Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждения высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ПЛАГИНА****, ПОЛУЧАЮЩЕГО РАСЧЁТНЫЕ ПАРАМЕТРЫ ЭЛЕМЕНТОВ СВЧ ИС В ADVANCED DESIGN SYSTEM**

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 581

Боровкова В.Д.

« » 2024 г.

Руководитель:

к.т.н., доцент каф. КСУП

Калентьев А.А.

« » 2024 г.

# **СОДЕРЖАНИЕ**

[1 ОПИСАНИЕ САПР 3](#_Toc179815197)

[1.1 Информация о выбранной САПР 3](#_Toc179815198)

[1.2 Описание API 3](#_Toc179815199)

[1.3 Обзор аналогов плагина 6](#_Toc179815200)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 7](#_Toc179815201)

[3 ПРОЕКТ СИСТЕМЫ 7](#_Toc179815202)

[3.1 Диаграмма пакетов 7](#_Toc179815203)

[4 СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 11](#_Toc179815204)

1. **ОПИСАНИЕ САПР**

## 1.1 Информация о выбранной САПР

Advanced Design System (ADS) — это система автоматизированного проектирования (САПР) от компании Keysight Technologies, предназначенная для создания и моделирования принципиальных систем. Программа позволяет исследовать взаимодействие ВЧ и СВЧ компонентов и схем, в том числе монолитных интегральных схем (ИС). ADS предоставляет наборы для схемотехнического и электромагнитного моделирования, а также для проведения измерений с помощью X-параметров, необходимых для проведения нелинейного моделирования системы. Помимо этого, в САПР включена технология моделирования на основе S-параметров.

ADS и специализированные приложения на его основе широко применяются в области радиочастотного проектирования, микроволновой техники, разработки антенных систем, фильтров и цифровых высокоскоростных схем. Система поддерживает русскую локализацию [1].

Прямым аналогом разрабатываемого плагина является плагин PDK Bridge для AWR.

## 1.2 Описание API

API (Application Program Interface) – программный интерфейс приложения, набор функций, позволяющий взаимодействовать с программой через другие программы. API для ADS представлен на языке Python [2].

На текущий момент Python API в ADS работает в однопоточном режиме, поэтому подключиться к запущенному экземпляру САПР из другого процесса не получится. Возможные сценарии использования Python API в ADS:

1. Запуск python-кода из экземпляра ADS;
2. Запуск сервера вызова удалённых процедур из-под ADS;
3. Запуск python-кода в режиме автоматизации, который предоставляет возможность работать c консольным экземпляром ADS.

Некоторые используемые методы и классы API представлены в таблицах 1.1 и 1.2.

Таблица 1.1 – Некоторые 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 схематик |
| CircuitSimulator | CircuitSimulator | Осуществляет моделирование |
| View | View | Описывает представление элемента |
| ParamBase | ParamBase | Описывает параметр элемента |

Таблица 1.2 – 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 схематик |
| 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 |

## 1.3 Обзор аналогов плагина

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 1.1.

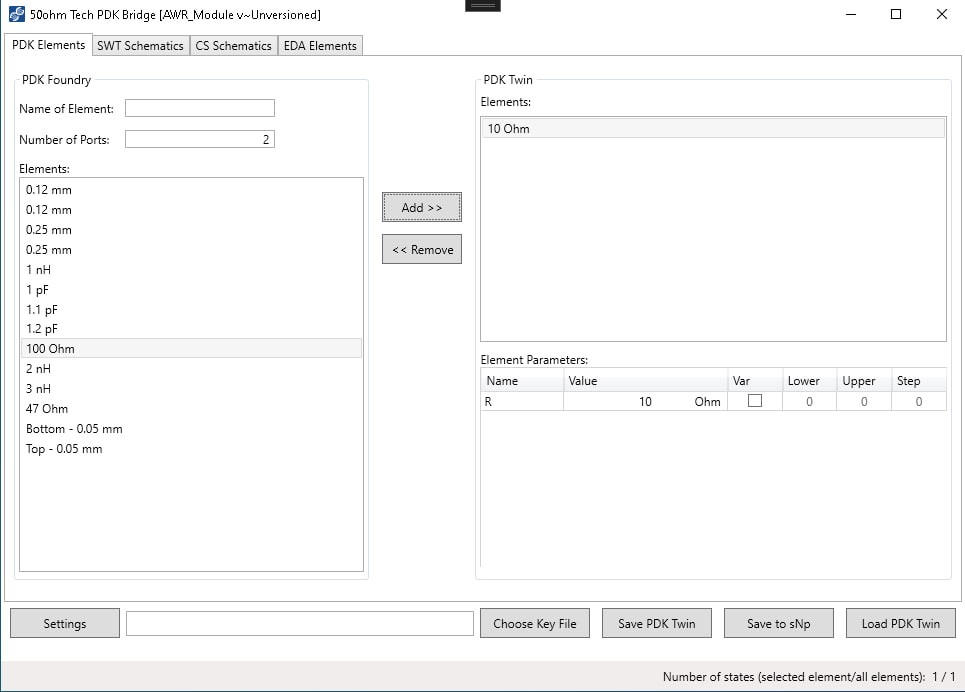


Рисунок 1.1 – PDK Bridge для AWR

# **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 ПРОЕКТ СИСТЕМЫ**

## 3.1 Диаграмма пакетов

Диаграмма пакетов разрабатываемого плагина представлена на рисунке 3.1.

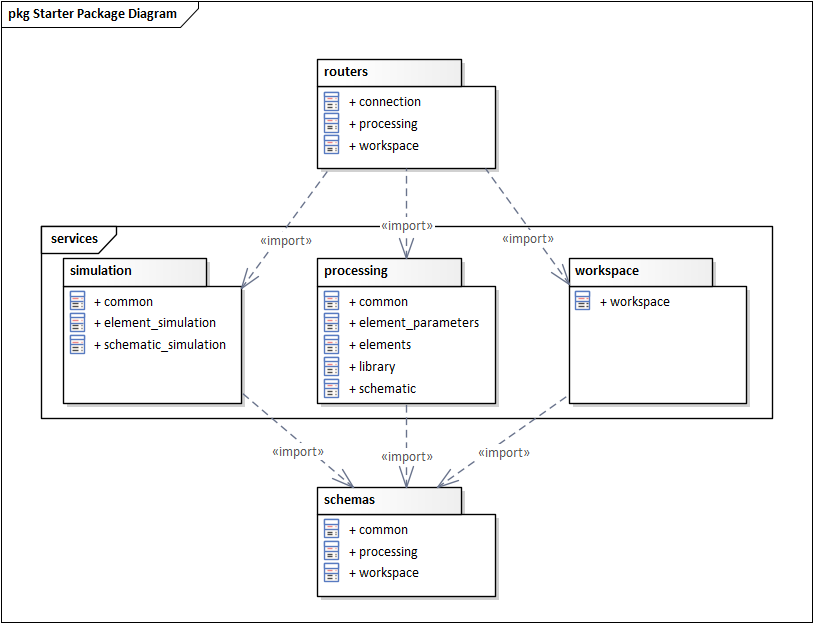


Рисунок 3.1 – Диаграмма пакетов серверной части плагина,получающего расчётные параметры элементов СВЧ ИС в Advanced Design System

Описание структуры PDK Twin представлено в таблицах 3.1 – 3.20.

Таблица 3.1 – 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 3.2 – 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 3.3 – 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 3.4 – 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 3.5 – Описание пакета workspace

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

Таблица 3.6 – 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 3.2.

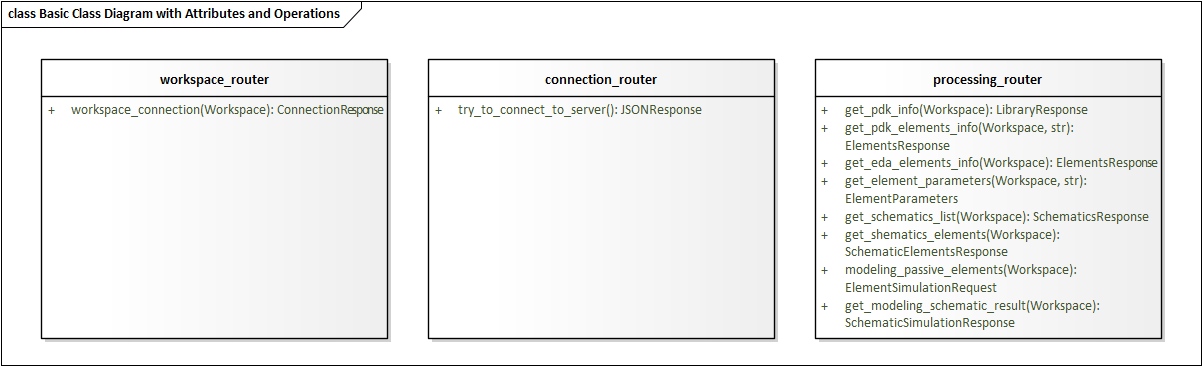


Рисунок 3.2 – UML-диаграмма пакета routers

Таблица 3.7 – Описание модуля connection

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

Таблица 3.8 – Описание модуля processing

|  |  |
| --- | --- |
| **Название** | **Описание** |
| get\_pdk\_info | Отправляет клиенту данные о текущем PDK для выбранного Workspace |

Окончание таблицы 3.8

|  |  |
| --- | --- |
| **Название** | **Описание** |
| 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 |
| 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 3.9 – Описание модуля workspace

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

UML-диаграмма пакета services представлена на рисунке 3.3.

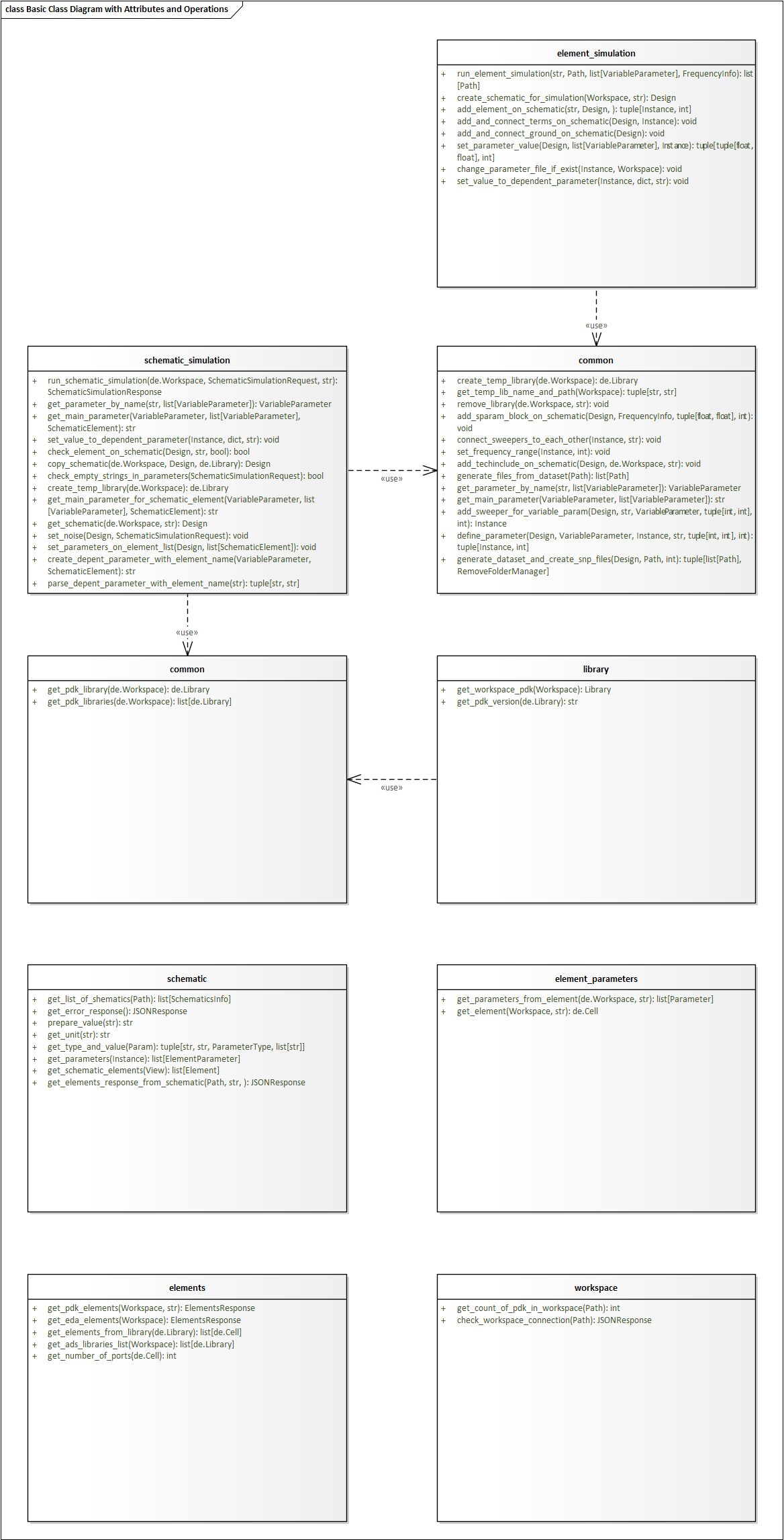


Рисунок 3.3 – UML-диаграмма пакета services

Таблица 3.10 – 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 разбиинием на SnP файлы |

Таблица 3.11 – 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 3.11

|  |  |
| --- | --- |
| **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 3.12 – 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 3.12

|  |  |
| --- | --- |
| **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 3.13 – Описание модуля common

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

Таблица 3.14 – 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 3.15 – 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 3.16 – Описание модуля library

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

Таблица 3.17 – 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 3.18 – 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 3.5.

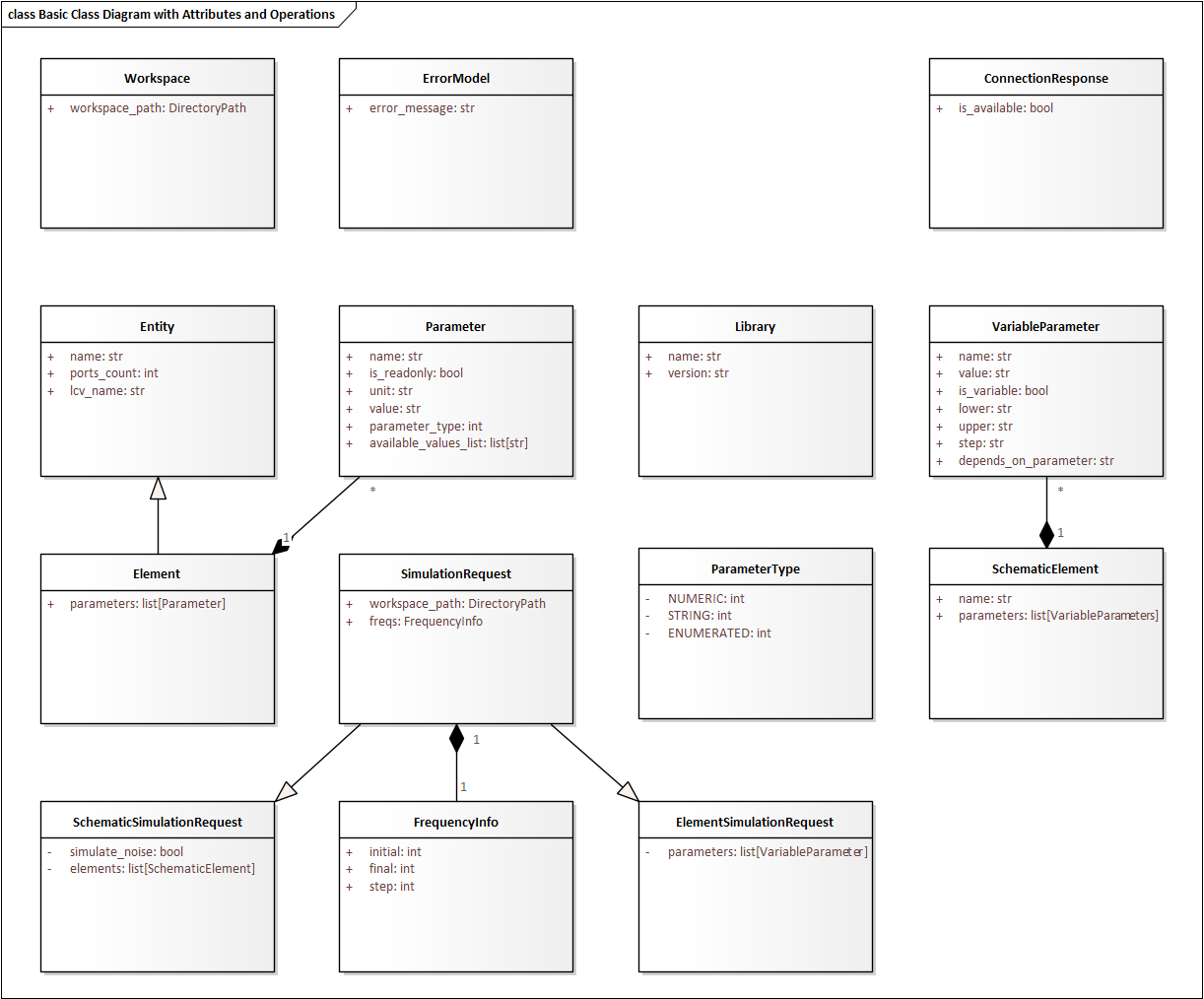


Рисунок 3.4 – UML-диаграмма пакета schemas

Таблица 3.19 – Описание модуля common

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

Таблица 3.20 – 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 3.20

|  |  |
| --- | --- |
| **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 3.21 – Описание модуля workspace

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

# **4 СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Официальный сайт Advanced Design System. [Электронный ресурс]. Режим доступа: свободный (дата обращения 07.10.24), <https://www.keysight.com/us/en/products/software/pathwave-design-software/pathwave-advanced-design-system.html>;
2. Документация “Keysight Advanced Design System API v.2024 upd.2”;