



LABPP_Automat ARCHICAD.docx

Документ



Расскажи, что ты узнал из этого документа. Что такое LABPP_Automate и в чем его ключевое преимущество.



Отличный вопрос — это как раз тот случай, когда документ **выглядит как «мануал по автоматизации»**, а на деле скрывает куда более сильную идею.

Я отвечу в три слоя:

1. **что это вообще такое,**
2. **что я реально узнал из документа,**
3. **в чём ключевое преимущество LABPP_Automat — и почему оно недооценено.**

1. Что такое LABPP_Automat (по сути, а не по названию)

LABPP_Automat — это встроенная в Archicad среда автоматизации и программирования, которая:

- запускается **внутри Archicad**,
- использует **собственный С-подобный язык** (интерпретатор),
- имеет **прямой доступ к элементам Archicad** (через `ac_request`, `ac_element`, свойства, слои, классификации и т.д.),
- умеет работать с **интерфейсом**, таблицами, файлами, БД, JSON,
- и даже содержит **простейший модуль ИИ (перцептрон Розенблатта)**.

Это не Python, не C++ API, не GDL.

Это **внутренняя automation-платформа**, ближе всего по духу к:

Visual Basic for Excel, но для Archicad — и сильно мощнее.

2. Что именно раскрывает этот документ (самое важное)

Документ — не обзорный, а **технически глубокий**. Из него чётко видно, что LABPP_Automat умеет:

♦ Управлять Archicad напрямую

- читать и менять **свойства элементов**
- работать со **слоями, этажами, зонами, классификациями**
- выбирать элементы, работать со списками GUID
- создавать элементы в проекте

📌 Это не экспорт → импорт. Это **прямое управление моделью**

♦ Работать с таблицами как с объектами

- собственный объект **ts_table**
- фильтрация, поиск, агрегация, формулы
- экспорт:
 - в CSV,
 - обратно в параметры объектов,
 - в интерфейсные элементы

📌 Это намного мощнее стандартных спецификаций Archicad

♦ Иметь собственный UI внутри Archicad

- диалоги (модальные / немодальные)
- панели, кнопки, меню, radio, listbox
- обработка событий (**eventreaction**)
- управление жизненным циклом окон

📌 Это **настоящие кастомные инструменты**, а не скрипты

♦ Общаться с внешним миром

- **JSON** (чтение / запись / валидация)
- **Excel / Word** через OLE (включая управление шрифтами, полями)
- **Базы данных:**
 - SQLite
 - PostgreSQL

- MySQL
 - запуск внешних программ (`run_cpp`, `run_from_file`)
-

♦ И да — там есть ИИ (но не LLM)

В документе описан модуль:

- `ts_aipercrozen`
- классический **перцептрон Розенблатта**
- обучение, сохранение матриц, применение

📌 Это не ChatGPT, но **важно другое**:
архитектура уже допускает интеллект внутри Archicad

3. Ключевое преимущество LABPP_Automat (самое главное)

🔥 Главное преимущество:

LABPP_Automat живёт *внутри* Archicad, а не снаружи

И из этого вытекают критически важные последствия.

🧠 1. Нет разрыва контекста

- Не нужен:
 - Python Bridge
 - HTTP
 - JSON-сервер
- Нет проблемы:
 - «Archicad не открыт»
 - «проект не активен»
 - «API не отвечает»

LABPP видит текущий элемент, текущее выделение, текущее состояние проекта.

⚡ 2. Реальное «реальное время»

В отличие от Python API:

- реакции на события
- интерактивные диалоги
- пошаговая логика

📌 Это **настоящая интеграция**, а не «скрипт-помощник».

🧩 3. Это идеальный “исполнитель” для AI-агента

Вот здесь начинается самое интересное.

Связка выглядит так:

LLM (внешний мозг)

- формирует решение / правила / JSON
- **LABPP_Automat (внутренние руки)**
- применяет к модели

Почему это лучше, чем Python API:

- LABPP:
 - сам валидирует,
 - сам знает структуру Archicad,
 - сам управляет интерфейсом пользователя.

LLM не нужно «знать Archicad» —
он работает **через LABPP как через DSL**.

🧠 4. LABPP — это DSL для BIM-логики

Документ показывает, что LABPP:

- хранит состояние,
- управляет процессами,
- умеет сложные проверки,
- работает с БД и таблицами.

📌 Это **язык бизнес-логики BIM**, а не просто API.

DSL (Domain-Specific Language) — это язык, созданный для одной конкретной области задач, а не «для всего подряд».

Коротко на примерах:

- **SQL** — DSL для работы с базами данных

- **HTML** — DSL для описания структуры веб-страниц
- **GDL в Archicad** — DSL для параметрических объектов
- **LABPP** — DSL для автоматизации и логики BIM внутри Archicad

Чем DSL отличается от обычного языка (Python, C++)

- он уже знает предметную область
- в нём меньше кода → больше смысла
- команды говорят «что сделать», а не «как именно это запрограммировать»



Поэтому ключевая мысль:

DSL — это язык, который говорит на языке твоей задачи.

Именно поэтому LABPP так хорошо ложится на связку с ИИ:
LLM думает → DSL исполняет.

Итог одним абзацем

LABPP_Automat — это встроенный в Archicad движок автоматизации, который:

- глубже Python API,
- интерактивнее C++ аддонов,
- гибче GDL,
- и идеально подходит как **исполнительный слой для AI-агентов**.

Если Python — это «удалённый пульт»,
то **LABPP — это нервная система внутри Archicad.**
