Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «ВОЛНОВОД» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 581

\_\_\_\_\_\_\_\_\_ Бислингер С.В.

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

Проверил:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Оглавление

1 Описание САПР 3

1.1 Описание программы 3

1.2 Описание API 4

1.3 Обзор аналогов 4

2 Описание предмета проектирования 4

3 Проект программы 4

3.1 Диаграмма классов 4

3.2 Макеты пользовательского интерфейса 5

Список использованных источников 8

**1 Описание САПР**

**1.1 Описание программы**

Проектирование новых видов и образцов машин, оборудования, устройств, аппаратов, приборов и других изделий представляет сложный и длительный процесс, включающий в себя разработку исходных данных, чертежей, технической документации, необходимых для изготовления опытных образцов и последующего производства, и эксплуатации объектов проектирования.

***Проектирование*** — это комплекс работ с целью получения описаний нового или модернизируемого технического объекта, достаточных для реализации или изготовления объекта в заданных условиях. В процессе проектирования возникает необходимость создания описания, необходимого для построения еще не существующего объекта. Получаемые при проектировании описания бывают окончательными или промежуточными. Окончательные описания представляют собой комплект конструкторско-технологической документации в виде чертежей, спецификаций, программ для ЭВМ и автоматизированных комплексов и т.д. [1]

Основной целью автоматизации является повышение качества исполнения процесса. Автоматизированный процесс обладает более стабильными характеристиками, чем процесс, выполняемый в ручном режиме. Во многих случаях автоматизация процессов позволяет повысить производительность, сократить время выполнения процесса, снизить стоимость, увеличить точность и стабильность выполняемых операций.

***Система автоматизированного проектирования (САПР)***— это организационно-техническая система, состоящая из комплекса средств автоматизации проектирования, взаимодействующего с подразделениями проектной организации и выполняющая автоматизированное проектирование.

Однако, с приходом на отечественный рынок иностранных систем, широкое распространение получили аббревиатуры *CAD* (Computer Aided Design), которую можно перевести, как проектирование с применением компьютера, и *CAD-system*, которую можно перевести, как система для проектирования с помощью компьютера. [2]

В настоящее время в среде специалистов по САПР многие термины утратили свой первоначальный смысл, а термин *САПР* теперь обозначает программу для автоматизированного проектирования.

Для реализации плагина будет использоваться программа   
«КОМПАС-3D» версии 20.

***Компас-3D*** – это система трехмерного моделирования деталей и сборок, используемая для проектирования изделий в машиностроении и строительстве — от изделий народного потребления до авиа-, судостроения и продукции военного назначения. [3]

Система «КОМПАС-3D» отличается проектированием изделий любой сложности, простотой освоения, бесплатной технической поддержкой, автоматизацией отраслевых задач и многим другим.

**1.2 Описание API**

Аббревиатура ***API*** расшифровывается как «Application Programming Interface» (интерфейс программирования приложений, программный интерфейс приложения). [4]

Сегодня встречаются задачи, решение которых не реализованы в CAD-системах. Чаще всего это очень узкоспециализированные задачи, которые встречаются на каком-то конкретном предприятии или подотрасли. Для решения подобных задач и их автоматизации используется API.

***API КОМПАС-3D*** — это ориентированные на прикладного программиста инструментальные средства разработки приложений (библиотек конструктивов, прикладных САПР) на базе системы КОМПАС. API КОМПАС-3D включает в свой состав API 5 и API 7.

Главным интерфейсом API системы КОМПАС является ***KompasObject.***[5] Методы этого интерфейса, реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа).

Ниже в таблице 1.1 представлены основные свойства и методы интерфейса KompasObject.

Таблица 1.1 – Методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Document3D() |  | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| GetParamStruct(short structType) | structType – тип интерфейса параметров | StructType2D | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| Visible |  | bool | Свойство видимости приложения |
| Quit() |  |  | Метод для закрытия активного окна приложения КОМПАС |

В таблице 1.2 представлены методы интерфейса ksEntity, необходимые для разработки плагина.

Таблица 1.2 – Методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблице 1.3 представлены свойства и методы интерфейса ksDocument2D, необходимые для разработки плагина

Таблица 1.3 – Методы интерфейса ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ksRectangle(  ksRectangleParam param, int style) | param – параметры прямоугольника.  style – стиль линии. | int | Получить указатель на прямоугольник на двумерной плоскости либо 0 в случае ошибки |
| ksCircle(  double xc, double yc, double rad, int style) | xc, yc – координаты центра окружности.  rad – радиус окружности.  style – стиль линии. | int | Получить указатель на окружность на двумерной плоскости либо 0 в случае ошибки |

В таблице 1.4 представлены свойства и методы интерфейса ksDocument3D, необходимые для разработки плагина.

Таблица 1.4 – Методы интерфейса ksDocument3D

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание | |
| Create (bool invisible, bool \_typeDoc) | invisible – признак режима редактирования документа (true – невидимый режим, false – | bool | Создать документ-модель (деталь или сборку) | |
|  | видимый режим),  typeDoc – тип документа (true – деталь, false – сборка). |  | |  |
| GetPart(int type) | type – тип компонента из перечисления Типы компонентов. | ksPart | | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.5 представлены методы интерфейса ksPart, необходимые для разработки плагина.

Таблица 1.5 – Свойства и методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| EntityCollection(short objType) | objType – тип объектов, содержащихся в массиве. | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | objType – тип объекта | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | Type – тип компонента | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | objType – тип объекта | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.6 представлены типы объектов документа-модели, необходимые для разработки плагина.

Таблица 1.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_unknown | Неизвестный (включает все объекты) |  |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_planeYOZ | Плоскость YOZ | ksPlaneParam |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_sketch | Эскиз | ksSketchDefinition |
| o3d\_face | Грань | ksFaceDefinition |
| o3d\_baseExtrusion | Базовая операция выдавливания | ksBaseExtrusionDefinition |

**1.3 Обзор аналогов**

**Пакет μWave Wizard**

Пакет μWave Wizard представляет собой специализированное программное обеспечение для проектирования СВЧ устройств, использующее метод согласованных мод (Mode-Matching), который наилучшим образом подходит для моделирования и оптимизации пассивных СВЧ устройств и антенн. Метод согласованных мод и его производные (быстрые гибридные методы граничного контура (MM / boundary contour) и конечных элементов (MM / finite-element method) обеспечивают непревзойденную скорость и точность вычислений для этих задач. [6]

Секрет быстродействия пакета μWave Wizard кроется в использовании метода согласованных мод (Mode-Matching) и его производных для анализа сложных трехмерных структур, где на первый взгляд, избежать полного 3D EM анализа невозможно. Тем не менее, для структур с очень сложной геометрией в пакете μWave Wizard сейчас реализован 3D FEM вычислитель, обеспечивающий ряд функциональных возможностей недоступных в методе согласованных мод (MM).

Мощный набор численных методов, реализованных в пакете μWave Wizard, дополняется простым и эргономичным пользовательским интерфейсом, обеспечивающим требуемую гибкость и открытость. Дальнейшее развитие пакета предполагает последовательную реализацию последних теоретических и математических подходов анализа СВЧ устройств.

Типичными приложениями для программы μWave Wizard являются пассивные волноводные структуры и компоненты, такие как волноводные и комбинированные фильтры, мультиплексоры, ответвители, переходы, рупорные антенны, кластерные фидерные системы, поляризаторы, мосты и многое другое. Библиотеки программы объединяют свыше 230 различных волноводных, комбинированных и коаксиальных элементов.

Интерфейс пакета показан на рисунке 1.1.



Рисунок 1.1 — Интерфейс пакета **μWave Wizard**

**2 Описание предмета проектирования**

Предметом проектирования является волновод.

Волновод являются полезным и функциональным каналом для распространения волны, широко применяющиеся в разных областях техники.[7]

Волноводы служат для передачи энергии в СВЧ трактах (например, от передатчика к антенне). Такой тракт обычно состоит из различных (по форме и размерам) радиволноводов, угловых изгибов и пр. Для сочленения радиоволноводов разных поперечных сечений применяются плавные волноводные переходы с переменным сечением (например, рупорный переход).

Под желание заказчика в волноводе могут изменять параметры, приведенные ниже:

1. A — диаметр отверстий в креплениях 3.5 до 4.8 мм;
2. B — радиус фаски креплений от 1 до 7 мм;
3. C — высота креплений от 65 до 100 мм;
4. D — ширина креплений от 80 до 150 мм;
5. E — толщина креплений от 10 до 20 мм;
6. F — длина волновода от 300 до 1000 мм;
7. G — ширина сечения от 30 до 100 мм;
8. O — толщина стенки сечения от 5 до 10 мм;
9. K — высота сечения от 15 до 50 мм;
10. H — расстояние от угла сечения до отверстия в креплении от 20 до 50 мм.

Плагин имеет следующие зависимости:

1. Ширина креплений D должна быть строго на 50 мм больше ширины сечения G;

2. Высота креплений C должна быть строго на 50 мм больше высоты сечения K;

3. Ширина сечения G должна быть в два раза больше высоты сечения K.

На рисунке 2.1 показан общий вид волновода:



Рисунок 2.1 – Общий вид волновода

На рисунках 2.2 и 2.3 представлены передний и верхний вид волновода с указанными параметрами:



Рисунок 2.2 – Вид спереди



Рисунок 2.3 – Вид сверху в разрезе

**3 Проект программы**

**3.1 Диаграмма классов**

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.[8] Целью создания диаграммы классов является графическое представление статической структуры элементов системы. Диаграмма классов плагина представлена на рисунке 3.1.

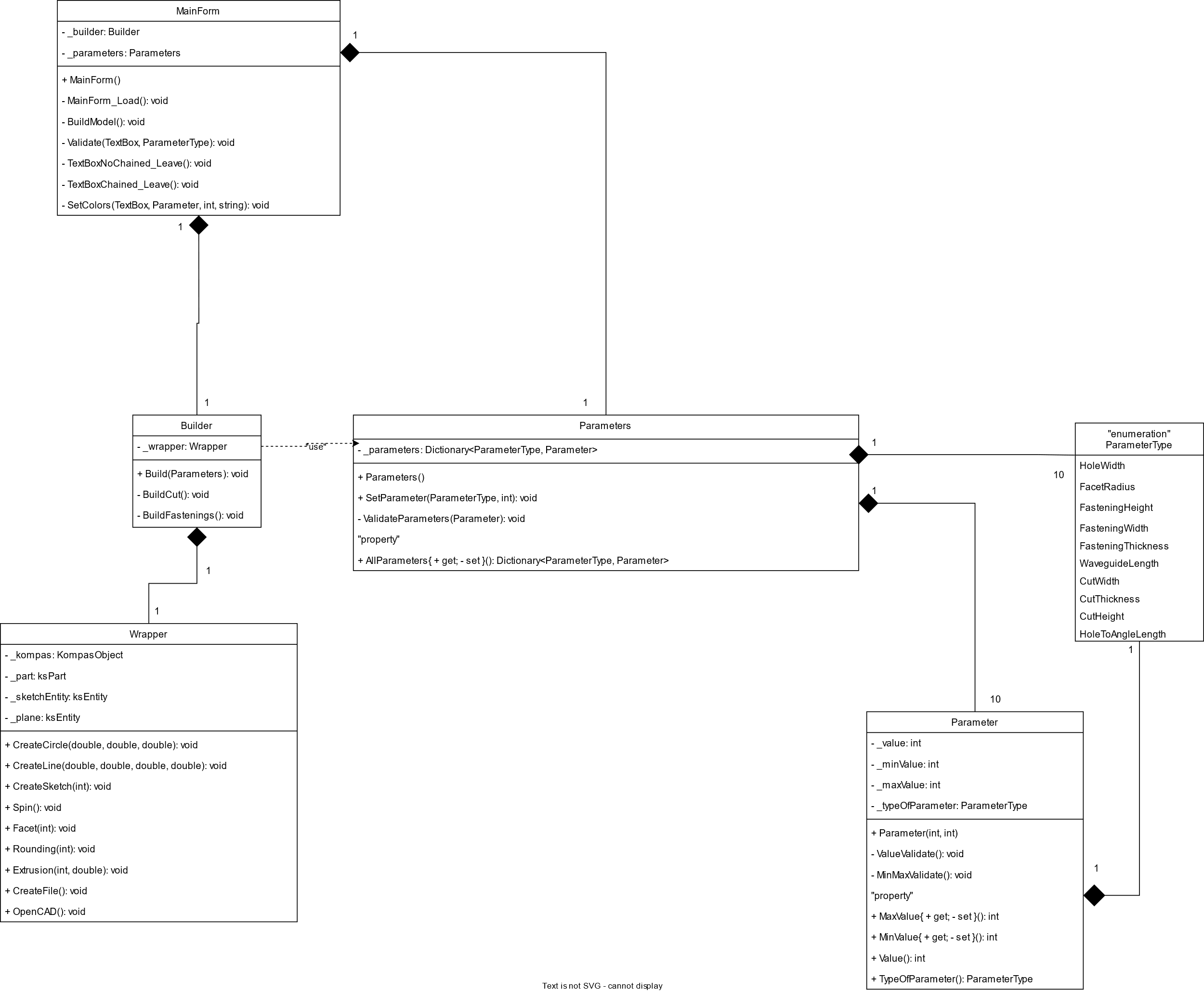




Рисунок 3.1 — Диаграмма классов плагина «Волновод»

В таблицах ниже представлена информация о свойствах и методах каждого из классов.

Таблица 3.1 − Свойства класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранит в себе объект построения |
| \_parameters | Parameters | Хранит в себе параметры для объекта построения |

Таблица 3.2 − Методы класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| BuildModel | − | Запуск построения модели по заданным параметрам |
| MainForm | − | Конструктор MainForm |
| Validate | TextBox, ParameterType | Проверка введённых данных по формату и валидация параметров |
| TextBoxNoChained\_Leave | − | Обработчик выхода из текстбокса без зависимых параметров |
| TextBoxChained\_Leave | − | Обработчик выхода из текстбокса с зависимыми параметрами |
| SetColors | TextBox. Paramater, int, string | Устанавливает цвета для текстбоксов по результатам проверки, int − выбор цвета для закраски, string – передаваемый текст ошибки |
| MainForm\_Load | − | Инициализация параметров при загрузке формы |

Таблица 3.3 − Свойства класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parameters | Dictionary<ParameterType, Parameter> | Хранит в себе словарь всех параметров |

Таблица 3.4 − Методы класса Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходные параметры | Описание |
| ValidateParameters | Parameter | − | Валидирует зависимые параметры |

Продолжение таблица 3.4

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходные параметры | Описание |
| SetParameter | ParameterType, int | − | Устанавливает параметр |
| AllParameters | − | Dictionary<ParameterType, Parameter> | Возвращает и задаёт словарь параметров |
| Parameters | − | − | Конструктор класса |

Таблица 3.5 − Свойства класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_wrapper | Wrapper | Хранит в себе объект обёртки API |

Таблица 3.6 − Методы класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| Build | Parameters | Построение модели по заданным параметрам |
| BuildCut | Parameters | Построение сечения волновода |
| BuildFastenings | Parameters | Построение креплений волновода |

Таблица 3.7 − Свойства класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_maxValue | int | Максимально допустимое значение параметра |
| \_minValue | int | Минимально допустимое значение параметра |
| \_value | int | Значение параметра |
| \_typeOfParameter | ParameterType | Тип параметра |

Таблица 3.8 − Методы класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| ValueValidate | − | Сравнивает полученное значение с максимальным и минимальным возможными |
| MaxValue | − | Свойство для полня \_maxValue |
| MinValue | − | Свойство для полня \_minValue |
| Value | − | Свойство для поля \_value |
| TypeOfParameter | − | Свойство для поля \_typeOfParameter |
| Parameter | int. int | Конструктов класса, int – максимальное значение, int – минимальное значение |
| MinMaxValidate | − | Сравнивает минимальное и максимальное значение |

Таблица 3.9 − Свойства класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_kompas | KompasObject | Поле, хранящее в себе экземпляр программы Компас |
| \_part | ksPart | Поле, хранящее в себе основную модель |
| \_sketchEntity | ksEntity | Поле, хранящее в себе текущий эскиз |
| \_plane | ksEntity | Поле, хранящие в себе выбранную плоскость |

Таблица 3.10 − Методы класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateCircle | double, double, double | Создание круга по точке и радиусу(double координаты x и y для точки, и double - радиус) |
| CreateLine | double, double, double, double | Создание линии по двум точкам |

Продолжение таблицы 3.10

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateSketch | int | Создание эскиза (по int выбираем базисную плоскость) |
| Spin | − | Вращение эскиза |
| Facet | int | Создание фаски (int - размер фаски) |
| Rounding | int | Создание скругления (int - радиус скругления) |
| Extrusion | int, double | Выдавливание эскиза (int - тип, double - глубина) |
| CreateFie | − | Создание файла |
| OpenCAD | − | Открытие Компас3D |

**3.2 Макеты пользовательского интерфейса**

Пользовательский интерфейс представляет собой форму для ввода параметров. На форме присутствует чертёж с параметрами для демонстрации параметров волновода и поля для ввода. Пользователь вводит значения самостоятельно, опираясь на подсказки, отображенные около полей. При нажатии на кнопку «Построить» проводится проверка зависимых параметров и, если условия соблюдены, строится 3D-модель волновода. На рисунке 3.2 представлен макет пользовательского интерфейса.

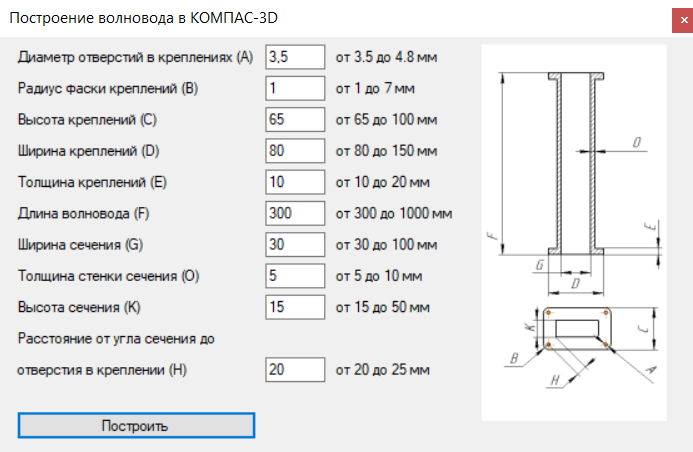


Рисунок 3.2 — Макет пользовательского интерфейса

Проверка правильности ввода значений проводится по ходу заполнения полей. Если поле заполнено неправильно, то есть пользователь ввел значение, превышающее границы, то оно подсвечивается красным цветом, сигнализирующем об ошибке.

Если же введены некорректные значения, и пользователь решил построить модель, несмотря на них, кнопка построения будет неактивна, пока не будут введены корректные значения. На рисунке 3.3 представлено превышение предела допустимого значения для диаметра отверстий в креплениях (5 мм, когда максимальное 4.8 мм).

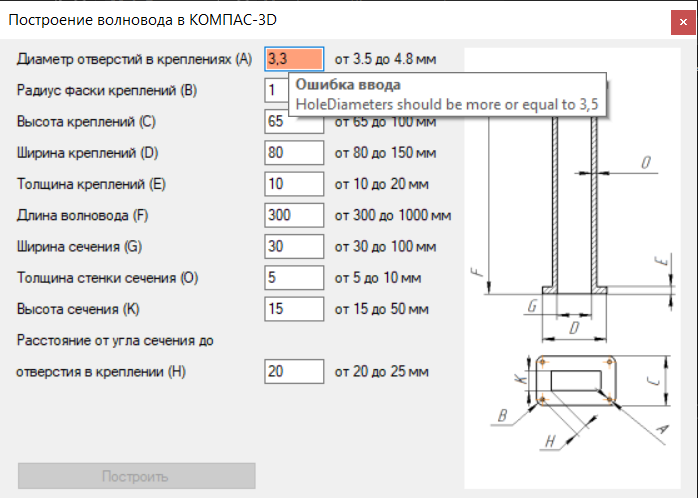


Рисунок 3.3 — Пример обработки ошибок при построении модели

**Список использованных источников**

1. Общие сведения о САПР [Электронный ресурс]. – Режим доступа: http://www.hi-edu.ru/e-books/xbook116/01/part-002.htm, свободный (дата обращения: 16.10.2024).

2. Обзор популярных систем автоматического проектирования (CAD) [Электронный ресурс]. – Режим доступа: https://www.pointcad.ru/novosti/obzor-sistem-avtomatizirovannogo-proektirovaniya, свободный (дата обращения: 16.10.2024).

3. КОМПАС-3D: О программе [Электронный ресурс]. – Режим доступа: https://kompas.ru/kompas-3d/about/, свободный (дата обращения: 18.10.2024).

4. Что такое API? [Электронный ресурс]. – Режим доступа: https://dev.by/news/chto-takoe-api-prostym-yazykom, свободный (дата обращения: 18.10.2024).

5. Работа с API КОМПАС-3D [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/ascon/blog/328088/, свободный (дата обращения: 20.10.2024).

6. µWave Wizard. [Электронный ресурс]. — Режим доступа: https://www.mician.com/products/μwave-wizard/ (дата обращения 20.10.2024).

7. ГОСТ 13317-89 — «Элементы соединения СВЧ трактов радиоизмерительных приборов. Присоединительные размеры.» (дата обращения 20.10.2024)

8. Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. – СПб.: Символ-Плюс, 2019. – 192 с.