Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОННИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Разработка плагина «Серверная часть плагина, получающего расчётные параметры элементов СВЧ ИС в системе автоматизированного проектирования Advanced Design System»

Пояснительная записка по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 581

\_\_\_\_\_\_\_\_\_ В.Д.Боровкова

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Проверил:

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**Оглавление**

[Введение 3](#_Toc185845898)

[1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_Toc185845899)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 5](#_Toc185845900)

[3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 6](#_Toc185845901)

[4 НАЗНАЧЕНИЕ ПЛАГИНА 7](#_Toc185845902)

[5 ОБЗОР АНАЛОГОВ 8](#_Toc185845903)

[6 ОПИСАНИЕ РЕАЛИЗАЦИИ 9](#_Toc185845904)

[7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 20](#_Toc185845905)

[8 ТЕСТИРОВАНИЕ ПЛАГИНА 22](#_Toc185845906)

[8.1 Функциональное тестирование 22](#_Toc185845907)

[8.2 Модульное тестирование 26](#_Toc185845908)

[8.3 Нагрузочное тестирование 30](#_Toc185845909)

[ЗАКЛЮЧЕНИЕ 32](#_Toc185845910)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc185845911)

# Введение

При проектировании СВЧ-устройств учет особенностей целевого технологического процесса часто вызывает сложности. Замена идеальных элементов на реальные модели изменяет характеристики устройства, что может потребовать доработки схемы, занимая много времени. Решением может быть автоматизированный синтез с использованием PDK (Process Design Kit), который содержит данные для проектирования под выбранную технологию. Однако доступность PDK ограничена определенными САПР, что затрудняет их применение в сторонних модулях. Создание «дубликатов» библиотек, например, табличных моделей, помогает упростить данную задачу.

Решением данной проблемы может стать плагин, получающий расчётные параметры элементов СВЧ ИС в САПР Keysight Advanced Design System (ADS), с помощью чего можно сгенерировать дубликат библиотеки компонентов (PDK Twin). Дубликат PDK используется в программе синтеза схемных решений СВЧ-устройств, что позволяет сократить время производства изделий в разы.

В данной работе рассматривается разработка серверной части плагина, получающего расчётные параметры элементов СВЧ ИС в САПР ADS.

# 1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

Целью работы является разработка серверной части плагина, получающего расчётные параметры элементов СВЧ ИС в САПР ADS. Система должна быть выполнена в качестве сторонней программы, осуществляющей непосредственное взаимодействие с программой ADS через API.

АС должна представлять собой консольное приложение, которое будет запускаться вместе с программой PDK Bridge, и через API ADS обращаться к заданному пользователем workspace. В обязанности серверной части плагина будет входить:

− получение пакетов с командами от клиента в виде json-формата;

− обработка полученных команд. В обработку будут входить: обработка пакета с командой, определение команды и её параметров, взаимодействие с ADS через Python API, получение результата из ADS в виде json-файлов;

− отправка результата выполнения команды обратно клиенту в виде файла в json-формате.

В рамках проекта были поставлены следующие задачи:

1. Выбор темы проектирования (09.09.2024-16.09.2024)
2. Составить техническое задание (16.09.2024 – 30.09.2024);
3. Составить проект системы (30.09.2024 – 27.10.2024);
4. Реализовать плагин (27.10.2024 – 08.12.2024);
5. Доработать плагин (08.12.2024– 31.12.2024);
6. Составить пояснительную записку (08.12.2024– 31.12.2024).

В результате анализа было отмечено следующее:

1. В связи с тем, что API ADS на данный момент находится в разработке, оно может не предоставлять некоторую функциональность, которая необходима при разработке;
2. Разработка может занять больше запланированного времени из-за отсутствия опыта работы на данном стеке.

# 2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

PDK Twin ‒ специальный формат данных, разработанный компанией «50ohm Technologies», представляющий из себя набор табличных моделей с параметрами рассеивания и шумов.

Система должна быть выполнена в качестве отдельного плагина САПР ADS, который запускается отдельно.

Входные параметры для работы с серверной частью плагина:

* Путь до Workspace, в котором подключен только один PDK;
* Запрос с клиента в виде json-файла.

# 3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ

В ходе разработки были использованы следующие инструменты:

* Python;
* Keyshight ADS API;
* FastAPI;
* Pydantic.

Приложение написано на Python – высокоуровневом языке программирования общего назначения с динамической типизацией и автоматическим управлением памятью. Его ключевыми особенностями являются: простота синтаксиса, интерпертируемость, поддержка различных парадигм и кроссплатформенность.

Keyshight ADS API **–** официальный API для системы автоматизированного проектирования ADS, предоставляющий необходимые методы для взаимодействия с САПР. Написан на языке программирования Python [1].

FastAPI – фреймворк для создания API, написанный на Python. Его преимуществами являются: высокая производительность, поддержка аннотаций типов для автоматической проверки входных данных, модульность и масштабируемость [2].

**Pydantic** — это Python-библиотека для работы с данными, которая предоставляет мощный инструмент для валидации, сериализации и десериализации данных, основываясь на аннотациях типов Python. Она используется для проверки входных данных и создания строготипизированных моделей в Python-программах [3].

**4 НАЗНАЧЕНИЕ ПЛАГИНА**

Назначение разрабатываемого плагина обусловлено быстрым моделированием и расчетом параметров элементов СВЧ ИС в системе ADS. Благодаря данному расширению, инженеры и разработчики СВЧ ИС могут получить информацию из САПР о загруженном PDK, элементах PDK, схематиках проекта САПР, информации об УГО, а также результатах моделирования.

**5 ОБЗОР АНАЛОГОВ**

Аналогом разрабатываемого плагина является PDK Bridge для AWR – приложение, разработанное компанией «50ohm Technologies», позволяющее получить табличные модели с параметрами рассеивания, а также шумовые параметры СВЧ компонентов PDK путем взаимодействия с API САПР «AWR Design Environment». Полученные данные сохраняются в дубликат PDK (PDK Twin). Дубликат PDK используется в программе синтеза схемных решений СВЧ-устройств, что позволяет сократить время производства изделий в разы. В данный момент приложение является недоступным для общего пользования. Интерфейс программы представлен на рисунке 5.1.

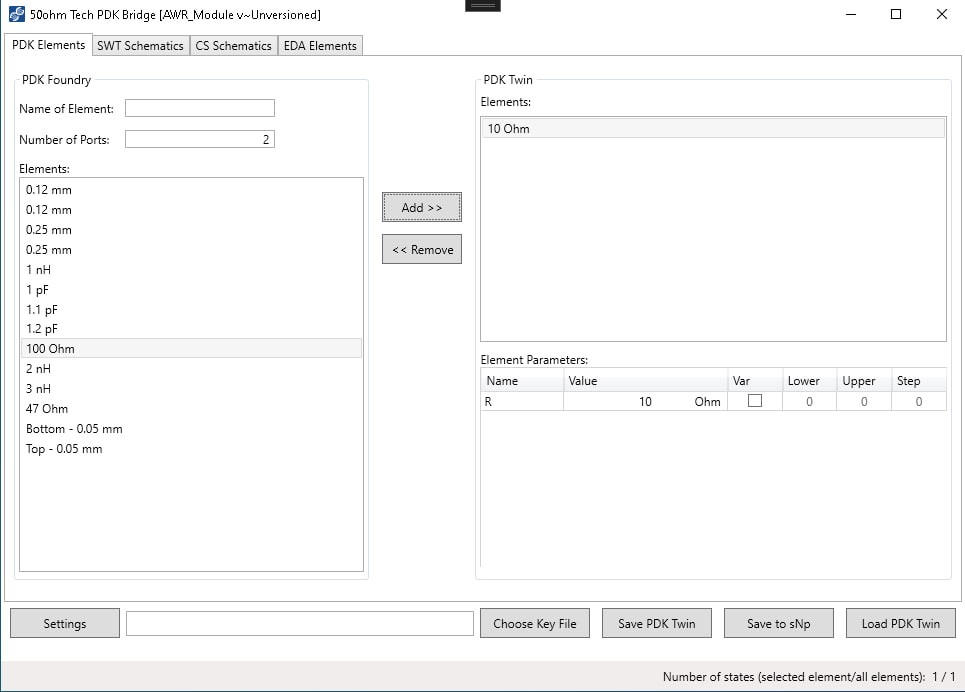


Рисунок 5.1 – PDK Bridge для AWR

**6 ОПИСАНИЕ РЕАЛИЗАЦИИ**

Диаграмма пакетов разрабатываемого плагина представлена на рисунке 6.1.

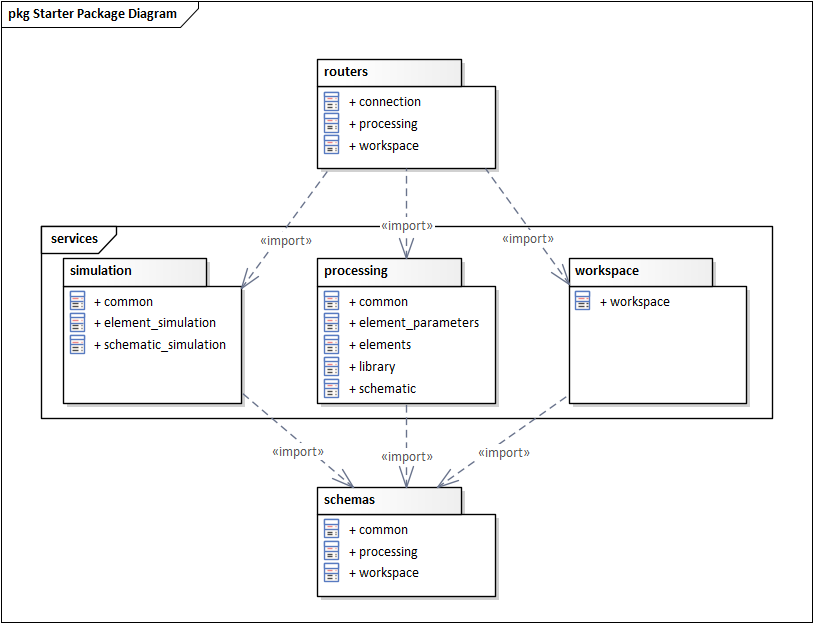


Рисунок 6.1 – Диаграмма пакетов серверной части плагина,получающего расчётные параметры элементов СВЧ ИС в Advanced Design System

Описание пакетов представлено в таблицах 6.1 – 6.21.

Таблица 6.1 – Описание пакета routers

|  |  |
| --- | --- |
| **Модуль** | **Описание** |
| connection | Хранит метод роутера для проверки соединения с сервером |
| processing | Содержит роутеры для пакетов обработки данных |
| workspace | Содержит роутер для пакета, получающего информацию о подключённом workspace |

Таблица 6.2 – Описание пакета services

|  |  |
| --- | --- |
| **Пакет** | **Описание** |
| simulation | Пакет, содержащий модули, выполняющие моделирование |
| processing | Пакет, содержащий модули, взаимодействующие с элементами или схематиками |
| workspace | Содержит модуль, проверяющий подключение к workspace |

Таблица 6.3 – Описание пакета simulation

|  |  |
| --- | --- |
| **Модуль** | **Описание** |
| common | Модуль, содержащий общие методы для моделирования |
| element\_simulation | Модуль, выполняющий моделирование элемента |
| schematic\_simulation | Модуль, выполняющий моделирование схематика |

Таблица 6.4 – Описание пакета processing

|  |  |
| --- | --- |
| **Модуль** | **Описание** |
| common | Модуль, содержащий общие методы обработки данных |
| elements | Модуль, который получает список элементов PDK и элементов EDA |
| element\_parameters | Модуль, который получает список параметров элемента |
| library | Модуль, получающий информацию о PDK |
| schematic | Модуль, получающий список схематиков проекта и список элементов и параметров заданного схематика |

Таблица 6.5 – Описание пакета workspace

|  |  |
| --- | --- |
| **Модуль** | **Описание** |
| workspace | Модуль, проверяющий подключение к workspace |

Таблица 6.6 – Описание пакета schemas

|  |  |
| --- | --- |
| **Модуль** | **Описание** |
| common | Модуль, содержащий общие модели |
| processing | Модуль, который содержит модели для обработки данных |
| workspace | Модуль, который содержит модель для проверки подключения к workspace |

UML-диаграмма пакета routers представлена на рисунке 6.2.

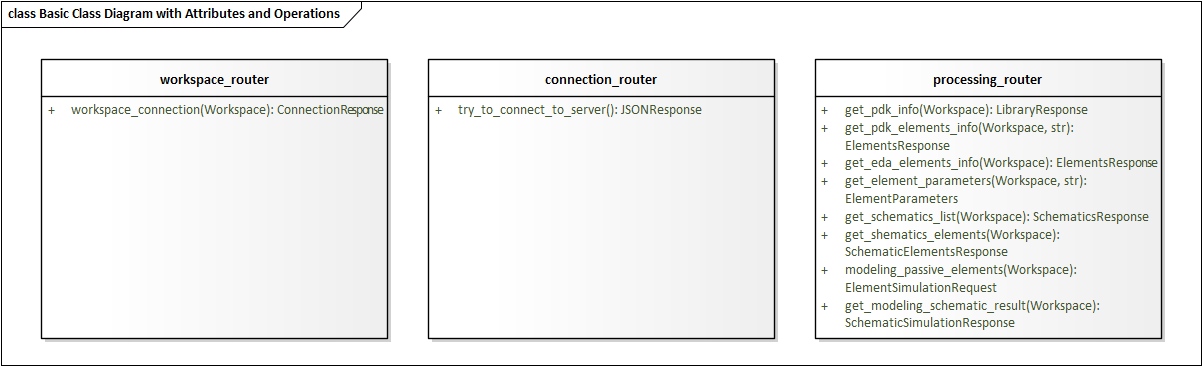


Рисунок 6.2 – UML-диаграмма пакета routers

Таблица 6.7 – Описание модуля connection

|  |  |
| --- | --- |
| **Название** | **Описание** |
| try\_to\_connect\_to\_server | Метод роутера для проверки соединения с сервером |

Таблица 6.8 – Описание модуля processing

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_pdk\_info | Отправляет клиенту данные о текущем PDK для выбранного Workspace |
| get\_pdk\_elements\_info | Отправляет клиенту данные об элементах PDK для выбранного Workspace |
| get\_eda\_elements\_info | Отправляет клиенту данные об элементах EDA для выбранного Workspace |
| get\_element\_parameters | Отправляет клиенту данные о параметрах для выбранного элемента |

Окончание таблицы 6.8

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_element\_symbol\_result | Отправляет клиенту информацию об УГО переданного элемента |
| get\_schematics\_list | Метод роутера для получения схематиков проекта |
| get\_shematics\_elements | Метод роутера для получения элементов и их параметров из схематиков |
| modeling\_passive\_elements | Метод роутера для моделирования пассивных элементов |
| get\_modeling\_schematic\_result | Отправляет клиенту список файлов с результатами моделирования схематика |

Таблица 6.9 – Описание модуля workspace

|  |  |
| --- | --- |
| **Название** | **Описание** |
| workspace\_connection | Необходимо передать JSON - файл, хранящий путь до Workspace |

UML-диаграмма пакета services представлена на рисунке 6.3.

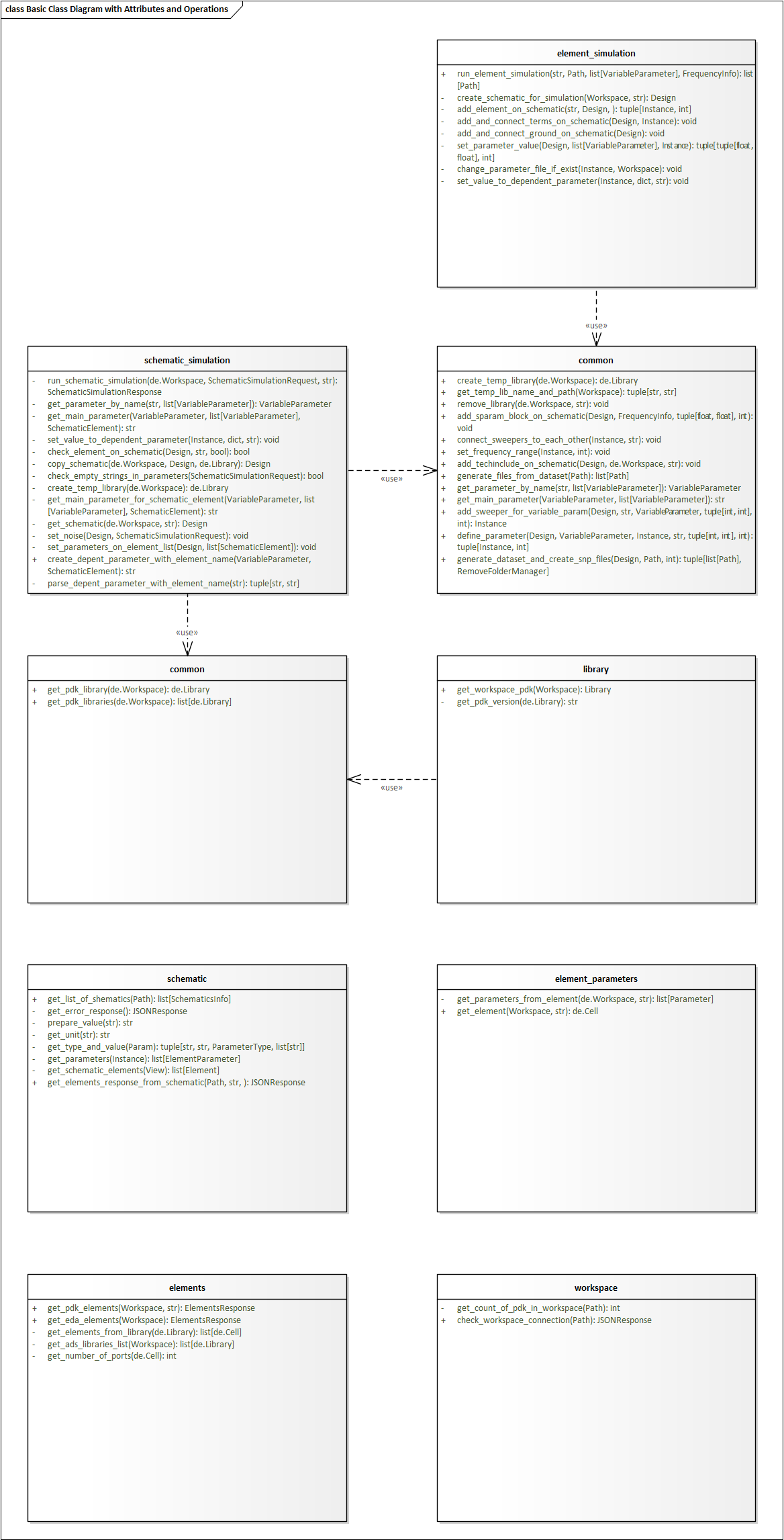


Рисунок 6.3 – UML-диаграмма пакета services

Таблица 6.10 – Описание модуля common пакета simulation

|  |  |
| --- | --- |
| **Название** | **Описание** |
| create\_temp\_library | Создает временную библиотеку |
| get\_temp\_lib\_name\_and\_path | Генерирует имя и путь для временной библиотеки |
| remove\_library | Удаление библиотеки со схемой |
| add\_sparam\_block\_on\_schematic | Добавление блока моделирования S-параметров на схему |
| connect\_sweepers\_to\_each\_other | Устанавливает связь между свиперами. |
| set\_frequency\_range | Устанавливает частоты для блока моделирования S-параметров |
| add\_techinclude\_on\_schematic | Добавляет на схему элемент PDK, с помощью которого возможно моделирование |
| generate\_files\_from\_dataset | Генерация SnP файлов из датасета |
| get\_parameter\_by\_name | Получение параметра по его имени |
| get\_main\_parameter | Рекурсивный метод поиска главного параметра, от которого зависят другие. |
| add\_sweeper\_for\_variable\_param | Добавляет свипер на схему для варьируемого параметра |
| define\_parameter | Определяет параметр: если он варьируемый, то добавляет на схему свипер, если нет - устанавливает переданное значение |
| generate\_dataset\_and\_create\_snp\_files | Создание датасета схемы с дальнейшим разбиением на SnP файлы |

Таблица 6.11 – Описание модуля element\_simulation

|  |  |
| --- | --- |
| **Название** | **Описание** |
| run\_element\_simulation | Моделирование элемента на переданном частотном диапазоне в различных конструктивных состояниях и формирование ответного пакета с результатами моделирования |
| create\_schematic\_for\_simulation | Создание временной библиотеки со схемой. |

Окончание таблицы 6.11

|  |  |
| --- | --- |
| **Название** | **Описание** |
| add\_element\_on\_schematic | Общий метод добавления элемента и свиперов на схему |
| add\_and\_connect\_terms\_on\_schematic | Соединение элемента с портами на схеме |
| add\_and\_connect\_ground\_on\_schematic | Создание и соединение земли на схеме |
| set\_parameter\_value | Установка значений для параметров элемента |
| change\_parameter\_file\_if\_exist | Проверяет, есть ли у элемента параметр File и устанавливает параметром полный путь до SnP файла |
| set\_value\_to\_dependent\_parameter | Установка значения зависимого параметра |

Таблица 6.12 – Описание модуля schematic\_simulation

|  |  |
| --- | --- |
| **Название** | **Описание** |
| run\_schematic\_simulation | Выполняет моделирование схематика на переданном частотном диапазоне  и формирует ответный пакет с результатами моделирования |
| copy\_schematic | Делает копию схематика |
| check\_element\_on\_schematic | Проверяет наличие элемента на схематике |
| check\_empty\_strings\_in\_parameters | Проверяет, подаются ли на вход значения варьируемого диапазона для свипируемого параметра |
| get\_main\_parameter\_for\_schematic\_element | Рекурсивный метод поиска главного параметра, от которого зависят другие |
| get\_parameter\_by\_name | Получение параметра по его имени |
| get\_schematic | Забирает схематик по LCV-имени из workspace |
| set\_noise | Устанавливает режим моделирования с шумами или без шумов |

Окончание таблицы 6.12

|  |  |
| --- | --- |
| **Название** | **Описание** |
| set\_parameters\_on\_element\_list | Устанавливает заданные параметры для каждого элемента из списка |
| create\_depent\_parameter\_with\_element\_name | Создает имя для ключа и значения зависимых параметров |
| parse\_depent\_parameter\_with\_element\_name | Парсинг имени для ключа и значения зависимых параметров |

Таблица 6.13 – Описание модуля common

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_pdk\_library | Получает PDK в Workspace |
| get\_pdk\_libraries | Получает список всех PDK, добавленных в Workspace |

Таблица 6.14 – Описание модуля element\_parameters

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_parameters\_from\_element | Получает данные о параметрах элемента |
| get\_element | Парсит строку по типу  "ads\_rflib\_\_C\_\_symbol" и "ads\_rflib:C:symbol"  и находит соответствующий элемент |

Таблица 6.15 – Описание модуля elements

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_pdk\_elements | Получает данные об элементах PDK |
| get\_eda\_elements | Получает данные об элементах EDA |
| get\_elements\_from\_library | Получает элементы библиотеки |
| get\_ads\_libraries\_list | Получает список ADS-библиотек |

Таблица 6.16 – Описание модуля library

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_workspace\_pdk | Получает данные о текущих PDK |
| get\_pdk\_version | Получает версию PDK |

Таблица 6.17 – Описание модуля schematic

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_list\_of\_shematics | Метод возвращает список схематиков в Workspace и информацию об их портах |
| get\_error\_response | Возвращает ответ пользователю в случае отсутсвия Workspace с таким названием |
| prepare\_value | Метод убирает лишние символы для определения числа |
| get\_unit | Метод перевода единицы измерения в систему СИ |
| get\_type\_and\_value | Определяет тип параметра и его список значений |
| get\_parameters | Возвращает список параметров элемента на схеме в виде JSON |
| get\_schematic\_elements | Возвращает список элементов на схеме в виде JSON |
| get\_elements\_response\_from\_schematic | Метод получения всех элементов на схеме |

Таблица 6.18 – Описание модуля workspace

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_count\_of\_pdk\_in\_workspace | Проверка наличия PDK в Workspace |
| check\_workspace\_connection | Проверка возможности подключения к Workspace |

UML-диаграмма пакета schemas представлена на рисунке 6.4.

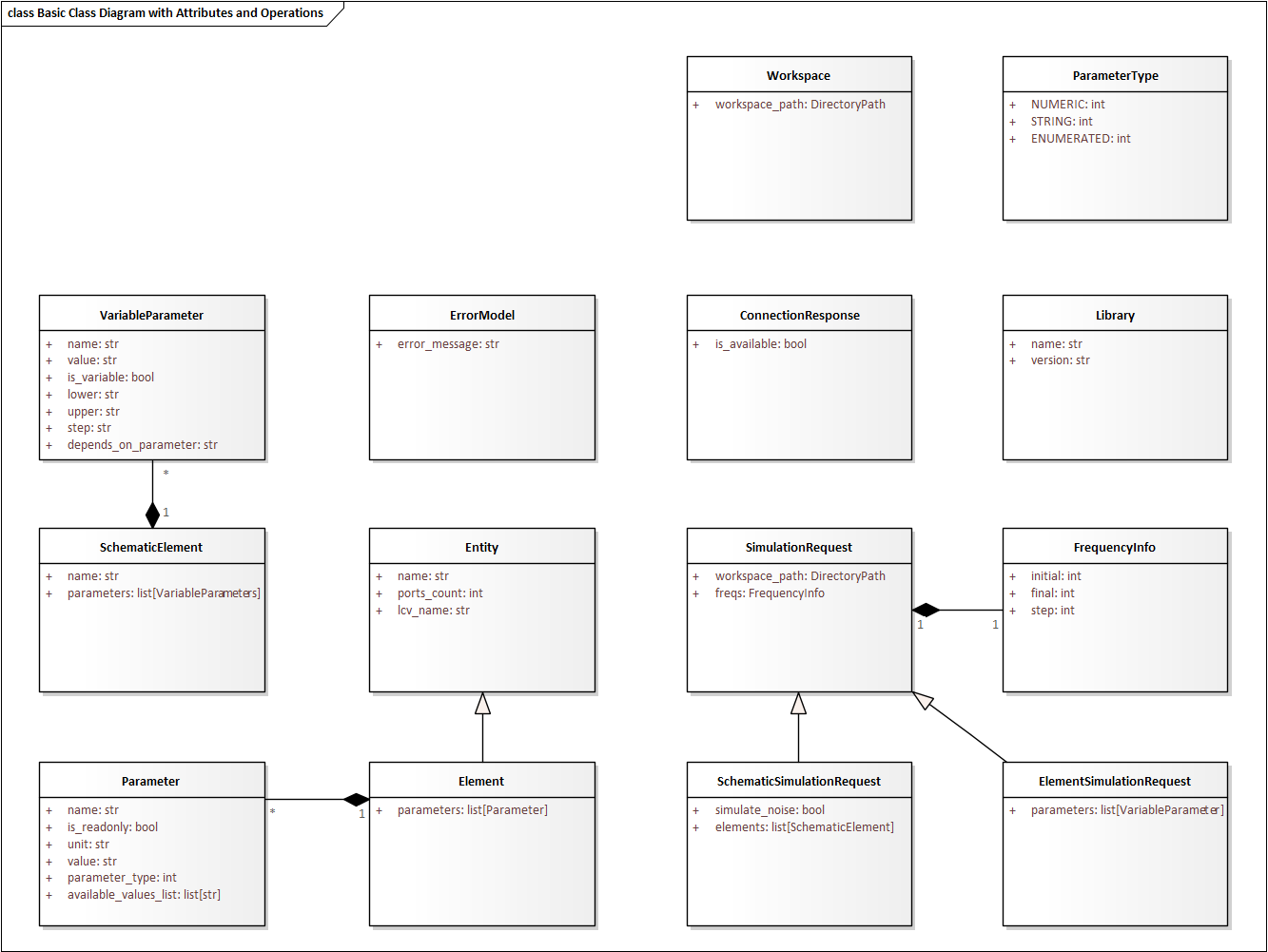


Рисунок 6.4 – UML-диаграмма пакета schemas

Таблица 6.19 – Описание модуля common

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Workspace | Модель валидации пути к Workspace |
| ErrorModel | Модель валидации JSON ответа подключения к Workspace |

Таблица 6.20 – Описание модуля processing

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Library | Модель хранения информации о PDK |
| Entity | Модель для описания сущности (элемента или схематика) |
| ParameterType | Перечисление типов параметров |

Окончание таблицы 6.20

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Parameter | Модель валидации параметра элемента |
| Element | Модель валидации элемента или схемы |
| VariableParameter | Модель валидации варьируемого параметра |
| SchematicElement | Модель валидации элемента схематика |
| FrequencyInfo | Модель валидации частоты моделирования |
| SimulationRequest | Базовый класс для запросов на моделирование |
| ElementSimulationRequest | Запрос на моделирование элемента библиотеки |
| SchematicSimulationRequest | Запрос на моделирование схематика |

Таблица 6.21 – Описание модуля workspace

|  |  |
| --- | --- |
| **Название** | **Описание** |
| ConnectionResponse | Модель валидации JSON ответа подключения к Workspace |

**7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ**

Серверная часть разрабатываемого плагина не предусматривает наличие графического интерфейса, она осуществляет прямое взаимодействие с САПР ADS через API. Используемые методы и классы API описаны в таблицах 7.1 и 7.2.

Таблица 7.1 – Используемые классы API

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Workspace | Workspace | Описывает взаимодействие с workspace |
| Library | Library | Описывает библиотеку |
| Cell | Cell | Описывает элемент библиотеки |
| Design | Design | Описывает схематик |
| Instance | Instance | Описывает элемент, добавленный на схему |
| CircuitSimulator | CircuitSimulator | Осуществляет моделирование |
| View | View | Описывает представление элемента |
| ParamBase | ParamBase | Описывает параметр элемента |

Таблица 7.2 – Некоторые используемые методы API

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых  данных | Описание |
| Worksapce.open() | str | void | Открывает workspace по заданному пути |
| Workspace.libraries | – | list[Library] | Возвращает список библиотек текущего workspace |
| Library.name | – | str | Возвращает имя библиотеки |
| Library.cells | – | list[Cell] | Возвращает список элементов библиотеки |
| Cell.view() | str | View | Возвращает требуемое представление элемента |
| View.get\_design() | – | Design | Возвращает представление схематика |
| Design.add\_instance() | CellviewRefLike, TScaledPoint, str, float | void | Добавляет элемент на схему |
| Design.get\_instance() | str | Instance | Возвращает элемент схематика по имени |
| Instance.parameters | – | ParamBaseCollection | Возвращает список параметров элемента |
| ParamBase.value | – | str | Возвращает значение параметра |
| CircuitSimulator.run\_netlist() | str, str, str, str | void | Запускает моделирование и сохраняет результаты в указанную директорию в формате .ds |

**8 ТЕСТИРОВАНИЕ ПЛАГИНА**

Тестирование позволяет убедиться в работоспособности программы, выявить ошибки при изменении функциональности.

**8.1 Функциональное тестирование**

При ошибке программа возвращает клиенту соответствующий код HTTP-запроса и json-файл с сообщением об ошибке. Примеры некоторых ответных пакетов в случае ошибки приведены на рисунках 8.1-8.4.

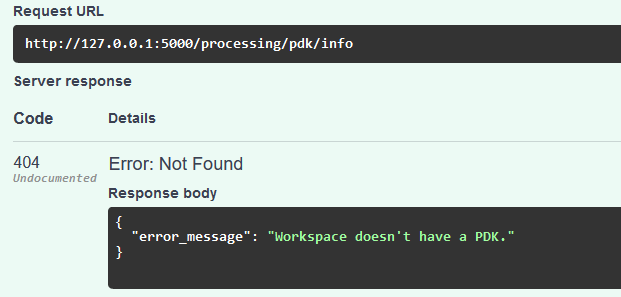


Рисунок 8.1 – Ответный пакет, если указанный workspace не содержит PDK



Рисунок 8.2 – Ответный пакет, если workspace не содержит указанной схемы



Рисунок 8.3 – Ответный пакет, если для моделирования был указан несуществующий элемент

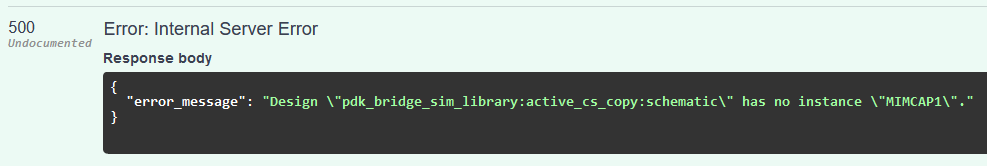


Рисунок 8.4 – Ответный пакет, если схема для моделирования не содержит указанного элемента

Примеры json-пакетов при успешном выполнении программы приведены на рисунках 8.5-8.10.



Рисунок 8.5– Ответный пакет с информацией о PDK

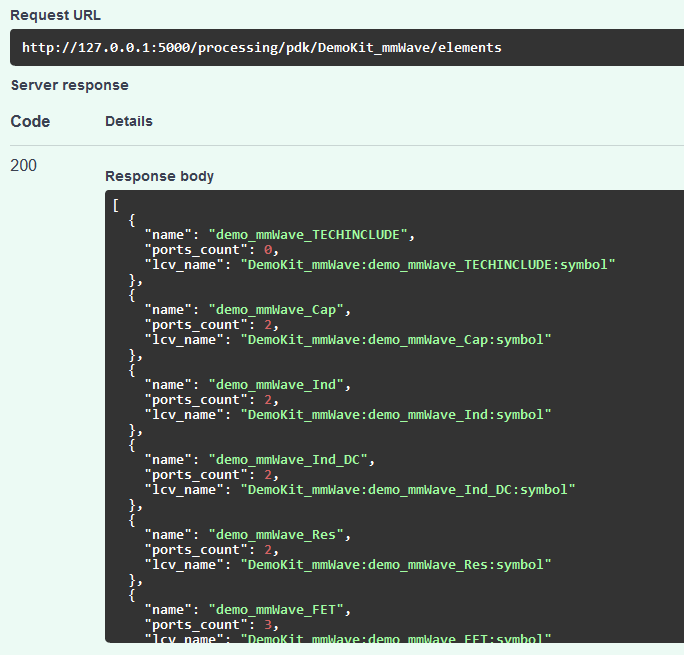


Рисунок 8.6 – Ответный пакет с элементами указанного PDK

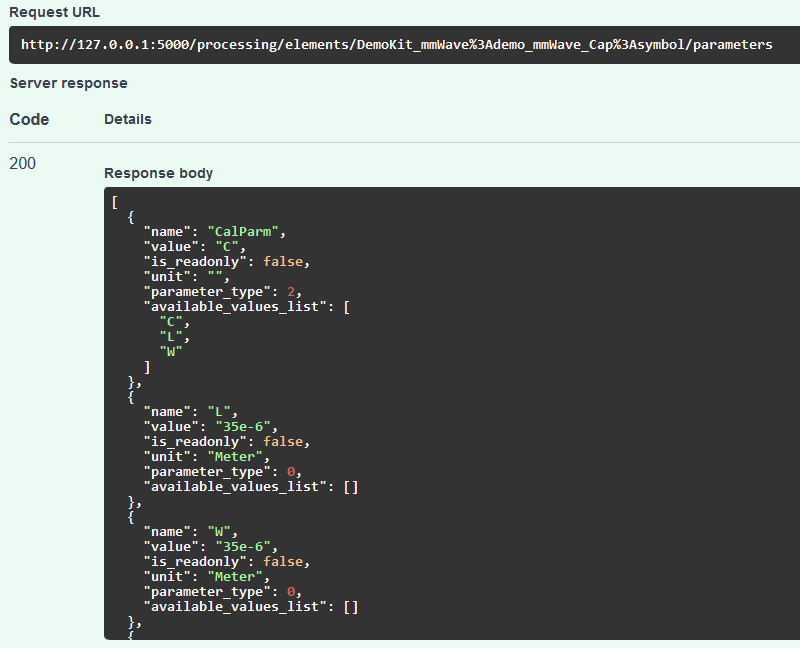


Рисунок 8.7 – Ответный пакет с параметрами указанного элемента

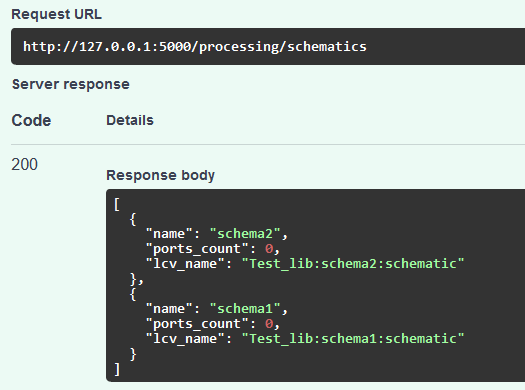


Рисунок 8.8 – Ответный пакет со схемами, которые содержатся в указанном workspace

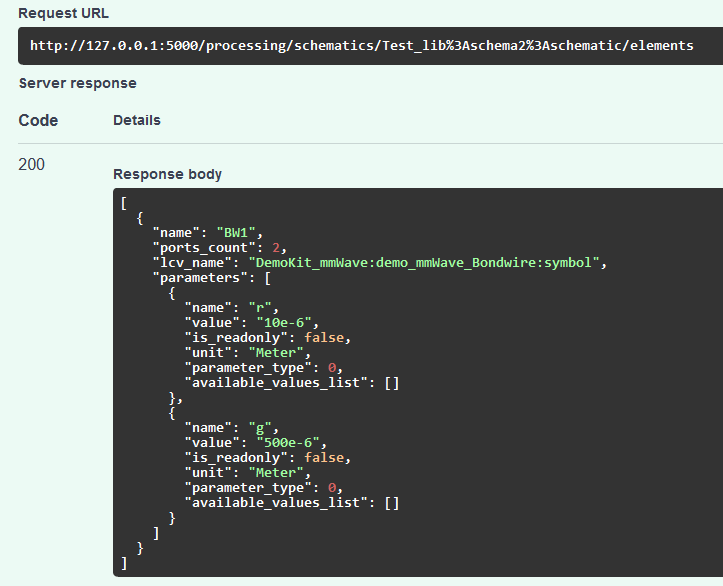


Рисунок 8.9 – Ответный пакет с элементами и их параметрами указанной схемы



Рисунок 8.10 – Ответный пакет, содержащий путь до файла с результатами моделирования элемента



Рисунок 8.11 – Ответный пакет, содержащий путь до файла с результатами моделирования схематика

**8.2 Модульное тестирование**

Для тестирования отдельных модулей была использована библиотека Pytest. Были протестированы модули: element\_parameters, element\_simulation, elements, library, schematic, schematic\_simulation, workspace. Программа покрыта тестами на 100%. Все модульные тесты выполняются успешно (рисунок 8.12).

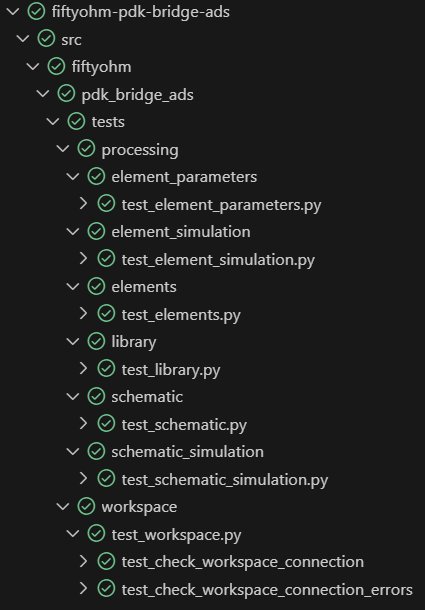


Рисунок 8.12 – Успешное выполнение модульных тестов

Описание модульных тестов плагина приведено в таблицах 8.1-8.7.

Таблица 8.1 – Тесты модуля workspace

|  |  |
| --- | --- |
| Название | Описание |
| test\_check\_workspace\_connection | Позитивный тест проверки возможности подключения к Workspace. |
| test\_check\_workspace\_connection\_error | Негативный тест проверки возможности подключения к Workspace. |

Таблица 8.2 – Тесты модуля library

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_workspace\_pdk\_correct | Позитивный тест на взятие информации о PDK из корректного workspace |

Окончание таблицы 8.2

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_workspace\_pdk\_empty | Позитивный тест на взятие информации о PDK из workspace без добавленного PDK |
| test\_get\_workspace\_pdk\_many\_pdk | Позитивный тест на взятие информации о PDK из workspace с двумя и более PDK |

Таблица 8.3 – Тесты модуля elements

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_pdk\_elements | Позитивный тест на получение данных об элементах PDK |
| test\_get\_eda\_elements | Позитивный тест на получение данных об элементах EDA |
| test\_get\_symbol\_for\_element | Позитивный тест на получение информации об УГО элемента |
| test\_get\_symbol\_for\_element\_error | Негативный тест на получение информации об УГО элемента |

Таблица 8.4 – Тесты модуля element\_parameters

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_parameters\_from\_element\_correct | Позитивный тест взятия параметров у элемента |
| test\_get\_parameters\_from\_element\_empty | Позитивный тест взятия параметров у элемента, у которого отсутствуют параметры |
| test\_get\_parameters\_from\_wrong\_named\_element | Негативный тест взятия параметров у элемента с некорректным именем |

Таблица 8.5 – Тесты модуля schematic

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_schematic\_list | Позитивный тест на получение списка схематиков заданного workspace |

Окончание таблицы 8.5

|  |  |
| --- | --- |
| Название | Описание |
| test\_get\_elements\_in\_schematic | Позитивный тест на получение списка элементов заданного схематика |
| test\_get\_elements\_in\_schematic\_error | Негативный тест на получение списка элементов заданного схематика |
| test\_get\_symbols\_for\_schematic\_elements | Позитивный тест на получение УГО элементов схематика |

Таблица 8.6 – Тесты модуля element\_simulation

|  |  |
| --- | --- |
| Название | Описание |
| test\_element\_simulation | Позитивный тест на получение результатов моделирования элемента |
| test\_simulation\_file\_with\_parameter\_file | Позитивный тест на получение результатов моделирования элемента с параметром «File» |
| test\_element\_simulation\_error | Негативный тест на получение результатов моделирования элемента |

Таблица 8.7 – Тесты модуля schematic\_simulation

|  |  |
| --- | --- |
| Название | Описание |
| test\_simulate\_schematic\_without\_sweep | Позитивный тест на получение результатов моделирования схематика без использования свипера |
| test\_simulate\_schematic\_with\_sweep | Позитивный тест на получение результатов моделирования схематика с использованием свипера |
| test\_simulate\_wrong\_named\_schematic | Негативный тест на получение результатов моделирования с некорректным именованием схематика |
| test\_simulate\_schematic\_with\_dependent\_parameters | Позитивный тест на получение результатов моделирования схематика с зависимыми параметрами |

**8.3 Нагрузочное тестирование**

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование. Тестирование производилось на виртуальной машине со следующей конфигурацией:

* Процессор AMD Ryzen 5 4600;
* 8 ГБ ОЗУ;
* Операционная система Windows 10 домашняя x64;
* Видеокарта VMware SVGA 3D.

Тестирование заключалось в моделировании 100000 схем с одним активным элементом. На рисунках 8.13 и 8.14 приведены результаты данного тестирования.

Рисунок 8.13 – График зависимости памяти ОЗУ от количества смоделированных схем

Рисунок 8.13 – График зависимости времени моделирования от количества смоделированных схем

Исходя из графика, представленного на рисунке 8.13, можно заметить линейное увеличение объёма занимаемой оперативной памяти в зависимости от количества смоделированных моделей до тех пор, пока оперативная память не закончится. Однако, после заполнения её объёма, работа плагина не останавливается, потому что начинает использоваться файл подкачки, восполняющий необходимость в оперативной памяти за счёт использования постоянной памяти.

Исходя из графика, представленного на рисунке 8.14, можно заметить, что среднее время моделирования схемы составляет 0,04 секунды и колеблется от 0,01 до 0,2 секунд. Это может быть связано с тем, что процесс моделирования схемы с одним активным элементом не нагружает оперативную память и процесс настолько, чтобы произошло замедление процесса.

# ЗАКЛЮЧЕНИЕ

В результате была разработана серверная часть плагина, получающего расчётные параметры элементов СВЧ ИС в системе автоматизированного проектирования Advanced Design System.

В процессе разработки были изучены:

* ADS API для программного взаимодействия с САПР ADS;
* Фреймворк FastAPI, использующийся для создания API;
* Библиотека pytest для написания модульных тестов Python-приложений.

По завершению работ были получены навыки проведения нагрузочного тестирования и взаимодействия с Python-стеком.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Документация “Keysight Advanced Design System API v.2024 upd.2”;
2. Официальный сайт FastAPI [электронный ресурс]. – URL: https://fastapi.tiangolo.com;
3. Официальный сайт Pydantic [электронный ресурс]. – URL: https://docs.pydantic.dev.