	2077	

# Ведомость

Наименование	Площадь	Кол-во
Блоки оконные из алюминиевых сплавов ГОСТ 21519-2003	1,44	4
Плитка 600x600 бежевая	1,400	4
Железобетон	0,36	1

## Работа с СУБД MySQL



MySQL

MySQL

LABPP.

1. libmysql.dll
2. libcrypto-3-x64.dll
3. libssl-3-x64.dll

ARCHICAD.

### connection\_init

MySQL ARCHICAD

```

:
int ires = ts_dbmysql(int iMySQL, "connection_init");
:
ires = -1/0 - /
iMySQL - .
:

```

```

int iMySQL;
object("create", "ts_dbmysql", iMySQL);
int ires;

```

```

ires = ts_dbmysql(iMySQL, "connection_init");
if (ires != 0) {
    return -1;
}
cout << "init success\n";

```

## connection\_open

```

(
).

:
int ires = ts_dbmysql(iMySQL, "connection_open", string server, string user, string
password, string basename, int port);

```

```

:
ires =
iMySQL -
server - IP ,
user -
password -
basename - ( MySQL
port - ( 3306);

:

string server = "31.41.196.220";
string user = "u0490570";
string password = "hP2uH9fdsaY2zpA5eA4";
string basename = "u0490570";
int port = 3306;
ires = ts_dbmysql(iMySQL, "connection_open", server, user, password, basename, port);
if (ires != 0) {
    return -1;
}
cout << "connection success\n";

```

## real\_query

```

.
unicode (
ts_table.

:
int ires = ts_dbmysql(iMySQL, "real_query", string query);

:
ires =
iMySQL -
query - SQL ( SELECT * FROM u0490570.my_table);

:

string query = "SELECT * FROM u0490570.my_table"; // запросить все записи из таблицы
my_table в базе данных u0490570
int j = 0;
ires = ts_dbmysql(iMySQL, "real_query", query);
if (ires != 0) {
    return -1;
}
cout << "query success\n";

```

## store\_result

```

        ,
        ,
        ts_table.

:
int ires = ts_dbmysql(iMySQL, "store_result");

:
ires = -1/0 - / ;
iMySQL - ;

:

ires = ts_dbmysql(iMySQL, "ires = ts_dbmysql(iMySQL, \"store_result\");
if (ires != 0) {
    return -1;
}
cout << "store result success\n");

```

## get\_num\_fields

```

:
int ires = ts_dbmysql(iMySQL, "get_num_fields");

:
ires = -1/0 - / ;
iMySQL - ;

:

ires = ts_dbmysql(iMySQL, "get_num_fields");
cout << "num fields = " << ires << "\n";

```

## get\_row\_count

```

:
int ires = ts_dbmysql(iMySQL, "get_row_count", count);

:
ires = -1/0 - / ;
iMySQL - ;
count - .

:

int count;
ires = ts_dbmysql(iMySQL, "get_row_count", count);
cout << "row count = " << count << "\n";

```

## export\_to\_table

```

        ,
        ,
        ts_table.

:
int ires = ts_dbmysql(iMySQL, "export_to_table", int iTable);

```



```

:
ires = -1/0 - / ;
iMySQL - ;
iTable - ts_table, .

```

```

:

int iTable;
object("create", "ts_table", iTable);

ts_dbmysql(iMySQL, "export_to_table", iTable);

```

## free\_result

```

.
-

:

int ires = ts_dbmysql(iMySQL, "free_result");

```

```

:
ires = -1/0 - / ;
iMySQL - ;

:
// освободить память результата запроса в подключении у объекта iMySQL
ts_dbmysql(iMySQL, "free_result");

```

## connection\_close

```

,
.

:

int ires = ts_dbmysql(iMySQL, "connection_close");

:
ires = -1/0 - / ;
iMySQL - ;

:
ts_dbmysql(iMySQL, "connection_close");

```

## Пример - получение таблицы с сайта из базы данных MySQL в диалог ARCHICAD

```

,
MySQL.

:
IP - 31.41.196.220 ( )
- u0490570
- hP2uH9fdsaY2zpA5eA4
( ) - u0490570
( ) - 3306;

```

Наименования материалов для заказчика			
№	Идентификатор	Наименование для заказчика	Идентификатор
12317	0xB578B49691C4...	Кабель с алюминиевыми жилами AcBBГнг(A)-LS 5x4	0xB569B49691C
12318	0xB578B49691C4...	Кабель с алюминиевыми жилами AcBBГнг(A)-LS 5x6	0xB569B49691C
12319	0xB578B49691C4...	Труба ПЭ100 SDR17-90x5,4	0xB577B49691C
12320	0xB578B49691C4...	Труба ПЭ100 SDR17-225x13,4	0xB577B49691C
12321	0xB578B49691C4...	Труба ПЭ100 SDR17-110x6,6	0xB577B49691C
12322	0xB578B49691C4...	Труба ПЭ100 SDR17-400x23,7	0xB577B49691C
12323	0xB578B49691C4...	Бетон B35 (M450) W6 F150	0xB569B49691C
12324	0xB578B49691C4...	Бетон B25 (M350) W6 F75	0xB569B49691C
12325	0xB578B49691C4...	Бетон B35 (M450) W4 F100	0xB569B49691C
12326	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1100 с нижним подкл...	0xB569B49691C
12327	0xB578B49691C4...	Утеплитель минераловатный 135кг/м3 200мм	0xB569B49691C
12328	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1100 с боковым подк...	0xB569B49691C
12329	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1400 с боковым подк...	0xB569B49691C
12330	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1400 с нижним подкл...	0xB569B49691C

```

int iListBox_MD, iButton_Ok_MD;
int iTable_MD;
int iDialog_MD;

int main()
{
    int iMySQL;
    object("create", "ts_dbmysql", iMySQL);
    int ires;
    ires = ts_dbmysql(iMySQL, "connection_init");
    if (ires != 0) {
        return -1;
    }
    cout << "init success\n";
    string server = "31.41.196.220";
    string user = "u0490570";
    string password = "hP2uH9fdsaY2zpA5eA4";
    string basename = "u0490570"; // scheme
    int port = 3306;
    ires = ts_dbmysql(iMySQL, "connection_open", server, user, password, basename,
port);
    if (ires != 0) {
        return -1;
    }
    cout << "connection success\n";

    // Query
    // string query = "SELECT * FROM u0490570_nsi.labpp_mat_table WHERE
labpp_mat_short_text = \"'Материал м² 2'\"";
    // string query = "SELECT * FROM u0490570_nsi.labpp_mat_table WHERE
labpp_mat_short_text LIKE \"'%м²%'\"";
    string query = "SELECT * FROM u0490570.my_table";

```

```

int j = 0;

ires = ts_dbmysql(iMySQL, "real_query", query);
if (ires != 0) {
    return -1;
}
cout << "query success\n";

ires = ts_dbmysql(iMySQL, "store_result");
if (ires != 0) {
    return -1;
}
cout << "store result success\n";

ires = ts_dbmysql(iMySQL, "get_num_fields");
cout << "num fields = " << ires << "\n";

int count;
ires = ts_dbmysql(iMySQL, "get_row_count", count);
cout << "row count = " << count << "\n";

int iTable;
object("create", "ts_table", iTable);

ts_dbmysql(iMySQL, "export_to_table", iTable);

ts_dbmysql(iMySQL, "free_result");
ts_dbmysql(iMySQL, "connection_close");
object("delete", iMySQL);

// Create Dialog
int x, y, w, h, offset;

object("create", "ts_dialog", iDialog_MD);
ts_dialog(iDialog_MD, "init_dialog", "palette", 0, 0, 350, 400);
ts_dialog(iDialog_MD, "set_as_main_panel");
ts_dialog(iDialog_MD, "SetGrowType", 3);

ts_dialog(iDialog_MD, "SetTitle", "Наименования материалов для заказчика");

object("create", "ts_dialogcontrol", iListBox_MD, "iListBox_MD");

ts_dialogcontrol(iListBox_MD, "init_control", "singleselectlistbox", iDialog_MD, 5,
5, 340, 350, 48, 20);
ts_dialogcontrol(iListBox_MD, "SetAnchorToPanelResize", 0, 0, 1, 1);

ts_table(iTable, "set_columns_width_when_export", 100);

ts_table(iTable, "mapping_columns_to_export",
    "id", "№", "",
    "_IDRRef", "Идентификатор", "",
    "_Description", "Наименование для заказчика", "",
    "_ParentIDRRef", "Идентификатор родителя", ""
);
ts_table(iTable, "export_to_dialogcontrol", iListBox_MD, -1, -1);

x = 10;
w = 98;
int delta = 10;
x = x + w + delta;
y = 370;
h = 20;

```

```

    x = x + w + delta;
    object("create", "ts_dialogcontrol", iButton_Ok_MD, "iButton_Ok_MD");
    ts_dialogcontrol(iButton_Ok_MD, "init_control", "button", iDialog_MD, x, y, w, h);
    ts_dialogcontrol(iButton_Ok_MD, "eventreaction", "Event_ButtonClicked",
"Event_ButtonClicked_MD");
    ts_dialogcontrol(iButton_Ok_MD, "settext", "Ok");
    ts_dialogcontrol(iButton_Ok_MD, "SetAnchorToPanelResize", 1, 1, 0, 0);

    bool bres;
    ts_dialog(iDialog_MD, "invoke", bres);

    //<-
    object("delete", iTable);
}

int Event_ButtonClicked_MD(int iDescr, string sDescr)
{
    if (sDescr == "iButton_Ok_MD") {
        ts_dialog(iDialog_MD, "SendCloseRequest");
    }
}

```

## Работа с СУБД PostgreSQL

PostgreSQL



PostgreSQL

LABPP.

PostgreSQL :

1. libiconv-2.dll
2. libintl-9.dll
3. libpq.dll
4. libwinpthread-1.dll

ARCHICAD.

### init

PostgreSQL ARCHICAD

```

:
int ires = ts_dbpostgresql(int iPostgreSQL, "init");
:
ires = -1/0 - /
iPostgreSQL - .
:

```

```

int iPostgreSQL;
object("create", "ts_dbpostgresql", iPostgreSQL);
int ires;
ires = ts_dbpostgresql(iPostgreSQL, "init");
if (ires != 0) {
    return -1;
}
cout << "init success\n";

```

## pqconnectdb

```

        (
    ):
int ires = ts_dbpostgresql(iPostgreSQL, "pqconnectdb", string connectstring);

    :
ires = -1/0 - / ;
iPostgreSQL - ;
connectstring - ;

    :

ires = ts_dbpostgresql(iPostgreSQL, "pqconnectdb", "user=postgres password=Ts0020010
host=localhost port=5432 dbname=postgres target_session_attrs=read-write");
if (ires != 0) {
    return -1;
}
cout << "connection success\n";

```

## pqexec

```

        .
        , unicode ( M²).
        , ts_table.

    :
int ires = ts_dbpostgresql(iPostgreSQL, "pqexec", string query);

    :
ires = -1/0 - / ;
iPostgreSQL - ;
query - SQL;

    :

string query = "DECLARE myportal CURSOR FOR select * from pg_type"; // запросить все
записи из таблицы pg_type и создать курсор myportal
int j = 0;
ires = ts_dbpostgresql(iPostgreSQL, "pqexec", query);
if (ires != 0) {
    return -1;
}
cout << "query success\n";

```

## pqnfields

```

        .

    :
int ires = ts_dbpostgresql(iPostgreSQL, "pqnfields");

    :
ires = -1/0 - / ;
iPostgreSQL - ;

    :

ires = ts_dbpostgresql(iPostgreSQL, "pqnfields");

```

```
cout << "num fields = " << ires << "\n";
```

## pqntuples - кол-во строк

```
int ires = ts_dbpostgresql(iPostgreSQL, "pqntuples", count);
```

```
ires = -1/0 - / ;
iPostgreSQL - ;
count - .
```

```
int count;
ires = ts_dbpostgresql(iPostgreSQL, "pqntuples", count);
cout << "row count = " << count << "\n";
```

## export\_to\_table

```
, , ts_table.
```

```
int ires = ts_dbpostgresql(iPostgreSQL, "export_to_table", int iTable);
```

```
int ires = ts_dbpostgresql(iPostgreSQL, "export_to_table", int iTable, int/string
colname1, string labppcoltype1, int/string colname2, string labppcoltype2, ...);
```

```
ires = -1/0 - / ;
iPostgreSQL - ;
iTable - ts_table, .
int/string colname -
```

```
int iTable;
object("create", "ts_table", iTable);
// вариант 1
ts_dbpostgresql(iPostgreSQL, "export_to_table", iTable);
// вариант 2
ts_dbpostgresql(iPostgreSQL, "export_to_table", iTable,
"id","string","count","int",5,"double");
```

## pqclear

```
int ires = ts_dbpostgresql(iPostgreSQL, "pqclear");
```

```
ires = -1/0 - / ;
iPostgreSQL - ;
```

```

:
// освободить память результата запроса в подключении у объекта iMySQL
ts_dbpostgresql(iPostgreSQL, "clear");

```

## pqfinish

```

:
int ires = ts_dbpostgresql(iPostgreSQL, "pqfinish");

```

```

:
ires = -1/0 - / ;
iPostgreSQL - ;

```

```

:
ts_dbpostgresql(iPostgreSQL, "pqfinish");

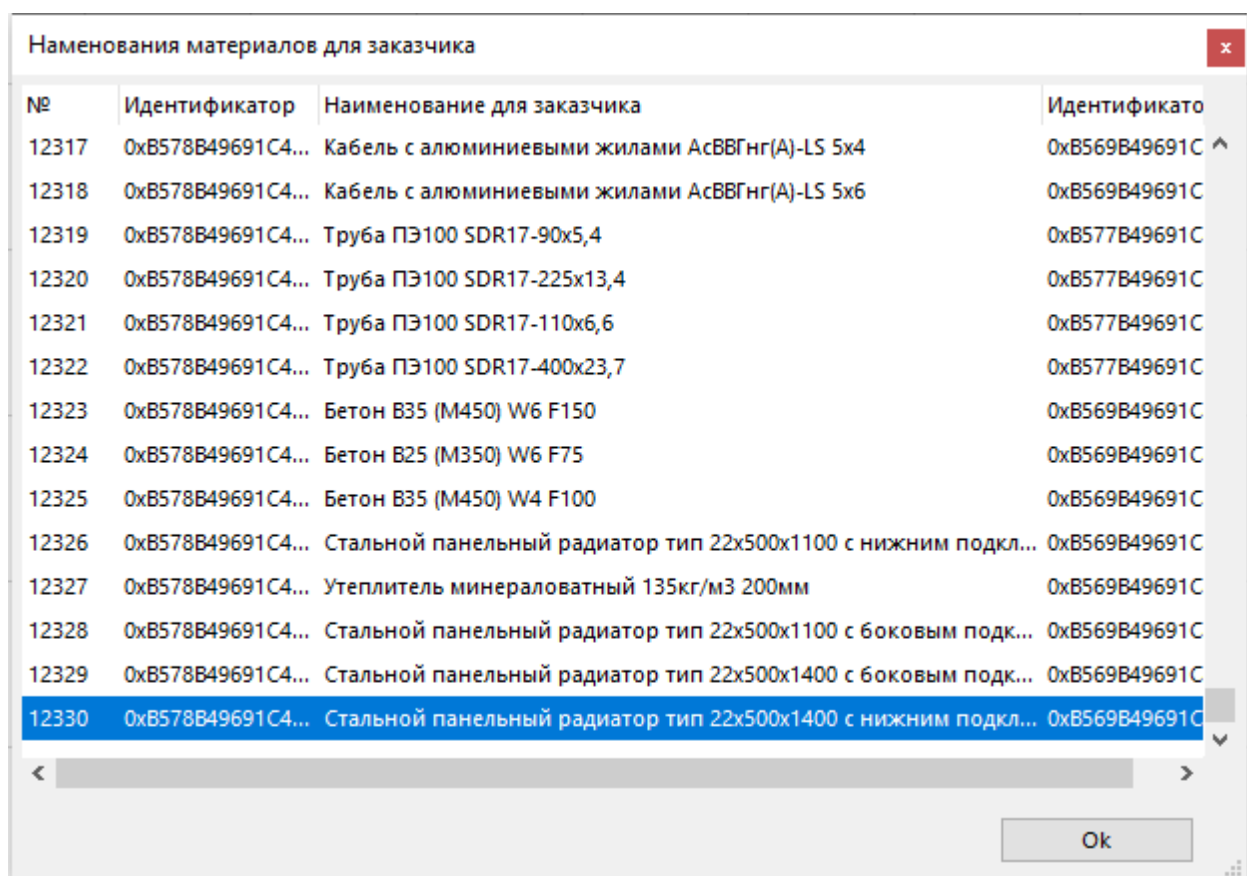
```

## Пример - получение таблицы с сайта из базы данных PostgreSQL в диалог ARCHICAD

```

: PostgreSQL.
- localhost
- labpp_user_ivanov
- pspsp
( ) - LABPP_ERM
( ) - 5432;

```



```

int iListBox_MD, iButton_Ok_MD;
int iTable_MD;

```

```

int iDialog_MD;

int main()
{
    int iPostgreSQL;
    object("create", "ts_dbpostgresql", iPostgreSQL);
    int ires;
    ires = ts_dbpostgresql(iPostgreSQL, "init");
    if (ires != 0) {
        return -1;
    }
    cout << "init success\n";

    ires = ts_dbpostgresql(iPostgreSQL, "pqconnectdb", "user=labpp_user_ivanov
password=pspsps host=localhost port=5432 dbname=LABPP_ERM target_session_attrs=read-
write");
    if (ires != 0) {
        return -1;
    }
    cout << "connection success\n";

    ires = ts_dbpostgresql(iPostgreSQL, "pqexec", "BEGIN");
    if (ires != 0) {
        return -1;
    }
    cout << "begin transaction success\n";

    ts_dbpostgresql(iPostgreSQL, "pqclear");

    ires = ts_dbpostgresql(iPostgreSQL, "pqexec", "DECLARE myportal CURSOR FOR select
* from pg_type");
    if (ires != 0) {
        return -1;
    }
    cout << "creating cursor success\n";
    ts_dbpostgresql(iPostgreSQL, "pqclear");

    ires = ts_dbpostgresql(iPostgreSQL, "pqexec", "FETCH ALL in myportal");
    if (ires != 0) {
        return -1;
    }
    cout << "fitch all success\n";

    int nfields = ts_dbpostgresql(iPostgreSQL, "pqnfields");
    coutvar << nfields;

    int ntuples = ts_dbpostgresql(iPostgreSQL, "pqntuples");
    coutvar << ntuples;
    // Считываем в таблицу
    int iTable;
    object("create", "ts_table", iTable);
    ts_dbpostgresql(iPostgreSQL, "export_to_table", iTable);
    string sstr;
    ts_table(iTable, "print_to_str", sstr);
    coutvar << sstr;
    //
    ts_dbpostgresql(iPostgreSQL, "pqclear");

    ires = ts_dbpostgresql(iPostgreSQL, "pqexec", "CLOSE myportal");
    if (ires != 0) {
        return -1;
    }
    cout << "close portal success\n";
}

```



```

ts_dbpostgresql(iPostgreSQL, "pqclear");

ires = ts_dbpostgresql(iPostgreSQL, "pqexec", "END");
if (ires != 0) {
    return -1;
}
cout << "end transaction success\n";

ts_dbpostgresql(iPostgreSQL, "pqclear");

// Create Dialog
int x, y, w, h, offset;

object("create", "ts_dialog", iDialog_MD);
ts_dialog(iDialog_MD, "init_dialog", "palette", 0, 0, 350, 400);
ts_dialog(iDialog_MD, "set_as_main_panel");
ts_dialog(iDialog_MD, "SetGrowType", 3);

ts_dialog(iDialog_MD, "SetTitle", "Наименования материалов для заказчика");

object("create", "ts_dialogcontrol", iListBox_MD, "iListBox_MD");

ts_dialogcontrol(iListBox_MD, "init_control", "singleselectlistbox", iDialog_MD, 5,
5, 340, 350, 48, 20);
ts_dialogcontrol(iListBox_MD, "SetAnchorToPanelResize", 0, 0, 1, 1);

ts_table(iTable, "set_columns_width_when_export", 100);

ts_table(iTable, "export_to_dialogcontrol", iListBox_MD, -1, -1);

x = 10;
w = 98;
int delta = 10;
x = x + w + delta;
y = 370;
h = 20;

x = x + w + delta;
object("create", "ts_dialogcontrol", iButton_Ok_MD, "iButton_Ok_MD");
ts_dialogcontrol(iButton_Ok_MD, "init_control", "button", iDialog_MD, x, y, w, h);
ts_dialogcontrol(iButton_Ok_MD, "eventreaction", "Event_ButtonClicked",
"Event_ButtonClicked_MD");
ts_dialogcontrol(iButton_Ok_MD, "settext", "Ok");
ts_dialogcontrol(iButton_Ok_MD, "SetAnchorToPanelResize", 1, 1, 0, 0);

bool bres;
ts_dialog(iDialog_MD, "invoke", bres);

//<-
object("delete", iTable);
object("delete", iPostgreSQL);
}

int Event_ButtonClicked_MD(int iDescr, string sDescr)
{
    if (sDescr == "iButton_Ok_MD") {
        ts_dialog(iDialog_MD, "SendCloseRequest");
    }
}

```

## Работа с СУБД SQLite



SQLite      sqlite3.dll      SQLite      ARCHICAD.      LABPP.

### init

SQLite      ARCHICAD

```

:
int ires = ts_dbsqlite(int iSQLite, "init");
:
ires = -1/0 - /
iSQLite - .
:

```

```

int iSQLite;
object("create", "ts_dbsqlite", iSQLite);
int ires;
ires = ts_dbsqlite(iSQLite, "init");
if (ires != 0) {
    return -1;
}
cout << "init success\n";

```

### open\_db

( ).

```

:
int ires = ts_dbsqlite(iSQLite, "open_db", string connectstring);
:
ires = -1/0 - / ;
iSQLite - ;
connectstring - ;
:
ires = ts_dbsqlite(iSQLite, "open_db", "E:/Work/ЛПН2402-SQLite/q.db");
if (ires != 0) {
    return -1;
}
cout << "connection success\n";

```

### prepare\_v2\_query - запрос к базе данных

unicode (  $m^2$  ).  
ts\_table.

```

:
int ires = ts_dbqlite(iSQLite, "", string query);

```

```

:
ires = -1/0 - / ;
iSQLite - ;
query - SQL;

:

string query = string query = "select * from my_first where (id <= 10000)OR(id >=
90000)";
int j = 0;
ires = ts_dbsqlite(iSQLite, "prepare_v2_query", query);
if (ires != 0) {
    return -1;
}
cout << "query success\n";

```

## column\_count

```

:
int ires = ts_dbsqlite(iSQLite, "column_count");

:
ires = -1/0 - / ;
iSQLite - ;

:

ires = ts_dbsqlite(iSQLite, "column_count");
cout << "num fields = " << ires << "\n";

```

## row\_count

```

:
int ires = ts_dbsqlite(iSQLite, "row_count", count);

:
ires = -1/0 - / ;
iSQLite - ;
count - .

:

int count;
ires = ts_dbsqlite(iSQLite, "row_count", count);
cout << "row count = " << count << "\n";

```

## export\_to\_table

```

, , ts_table.

:

:
int ires = ts_dbsqlite(iSQLite, "export_to_table", int iTable);
:
int ires = ts_dbsqlite(iSQLite, "export_to_table", int iTable, int/string colname1,

```

```
string labppcoltype1, int/string colname2, string labppcoltype2, ...);
```

```

:
ires = -1/0 - / ;
iSQLite - ;
iTable - ts_table, .
int/string colname -

:

int iTable;
object("create", "ts_table", iTable);
// вариант 1
ts_dbsqlite(iSQLite, "export_to_table", iTable);
// вариант 2
ts_dbsqlite(iSQLite, "export_to_table", iTable, "id","string","count","int",5,"double");

```

## finalize\_stmt

```
int ires = ts_dbsqlite(iSQLite, "finalize_stmt");
```

```

:
ires = -1/0 - / ;
iSQLite - ;

:
// освободить память результата запроса в подключении у объекта iSQLite
ts_dbsqlite(iSQLite, "finalize_stmt");

```

## close\_db

```
int ires = ts_dbsqlite(iSQLite, "close_db");
```

```

:
ires = -1/0 - / ;
iSQLite - ;

:
ts_dbsqlite(iPostgreSQL, "close_db");

```

## Пример - получение таблицы с сайта из базы данных PostgreSQL в диалог ARCHICAD

SQLite.

Наименования материалов для заказчика			
№	Идентификатор	Наименование для заказчика	Идентификатор
12317	0xB578B49691C4...	Кабель с алюминиевыми жилами AcBBГнг(A)-LS 5x4	0xB569B49691C
12318	0xB578B49691C4...	Кабель с алюминиевыми жилами AcBBГнг(A)-LS 5x6	0xB569B49691C
12319	0xB578B49691C4...	Труба ПЭ100 SDR17-90x5,4	0xB577B49691C
12320	0xB578B49691C4...	Труба ПЭ100 SDR17-225x13,4	0xB577B49691C
12321	0xB578B49691C4...	Труба ПЭ100 SDR17-110x6,6	0xB577B49691C
12322	0xB578B49691C4...	Труба ПЭ100 SDR17-400x23,7	0xB577B49691C
12323	0xB578B49691C4...	Бетон B35 (M450) W6 F150	0xB569B49691C
12324	0xB578B49691C4...	Бетон B25 (M350) W6 F75	0xB569B49691C
12325	0xB578B49691C4...	Бетон B35 (M450) W4 F100	0xB569B49691C
12326	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1100 с нижним подкл...	0xB569B49691C
12327	0xB578B49691C4...	Утеплитель минераловатный 135кг/м3 200мм	0xB569B49691C
12328	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1100 с боковым подк...	0xB569B49691C
12329	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1400 с боковым подк...	0xB569B49691C
12330	0xB578B49691C4...	Стальной панельный радиатор тип 22x500x1400 с нижним подкл...	0xB569B49691C

```

int iListBox_MD, iButton_Ok_MD;
int iTable_MD;
int iDialog_MD;

int main()
{
    int iSQLite;
    object("create", "ts_dbsqlite", iSQLite);
    int ires;
    ires = ts_dbsqlite(iSQLite, "init");
    if (ires != 0) {
        return -1;
    }
    cout << "init success\n";
    ires = ts_dbsqlite(iSQLite, "open_db", "E:/Work/ЛПП2402-SQLite/q.db");
    if (ires != 0) {
        return -1;
    }
    cout << "connection success\n";

    string query = "select * from my_first where (id <= 10000)OR(id >= 90000)";

    int j = 0;

    ires = ts_dbsqlite(iSQLite, "prepare_v2_query", query);
    if (ires != 0) {
        return -1;
    }
    cout << "query success\n";

    ires = ts_dbsqlite(iSQLite, "column_count");
    cout << "num fields = " << ires << "\n";
}

```

```

int iTable;
object("create", "ts_table", iTable);

ts_dbsqlite(iSQLite, "export_to_table", iTable, 0, "string", 1, "string", 2,
"string");
string sstr;
ts_table(iTable, "print_to_str", sstr);
coutvar << sstr;

ts_dbsqlite(iSQLite, "finalize_stmt");
ts_dbsqlite(iSQLite, "close_db");

// Create Dialog
int x, y, w, h, offset;

object("create", "ts_dialog", iDialog_MD);
ts_dialog(iDialog_MD, "init_dialog", "palette", 0, 0, 350, 400);
ts_dialog(iDialog_MD, "set_as_main_panel");
ts_dialog(iDialog_MD, "SetGrowType", 3);

ts_dialog(iDialog_MD, "SetTitle", "Наименования материалов для заказчика");

object("create", "ts_dialogcontrol", iListBox_MD, "iListBox_MD");

ts_dialogcontrol(iListBox_MD, "init_control", "singleselectlistbox", iDialog_MD, 5,
5, 340, 350, 48, 20);
ts_dialogcontrol(iListBox_MD, "SetAnchorToPanelResize", 0, 0, 1, 1);

ts_table(iTable, "set_columns_width_when_export", 100);

ts_table(iTable, "export_to_dialogcontrol", iListBox_MD, -1, -1);

x = 10;
w = 98;
int delta = 10;
x = x + w + delta;
y = 370;
h = 20;

x = x + w + delta;
object("create", "ts_dialogcontrol", iButton_Ok_MD, "iButton_Ok_MD");
ts_dialogcontrol(iButton_Ok_MD, "init_control", "button", iDialog_MD, x, y, w, h);
ts_dialogcontrol(iButton_Ok_MD, "eventreaction", "Event_ButtonClicked",
"Event_ButtonClicked_MD");
ts_dialogcontrol(iButton_Ok_MD, "settext", "Ok");
ts_dialogcontrol(iButton_Ok_MD, "SetAnchorToPanelResize", 1, 1, 0, 0);

bool bres;
ts_dialog(iDialog_MD, "invoke", bres);

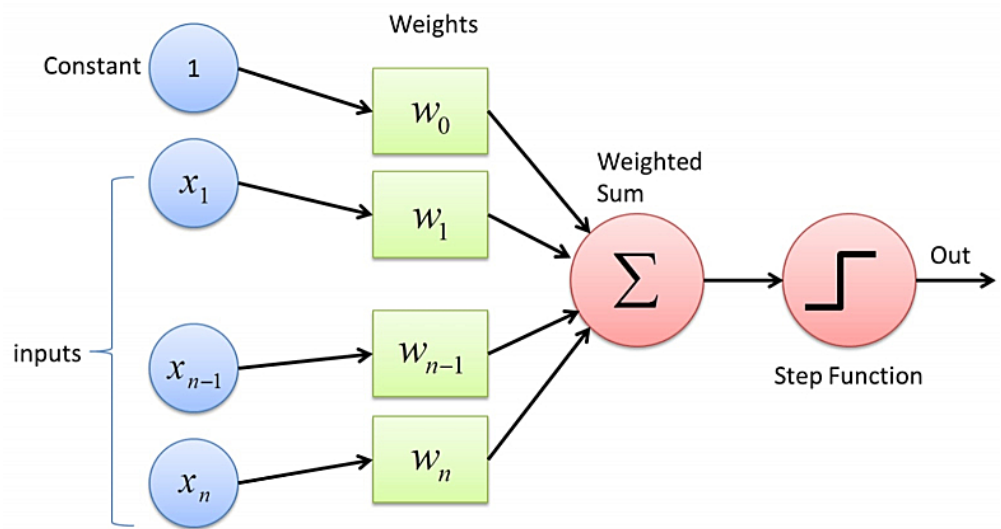
//<-
object("delete", iTable);
object("delete", iSQLite);
}

int Event_ButtonClicked_MD(int iDescr, string sDescr)
{
    if (sDescr == "iButton_Ok_MD") {
        ts_dialog(iDialog_MD, "SendCloseRequest");
    }
}

```

Искусственный интеллект

Нейросеть на основе перцептрона Розенблатта - ts\_aipercrozen



LABPP

init

```
int ires = ts_aipercrozen(int iPercRozen, "init", int in1,int in2,int inn,int out);
```

ires = -1    0,  
iPercRozen -  
int in1 .... out -

500:2

0	500	200
1	200	50
2	50	5
3	5	2

```

int iPercRozen;
object("create", "ts_percrozen", iPercRozen);
int ires = ts_aipercrozen(int iPercRozen, "init", 500, 200, 50, 5, 2);

```

### task\_add\_to\_list

```

int ires = ts_aipercrozen(int iPercRozen, "task_add_to_list", string taskname, int
iTableIN, int iTableOUT);

```

```

ires -
taskname -
iTableIN - double;
iTableOUT -

```

200 2

```

int iTable11;
object("create", "ts_table", iTable11);
ts_table(iTable11, "add_column", 0, "double", "Вход");
int i, icount;
icount = 200;
for (i = 0; i < icount; i++)
{
    ts_table(iTable11, "add_row", 0, i+1);
}

int iTable12;
object("create", "ts_table", iTable12);
ts_table(iTable12, "add_column", 0, "double", "Выход для обучения");
ts_table(iTable12, "add_column", 1, "double", "Выход реальный");
ts_table(iTable12, "add_row", 0, 0.77, 1, 0);
ts_table(iTable12, "add_row", 0, 0.88, 1, 0);
ts_aipercrozen(iPercRozen, "task_add_to_list", "Задача 1", iTable11, iTable12);
int iTable1;
object("create", "ts_table", iTable1);
ts_table(iTable1, "add_column", 0, "double", "Вход");
icount = 200;
for (i = 0; i < icount; i++)
{
    ts_table(iTable1, "add_row", 0, 100 - i);
}

int iTable22;
object("create", "ts_table", iTable22);
ts_table(iTable22, "add_column", 0, "double", "Выход для обучения");
ts_table(iTable22, "add_column", 1, "double", "Выход реальный");
ts_table(iTable22, "add_row", 0, 0.55, 1, 0);
ts_table(iTable22, "add_row", 0, 0.77, 1, 0);

ts_aipercrozen(iPercRozen, "task_add_to_list", "Задача 2", iTable1, iTable22);

```



```

ts_aipercrozen(iPercRozen, "learn_task_list", 200000);

ts_aipercrozen(iPercRozen, "run_task_list");

ts_aipercrozen(iPercRozen, "get_task_result_to_table", 0, iTable12,false,1);
string str;
ts_table(iTable12, "print_to_str", str);
coutvar << str;
ts_aipercrozen(iPercRozen, "get_task_result_to_table", 1, iTable22,false,1);
ts_table(iTable22, "print_to_str", str);
coutvar << str;

```

## learn\_task\_list

```

:
:
:
int ires = ts_aipercrozen(int iPercRozen, "learn_task_list", int learncount);

:
:
ires - 0;
iPercRozen - ;
learncount - .

```

## run\_task\_list

```

:
:
int ires = ts_aipercrozen(iPercRozen, "run_task_list");

:
:
ires - 0

```

## get\_task\_result\_to\_table

```

:
:
int ires = ts_aipercrozen(int iPercRozen, "get_task_result_to_table", int iTaskIndex, int
iTableOUT, bool bDoClearTable, int iColumnIndex);

:
:
ires - 0;
iPercRozen - ;
iTaskIndex - ;
iTableOUT - ;
bDoClearTable - , double iColumnIndex.

iColumnIndex - ,

```

## task\_list\_clear

```

:
:
ts_aipercrozen(int iPercRozen, "task_list_clear");

:
:
iPercRozen - .

```

**store\_matrices и restore\_matrices**

```

:
ts_aipercrozen(iPercRozen, "store_matrices");
ts_aipercrozen(iPercRozen, "restore_matrices");

```

**clear**

```

:
init

:
ts_aipercrozen(iPercRozen, "clear");

```

**ts\_aipercrozen\_matrices****init**

```

:
( . . . ).
:
int ires = ts_aipercrozen_matrices(int iPercRozenMatrices, "init", int in1,int in2,int
inn,int out);

```

```

:
ires = -1 0,
iPercRozenMatrices -
int in1 .... out -

```

**get\_from\_percrozen**

```

:
int ires = ts_aipercrozen_matrices(int iPercRozenMatrices, "get_from_percrozen", int
iPercRozen);

```

```

:
ires = -1 0, ;
iPercRozenMatrices - ;
iPercRozen -

```

```

:
int iPercRozenMatrices;
object("create", "ts_aipercrozen_matrices", iPercRozenMatrices);
ts_aipercrozen_matrices(iPercRozenMatrices, "get_from_percrozen", iPercRozen);

```

**set\_to\_percrozen**

```

:
int ires = ts_aipercrozen_matrices(int iPercRozenMatrices, "set_to_percrozen", int
iPercRozen);

```

```

:
ires = -1 0, ;

```

```
iPercRozenMatrices - ;
iPercRozen - .
```

```

:
Получить матрицы из базы данных и записать в перцептрон.

```

```

int iPercRozenMatrices;
object("create", "ts_aipercrozen_matrices", iPercRozenMatrices);

ts_aipercrozen_matrices(iPercRozenMatrices, "load_from_PostgreSQL", iPercRozen,
iDBPostgreSQL, sTableName, percGUID);
ts_aipercrozen_matrices(iPercRozenMatrices, "set_to_percrozen", iPercRozen);

ts_aipercrozen_matrices(iPercRozenMatrices, "set_to_percrozen", iPercRozen);

```

```
export_to_table
```

```
ts_table.
```

```

:
int ires = ts_aipercrozen_matrices(int iPercRozenMatrices, "export_to_table", int
iTable);

```

```

:
ires = -1 0, ;
iPercRozenMatrices - ;
iTable - .

```

			W1	W...	Wn
0	100	30	1.2	...	...
0	100	30	...	...	...
...	...	...	...	...	...
1	30	10	0.00012 3	...	...

```

:
int iTableMatrices;
object("create", "ts_table", iTableMatrices);
ts_aipercrozen_matrices(iPercRozenMatrices, "export_to_table", iTableMatrices);

```

```
import_from_table
```

```
ts_table.
```

```
/
```

```
,
```

```
, JSON
```

```

:
int ires = ts_aipercrozen_matrices(int iPercRozenMatrices, "import_from_table", int
iTable);

```

```

:
ires = -1 0, ;

```

```
iPercRozenMatrices - ;
iTable - .
```

```
ts_aipercrozen_matrices, ts_aipercrozen
( , ).
```

			W1	W...	Wn
0	100	30	1.2	...	...
0	100	30	...	...	...
...	...	...	...	...	...
1	30	10	0.00012 3	...	...

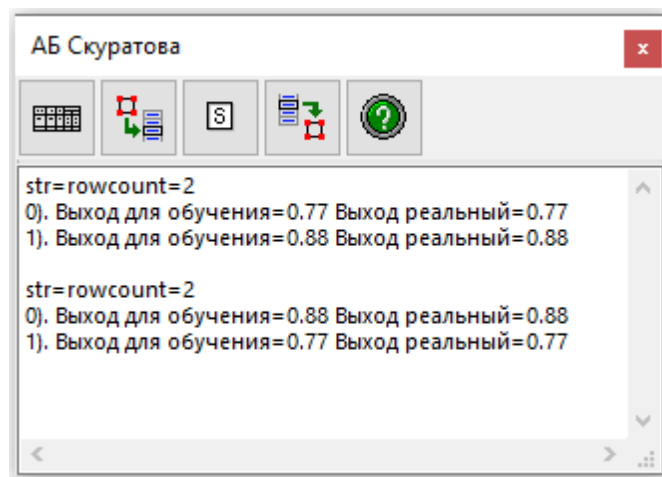
```
:
ts_aipercrozen_matrices(iPercRozenMatrices, "import_from_table", iTableMatrices);
```

```
clear
```

```
init
```

```
:
ts_aipercrozen_matrices(iPercRozen, "clear");
```

## Пример



```
int main()
{
    int iPercRozen;
    object("create", "ts_aipercrozen", iPercRozen);

    ts_aipercrozen(iPercRozen, "init", 200,100,10,5,2);

    int iTable11;
    object("create", "ts_table", iTable11);
```

```

ts_table(iTable11, "add_column", 0, "double", "Вход");
int i, icount;
icount = 200;
for (i = 0; i < icount; i++)
{
    ts_table(iTable11, "add_row", 0, i+1);
}

int iTable12;
object("create", "ts_table", iTable12);
ts_table(iTable12, "add_column", 0, "double", "Выход для обучения");
ts_table(iTable12, "add_column", 1, "double", "Выход реальный");
ts_table(iTable12, "add_row", 0, 0.77, 1, 0);
ts_table(iTable12, "add_row", 0, 0.88, 1, 0);

ts_aipercrozen(iPercRozen, "task_add_to_list", "Задача 1", iTable11, iTable12);

int iTable21;
object("create", "ts_table", iTable21);
ts_table(iTable21, "add_column", 0, "double", "Вход");
icount = 200;
for (i = 0; i < icount; i++)
{
    ts_table(iTable21, "add_row", 0, 200 - i);
}

int iTable22;
object("create", "ts_table", iTable22);
ts_table(iTable22, "add_column", 0, "double", "Выход для обучения");
ts_table(iTable22, "add_column", 1, "double", "Выход реальный");
ts_table(iTable22, "add_row", 0, 0.88, 1, 0);
ts_table(iTable22, "add_row", 0, 0.77, 1, 0);

ts_aipercrozen(iPercRozen, "task_add_to_list", "Задача 2", iTable21, iTable22);

ts_aipercrozen(iPercRozen, "learn_task_list", 200000);

ts_aipercrozen(iPercRozen, "run_task_list");

ts_aipercrozen(iPercRozen, "get_task_result_to_table", 0, iTable12, false, 1);
string str;
ts_table(iTable12, "print_to_str", str);
coutvar << str;
ts_aipercrozen(iPercRozen, "get_task_result_to_table", 1, iTable22, false, 1);
ts_table(iTable22, "print_to_str", str);
coutvar << str;

object("delete", iTable11);
object("delete", iTable12);
object("delete", iTable21);
object("delete", iTable22);
object("delete", iPercRozen);
}

```

## BIM Master

---

BIM Master -

BIM Master

LABPP\_Automat.

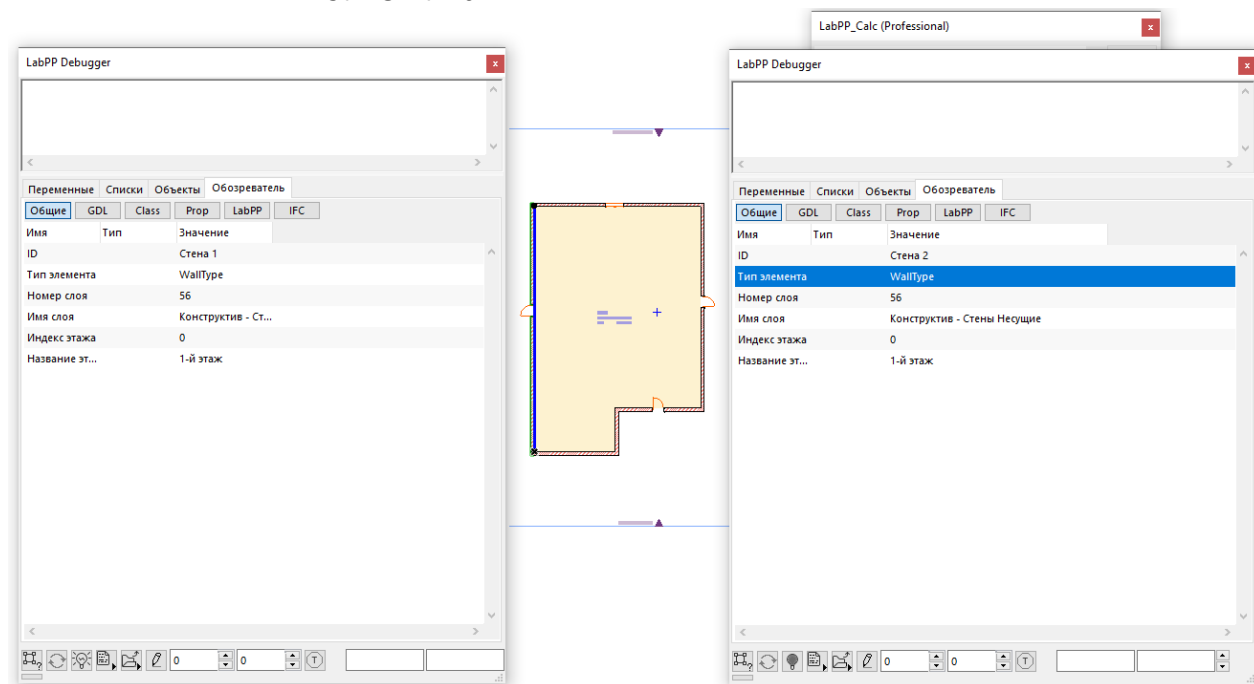
breakpoint()

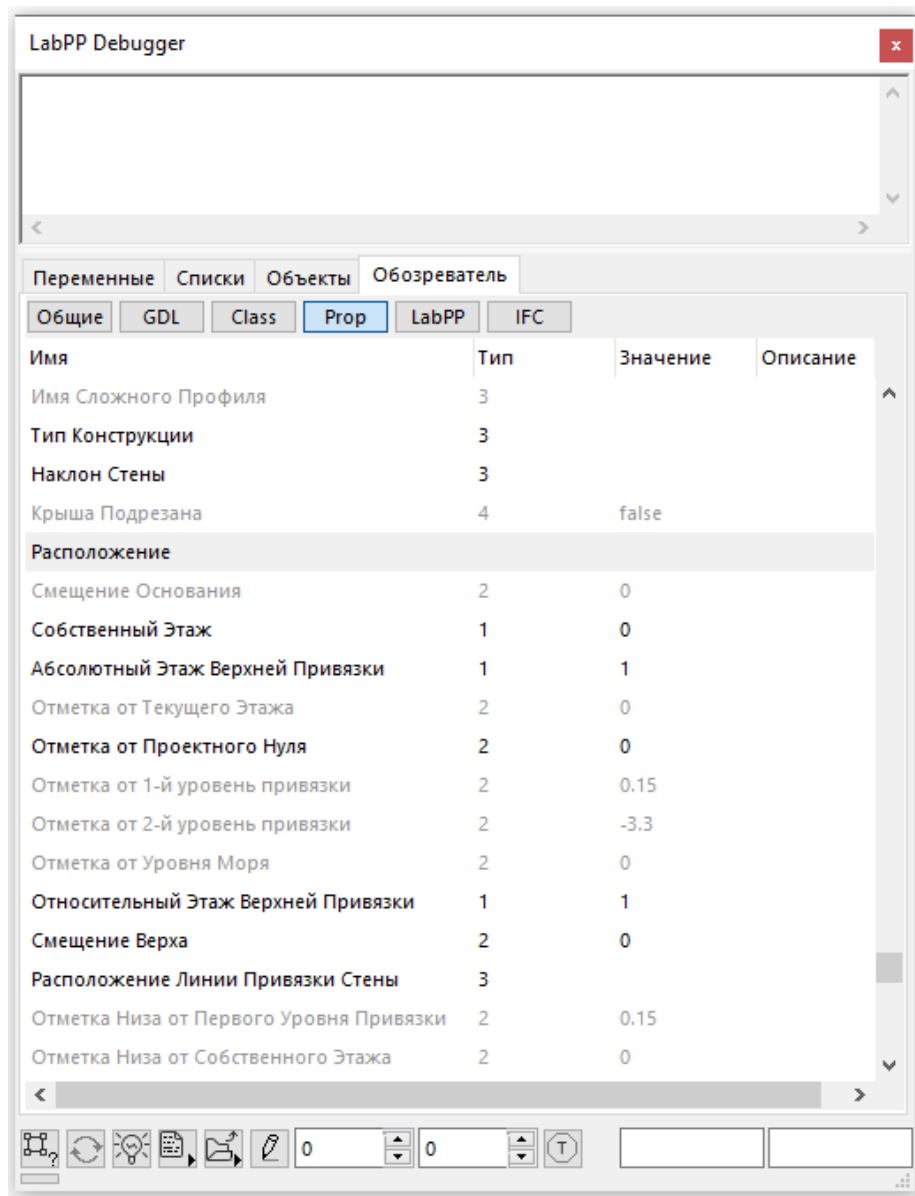
, GDL, IFC . . . ( ),

Ctrl+Shift+F9



Ctrl+Shift+F9





## breakpoint

```

        BIM Master'
    :
    breakpoint(int icondition);

    breakpoint(int icondition, bool/int/double/string condition2);

    :
    icondition -          BIM Master
  
```

## programcodegen

### get\_bimmaster\_cur\_selected\_parameter\_info

```

        BIM Master
    :
    programcodegen("get_bimmaster_cur_selected_parameter_info", string genesis, string
  
```

```
v_type, string varname, string svalue, string sdescription, string sgroup);
```

```

:
genesis - ( - , GDL . .)
v_type -

```

"string"	
"real"	
"integer"	
"boolean"	

```
varname -
svalue -
sdescription -
sgroup -
```

## get\_bimmaster\_cur\_element\_guidastext

```

        BIM Master                                GUID.
        :
programcodegen("get_bimmaster_cur_element_guidastext", string sGUIDAsText);
        :
sGUIDAsText - GUID,                                ,
BIM Master.

```

## show\_bimmaster\_as\_codegen

```

BIM Master
:
programcodegen("show_bimmaster_as_codegen");

```

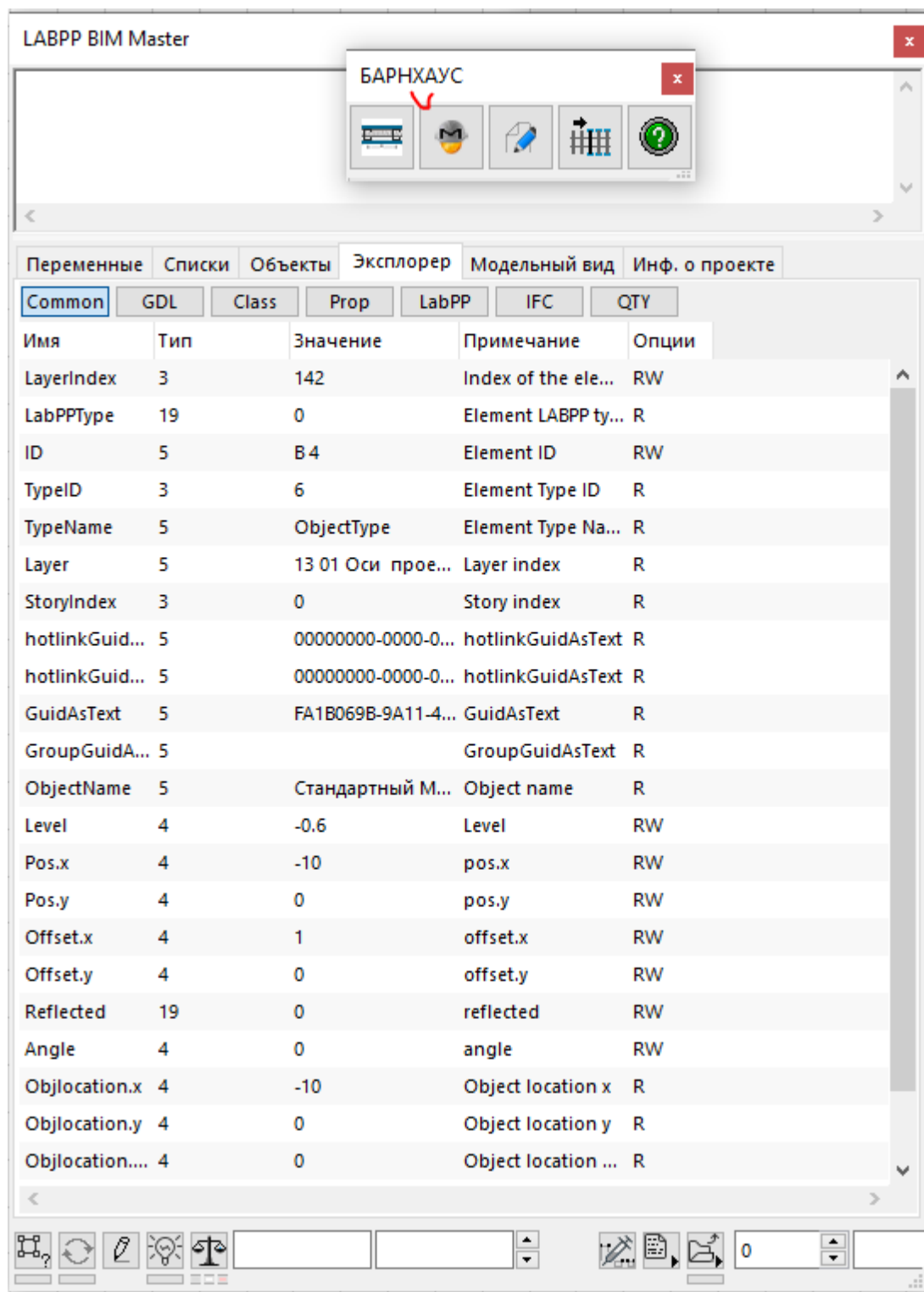
## show\_bimmaster

```

        BIM Master
    :
    programcodegen("show_bimmaster");

```





## Уроки

### Вычислить середину высоты элемента Морф относительно 0 проекта и записать в поле LabPP\_Calc

```
// Урок 1
// Для начала - в проекте ARCHICAD создать элемент типа Морф.
// задать ему классификатор "Лестница" по классификации "Классификация ARCHICAD"
```

```

// выбрать его
// открыть LabPP_Calc
// открыть в нем правой нижней маленькой кнопкой окно сообщений
// второй справа нижней маленькой кнопкой запустить этот программный модуль средствами
LabPP_Calc
// Результат - в расчетное поле "a" калькулятора запишется высота середины элемента Морф
а в комментарий - его ID
// Задание 1
// Сделать так, чтобы значение записывалось во второе поле
// Задание 2
// Создать в классификаторе "Классификация ARCHICAD" класс "Отделка стен".
// Присвоить этот класс элементу Морф
// Сделать так, чтобы в поле LabPP_Calc записывалась середина высоты этого элемента.
// Подсказка: при выборке элементов из проекта - слово "Лестница" надо на что-то заменить
// Задание 3
// Результат перед записью в поле увеличить на 10% (умножить на 1.1)

int main()
{
    // загрузить из проекта в список 1 все элементы, у которых присвоено любое
    значение классификатора ARCHICAD
    ac_request_special("add_elements_list_from_selection", 1, "MorphType", 2,
        "", "Cls", "Классификация ARCHICAD", "=", "Лестница", "");

    // запросить количество собранных элементов
    ac_request("get_loaded_elements_list_count", 1);
    int icount = ac_getnumvalue(); // получить количество в переменную
    coutvar << icount; // вывести в окно сообщений

    if (icount == 0)
    {
        cout << "В списке нет элементов";
        return -1;
    }
    // объявляем необходимые переменные для работы
    double dLevel, dHeight, elemlevel, elemprojectlevel;
    string sElemTypeName, sID;
    int ielemstoryindex;
    int i;
    int ires;
    // выполнить цикл icount раз
    // в нашем случае нам нужен один элемент, поэтому отработаем цикл только 1 раз на
    элементе с индексом 0
    icount = 1;

    for (i = 0; i < icount; i++)
    {
        ires = ac_request("set_current_element_from_list", 1, i); // сделать
        текущим i-вый элемент из списка 1

        ires = ac_request("get_element_value", "Level"); // запрашиваем значение
        свойства Level (для Морф - высота над уровнем его этажа)
        dLevel = ac_getnumvalue(); // получаем его в переменную
        coutvar << dLevel; // выводим в окно сообщений
        ires = ac_request("get_quantity_value", "max_height"); // запрашиваем
        максимальный размер Морф по вертикали
        dHeight = ac_getnumvalue(); // получаем его в переменную
        coutvar << dHeight; // выводим в окно сообщений
        elemlevel = dLevel + dHeight / 2; // рассчитываем уровень центра Морф по
        высоте относительно этажа
        ac_request("get_element_value", "StoryIndex"); // запрашиваем индекс этажа
        ielemstoryindex = ac_getnumvalue(); // получаем индекс этажа в переменную
        // получаем высоту центра Морф относительно 0 проекта в переменную
        elemprojectlevel
    }
}

```

```

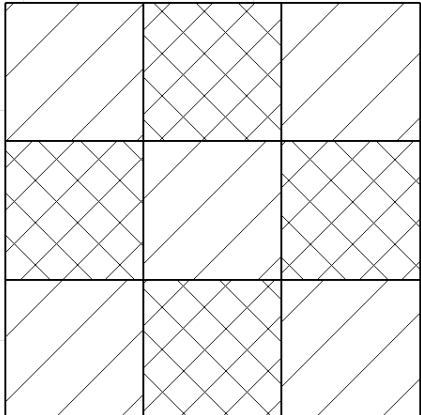
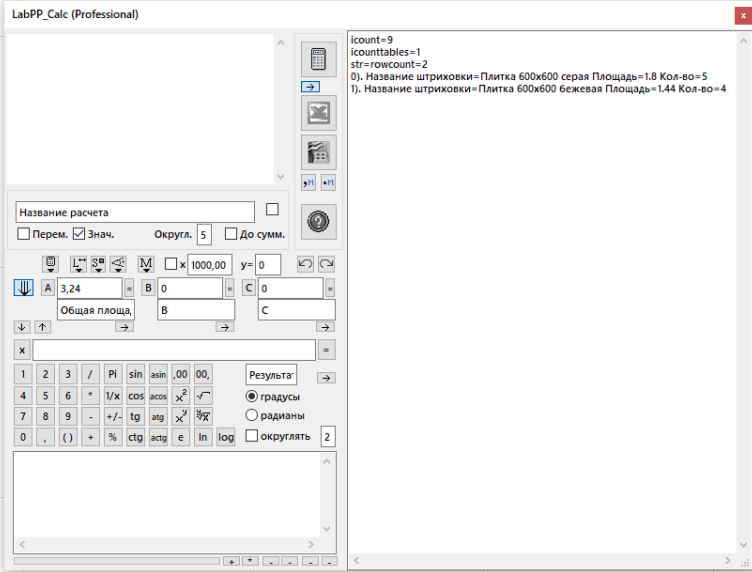
        ac_request("get_levelfromprojectnull_by_floorindexandlevel",
ielemstoryindex, elemlevel, elemprojectlevel);
        coutvar << elemprojectlevel; // выводим в окно сообщений
        ires = ac_request("get_element_value", "ID"); // запросить ID элемента
        sID = ac_getstrvalue(); // получить его в переменную
        coutvar << sID; // вывести ID в окно сообщений
    }
    string svalue = ecvt(elemprojectlevel);
    string scomment = "Высота элемента с ID=" + sID + "от 0 проекта";
    ac_request("interface", "calc_field", "set", "a", 1, svalue, scomment);
    // "a" - здесь имя ячейки у LabPP_Calc куда записать значение. Всего ячеек - a,b,c
и main
}

```

## Заполнить таблицу с расчетом плитки по элементам Штриховка и записать общую площадь в поле LabPP\_Calc

Раскладка плитки

Название штриховки	Площадь	Кол-во
Плитка 600x600 серая	1,80	5
Плитка 600x600 бежевая	1,44	4

```

// Урок 2
// Для начала - в проекте АРХИКАД в любом окне разложить штриховки 600x600. Задать им
типы штриховок произвольно.
// Поставить там же GDL элемент LABPP_Table.gsm.
// Выделить одновременно все штриховки и элемент таблицы.
// открыть LabPP_Calc
// открыть в нем правой нижней маленькой кнопкой окно сообщений
// второй справа нижней маленькой кнопкой запустить этот программный модуль средствами
LabPP_Calc
// Результат - в таблицу запишется список штриховок с названием типа и количеством
// Задание 1
// Переименовать колонку "Название штриховки" в "Тип плитки"
// Задание 2
// Поменять местами колонки.
// Задание 3
// Упорядочить строки по колонке с названием штриховки (см. команду "Sort")

```

```

int main()
{
    // загрузить из проекта в список 1 все выбранные элементы штриховок
    ac_request_special("add_elements_list_from_selection", 1, "HatchType", 2);
    // загрузить из проекта в список 2 все выбранные элементы таблиц
    ac_request_special("add_elements_list_from_selection", 2, "ObjectType", 2,

```

```

        "", "EP", "ObjectName", "=", "LABPP_Table", "");

// запросить количество собранных элементов штриховок
ac_request("get_loaded_elements_list_count", 1);
int icount = ac_getnumvalue(); // получить количество в переменную
coutvar << icount; // вывести в окно сообщений

if (icount == 0)
{
    cout << "В списке нет элементов штриховок";
    return -1;
}

// запросить количество собранных элементов штриховок
ac_request("get_loaded_elements_list_count", 2);
int icountTables = ac_getnumvalue(); // получить количество в переменную
coutvar << icountTables; // вывести в окно сообщений

if (icountTables == 0)
{
    cout << "В списке нет элемента таблицы";
    return -1;
}

int iTable;
object("create", "ts_table", iTable); // создать таблицу

// Создаем колонки таблицы
ts_table(iTable, "add_column", 0, "string", "Название штриховки");
ts_table(iTable, "add_column", 1, "double", "Площадь");
ts_table(iTable, "add_column", 2, "int", "Кол-во");
ts_table(iTable, "set_first_key", 0); // по этой колонке будет отслеживаться
одинаковость строк при суммировании показателей

// объявляем необходимые переменные для работы
double dSquare, dAllSquare=0;
string sFillName;
int i;
int ires;
// выполнить цикл icount раз чтобы заполнить таблицу.
// обратить внимание, что одинаковые строки в таблице суммируются автоматически
("add_row_sum")
for (i = 0; i < icount; i++)
{
    ires = ac_request("set_current_element_from_list", 1, i); // сделать
текущим i-ый элемент из списка 1

    ires = ac_request("get_element_value", "fillName"); // запрашиваем название
типа штриховки у элемента штриховки
    sFillName = ac_getstrvalue(); // получаем его в переменную
    ires = ac_request("get_quantity_value", "Surface"); // запрашиваем площадь
элемента штриховки
    dSquare = ac_getnumvalue(); // получаем его в переменную
    dAllSquare += dSquare; // суммируем общую площадь
    // Добавляем строку в таблицу, если уже есть - то суммируем
    ts_table(iTable, "add_row_sum", 0, sFillName, 1, dSquare, 2, 1);
}

// вывести содержимое таблицы в окно сообщений
string str;
ts_table(iTable, "print_to_str", str);
coutvar << str;

```

```

// вывести содержимое таблицы в элемент LABPP_Table.gsm
ac_request("set_current_element_from_list", 2, 0); // сделать текущим 0-ой элемент
из списка 2
bool withheader = true; // выводить с заголовками колонок
int startcolindex = 1; // стартовая колонка в LABPP_Table будет 1
int startObjectrow = 1; // стартовая строка в LABPP_Table будет 1 (т.е. с левого
верхнего угла)
int starttablerow = 0; // из таблицы iTable начинать с первой строки (индекс
первой строки 0)
int maxcolumns; // сколько колонок задействовано
int maxrows; // сколько строк получилось задействовано
// получить число колонок и строк
ts_table(iTable, "get_columns_count", maxcolumns);
ts_table(iTable, "get_rows_count", maxrows);
ts_table(iTable, "export_to_LABPP_Table", withheader, startcolindex,
startObjectrow, starttablerow, maxrows);
// Задать в объекте количество использованных строк
ac_request("set_object_property_value", "TS_RowsUsedCount", maxrows + 1);
// Задать в объекте количество использованных колонок
ac_request("set_object_property_value", "TS_ColumnsUsedCount", maxcolumns);
// Дать команду GDL объекту таблицы LABPP_Table выполнить внутренний перерасчет
ac_request("set_object_property_value", "TS_DoRecalcAll", 1);
// Удалить из памяти таблицу
object("delete", iTable);

// Записываем общую площадь в поле "a" калькулятора с комментарием
string svalue = ecvt(dAllSquare); // переводим общую площадь в текст
string scomment = "Общая площадь шриховок";
ac_request("interface", "calc_field", "set", "a", 1, svalue, scomment);
// "a" - здесь имя ячейки у LabPP_Calc куда записать значение. Всего ячеек - a,b,c
и main
}

```

## FAQ (часто задаваемые вопросы)

---

## Руководство по LABPP\_Automat для ARCHICAD

[LabPP Debugger.](#)