Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**Разработка плагина «Микроволновка» для «КОМПАС-3D v19»**

Пояснительная записка к лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 587-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Адамчук И. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2021

Оглавление

[1 Описание САПР 3](#_Toc70530941)

[2 Описание API 5](#_Toc70530942)

[3 Описание аналогов 11](#_Toc70530943)

[3.1 Программа для моделирования SketchUp 11](#_Toc70530944)

[4 Описание предмета проектирования 13](#_Toc70530945)

[5 Диаграмма классов плагина 14](#_Toc70530946)

[6 Пользовательский интерфейс 17](#_Toc70530947)

[7 Тестирование 20](#_Toc70530948)

[7.1 Функциональное тестирование 20](#_Toc70530949)

[7.2 Модульное тестирование 24](#_Toc70530950)

[7.3 Нагрузочное тестирование 26](#_Toc70530951)

[Заключение 28](#_Toc70530952)

[Список литературы 29](#_Toc70530953)

[Приложение А 30](#_Toc70530954)

# 1 Описание САПР

Система КОМПАС-3D [1] предназначена для создания трехмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология реализует возможность получения модели типовых изделий на основе ранее спроектированного прототипа.

Функциональные возможности моделирования:

* + - создание простых и сложных поверхностей;
    - ЗD-моделирование с учетом поля допуска;
    - поддержка всех возможностей твердотельного моделирования;
    - ассоциативное задание параметров элементов;
    - функциональность для моделирования деталей из листового материала — команды создания листового тела, сгибов, отверстий, жалюзи, буртиков, штамповок и вырезов в листовом теле, замыкания углов, а также выполнения развертки полученного листового тела (в том числе формирования ассоциативного чертежа развертки);
    - булевы операции над типовыми формообразующими элементами;
    - построение вспомогательных прямых и плоскостей, эскизов, пространственных кривых (ломаных, сплайнов, различных спиралей);
    - создание конструктивных элементов — фасок, скруглений, отверстий, ребер жесткости, тонкостенных оболочек;
    - специальные возможности, облегчающие построение литейных форм — литейные уклоны, линии разъема, полости по форме детали (в том числе с заданием усадки);
    - создание любых массивов формообразующих элементов и компонентов сборок;
    - вставка в модель стандартных изделий из библиотеки, формирование пользовательских библиотек моделей;
    - наложение сопряжений на компоненты сборки (при этом возможность автоматического наложения сопряжений существенно повышает скорость создания сборки);
    - обнаружение взаимопроникновения деталей;
    - специальные средства для упрощения работы с большими сборками;
    - возможность гибкого редактирования деталей и сборок, в том числе с помощью характерных точек;
    - переопределение параметров любого элемента на любом этапе проектирования, вызывающее перестроение всей модели;
    - возможность создания таблиц переменных в моделях и графических документах;
    - работа с зеркальными исполнениями деталей и сборочных единиц, создание зеркального текста;
    - учет усадки, свойств и параметров материалов, а также технологии производства окончательного проекта.

Функциональные возможности разработки документации:

* + - разработка документации, отвечающей требованием ГОСТ, ISO, DIN или стандартами, установленными на предприятии;
    - создание чертежей и технологической документации проекта;
    - создание вариантов исполнения для деталей и сборочных единиц.

# 2 Описание API

Взаимодействие внешнего приложения или подключаемого модуля с системой КОМПАС (с функциями моделирования, математическими функциями ядра системы и пр.) осуществляется посредством программных интерфейсов (API).

На данный момент в КОМПАС существуют API двух версий: API 5 и API 7 [2]. Обе версии реализуют различные функции системы и не являются взаимозаменяемыми.

При создании подключаемых модулей в большей мере используется интерфейс API 5 [2].

Методы взаимодействия:

* **Cr****eate** – создать документ-модель (деталь или сборку);

Синтаксис:

Create (BOOL invisible, BOOL \_typeDoc)

Входные параметры:

invisible – признак режима редактирования документа (TRUE – невидимый режим, FALSE – видимый режим);

typeDoc – тип документа (TRUE - деталь, FALSE – сборка).

Возвращаемое значение:

True – в случае успешного завершения.

* **ActiveDocument3D** – получить указатель на интерфейс текущего документа трехмерной модели;

Синтаксис:

ActiveDocument3D()

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели ksDocument3D.

* **Document3D** – получить указатель на интерфейс документа трехмерной модели;

Синтаксис:

Document3D()

Возвращаемое значение:

Указатель на интерфейс документа трехмерной модели ksDocument3D.

* **GetPart** – получить указатель на интерфейс компонента в соответствии с заданным типом;

Синтаксис:

GetPart(long type)

Входные параметры:

type - тип компонента из перечисления “Типы компонентов”.

Типы компонентов:

pInPlace\_Part – компонент, редактируемый на месте;

pNew\_Part – новый компонент;

pEdit\_Part – редактируемый компоненты;

pTop\_Part – главный компонент, в составе которого находится новый или редактируемый или указанный компонент.

Возвращаемое значение:

Указатель на интерфейс компонента ksPart или IPart.

Примечание:

Тип задается для нового, редактируемого или главного компонента либо тип равен номеру компонента в документе;

Функция используется, чтобы получить доступ к компоненту документа;

Деталь или сборка являются компонентами. Сборка, в свою очередь, состоит из компонентов - деталей и подсборок.

* **GetDefaultEntity** –получить указатель на интерфейс объекта, создаваемого системой по умолчанию;

Синтаксис:

GetDefaultEntity(short objType)

Входные параметры:

objType – тип объекта.

Возвращаемое значение:

Указатель на интерфейс ksEntity или IEntity.

Типы объектов (objType):

o3d\_planeXOY 1 - плоскость XOY

o3d\_planeXOZ 2 - плоскость XOZ

o3d\_planeYOZ 3 - плоскость YOZ

o3d\_pointCS 4 - точка начала системы координат

o3d\_axisOX 71 - ось OX

o3d\_axisOY 72 - ось OY

o3d\_axisOZ 73 - ось OZ

* **N****e****wEntity** – создать новый интерфейс объекта и получить указатель на него;

Синтаксис:

NewEntity(short objType)

Входные параметры:

objType – тип объекта.

Возвращаемое значение:

Указатель на интерфейс ksEntity или IEntity.

* **B****egin****Edit** – войти в режим редактирования эскиза;

Синтаксис:

BeginEdit()

Возвращаемое значение:

SketchDoc – указатель на интерфейс IFragmentDocument.

* **EndEdit** – выйти из режима редактирования эскиза;

Синтаксис:

EndEdit()

Возвращаемое значение:

TRUE – в случае успешного завершения;

FALSE – в случае неудачи.

* **ksLineSeg** – создать отрезок;

Синтаксис:

long ksLineSeg(double x1, double y1, double x2,   
 double y2, long style)

Входные параметры:

x1, y1 – координаты первой точки отрезка;

x2, y2 – координаты второй точки отрезка;

style – стиль линии.

Возвращаемое значение:

Указатель на отрезок – в случае удачного завершения;

0 – в случае неудачи.

* **ExtrusionParam** – получить указатель на интерфейс параметров элемента выдавливания;

Синтаксис:

ExtrusionParam()

Возвращаемое значение:

Указатель на интерфейс объекта ksExtrusionParam или IExtrusionParam.

* **SetSketch** – изменить указатель на интерфейс эскиза элемента;

Синтаксис:

SetSketch(LPDISPATCH sketch)

Входные параметры:

sketch – указатель на интерфейс эскиза ksEntity или IEntity.

Возвращаемое значение:

TRUE – в случае успешного завершения.

* **ksCircle** – создать окружность;

Синтаксис:

long ksCircle(double xc, double yc, double rad,   
long style)

Входные параметры:

xc, yc – координаты центра окружности;

rad – радиус окружности;

style – стиль линии.

Возвращаемое значение:

указатель на окружность – в случае удачного завершения;

0 – в случае неудачи.

* **ksCreateInsertionFragment** – создать вставку фрагмента;

Синтаксис:

int ksCreateInsertionFragment(char\*fileName)

Входные параметры:

fileName – полное имя файла фрагмента.

Возвращаемое значение:

1 – в случае удачи.

* **SaveAs** – сохранить документ – модель (деталь или сборку) с новым именем;

Синтаксис:

SaveAs(BSTR fileName)

Входные параметры:

filename – имя файла документа.

Возвращаемое значение:

TRUE – в случае успешного завершения.

* **radius** – радиус скругления;

Тип данных: double.

Синтаксис:

radius = iFilletDefinition.GetRadius() Получить свойство

iFilletDefinition.SetRadius(radius) Установить свойство

* **directionType** – направление выдавливания;

Тип данных: short.

Синтаксис:

directionType=iBaseExtrusion.GetDirectionType() Получить свойство iBaseExtrusion.SetDirectionType(directionType) Установить свойство

Значения свойства:

dtNormal – прямое направление;

dtReverse – обратное направление;

dtBoth – в обе стороны;

dtMiddlePlane – от средней плоскости.

* **Depth** – глубина выдавливания;

Тип данных: double.

Синтаксис:

Depth = iObject.GetDepth(normal) Получить свойство

iObject.SetDepth(normal, Depth) Установить свойство

Входные параметры:

TRUE - глуби­на выдавлива­ния в прямом направлении;

FALSE - глу­бина выдавли­вания в обрат­ном направлении.

Примечание:

Позволяет считывать и устанавливать глубину выдавливания.

# 3 Описание аналогов

# 3.1 Программа для моделирования SketchUp

**SketchUp** — программа для 3D дизайна и архитектурного проектирования [3]. В основном используется для моделирования жилых домов, мебели, интерьера. Есть инструменты для проектирования лестниц, электропроводки, санитарно-технических коммуникаций и оборудования.

Основная особенность — почти полное отсутствие окон предварительных настроек. Все геометрические характеристики во время или сразу после окончания действия инструмента задаются с клавиатуры. Ещё одна ключевая особенность — это инструмент Push/Pull позволяющий любую плоскость «выдвинуть» в сторону, создав по мере её передвижения новые боковые стенки.

Также можно отметить следующие возможности:

* Поддержка плагинов для экспорта, визуализации, создания физических эффектов (вращения, движения, взаимодействия созданных объектов между собой и пр.);
* Библиотека компонентов (моделей), материалов и стилей рабочей области, которые можно пополнять своими элементами или загружать готовые из сети Интернет;
* Возможность создания динамических объектов (например: открытие дверцы шкафа по клику указателя);
* Возможность построения сечений объектов;
* Поддержка создания моделей реальных предметов и зданий;
* Поддержка создания «компонентов» — элементов модели, которые могут быть созданы, затем использованы много раз, а потом отредактированы — и изменения, сделанные в компоненте, отразятся во всех местах, где он использован.

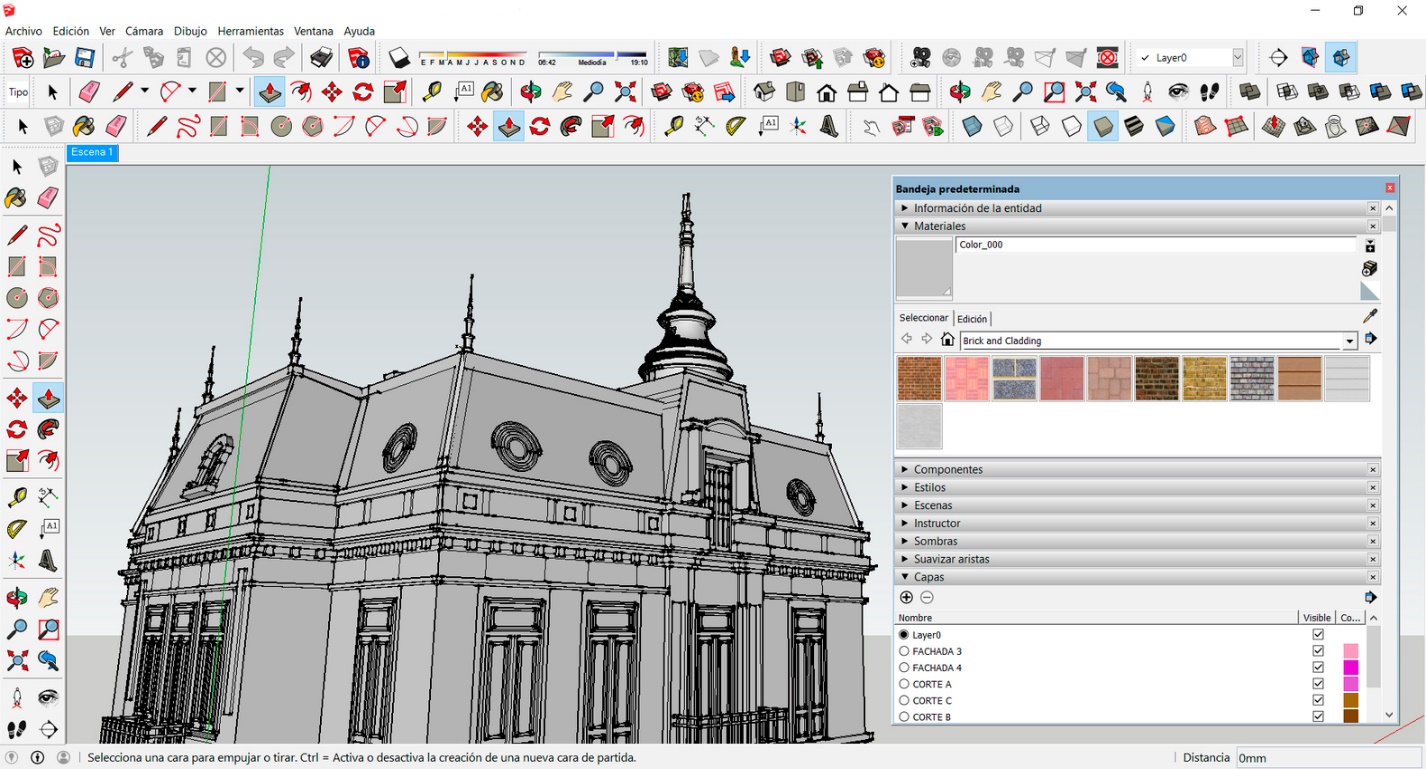
На рисунке 3.1 представлен интерфейс приложения “SketchUp”.

Рисунок 3.1 – Интерфейс приложения “SketchUp”

# 4 Описание предмета проектирования

Микроволновая печь [4] — электроприбор, позволяющий совершать разогрев водосодержащих веществ благодаря электромагнитному излучению дециметрового диапазона и предназначенный для быстрого приготовления, подогрева или размораживания пищи.

Параметры микроволновой печи:

* Габариты корпуса микроволновки: высота H (от 250 до 300 мм), длина L (от 400 до 500 мм), глубина W (от 300 до 450 мм).
* Габариты дверцы микроволновки: высота HD (от 200 до мм), длина LD (от 220 до мм).
* Габариты кнопок управления: диаметр D (от 20 до 40 мм), длина LB (от 60 до 100 мм), высота HB (от 30 до 50 мм).

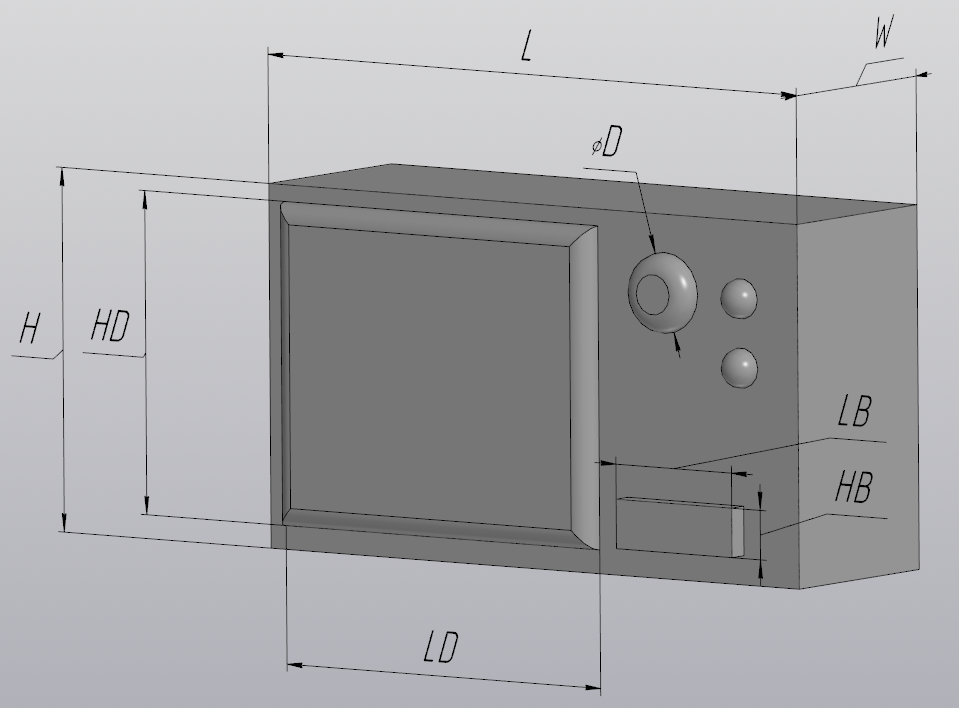
Образец 3D модели приведен на рисунке 4.1.

Рисунок 4.1 – Образец 3D модели микроволновки

# 5 Диаграмма классов плагина

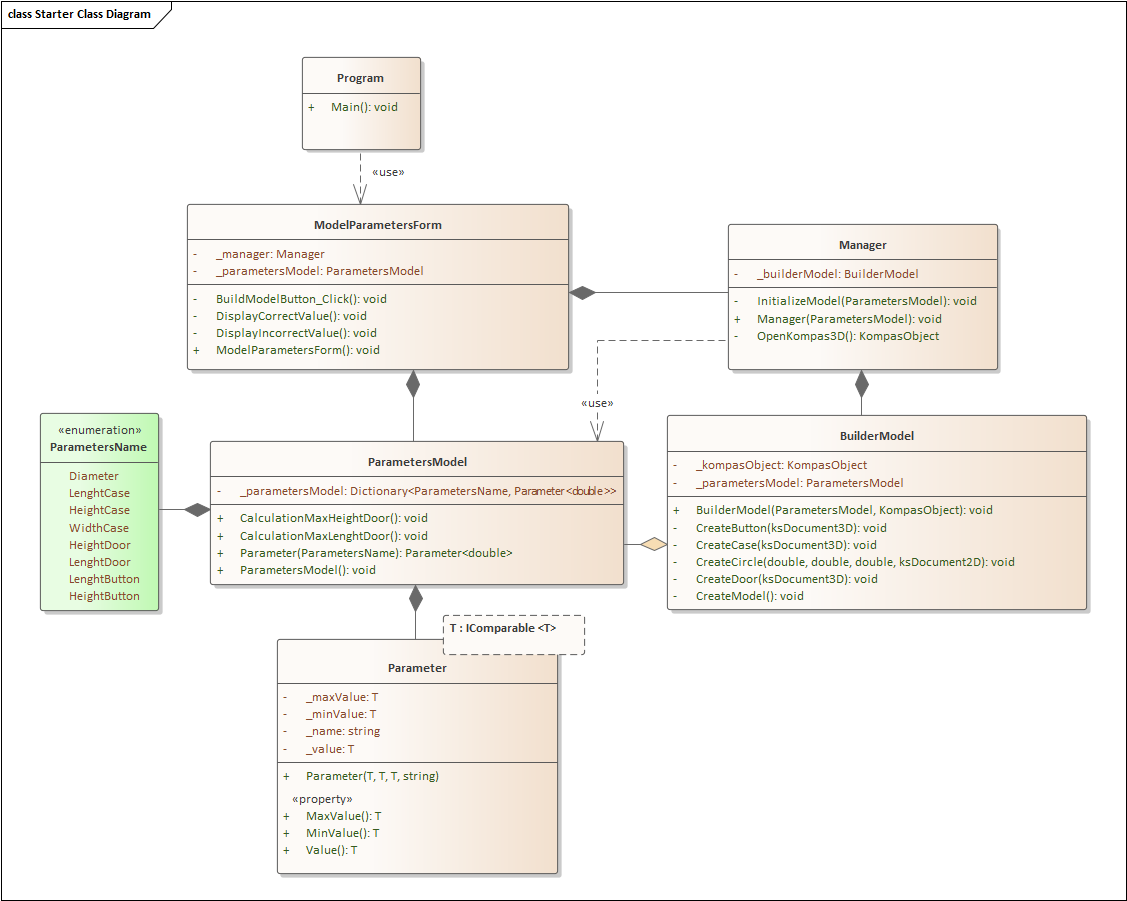
Диаграмма классов [5] описывает типы объектов системы и различного рада статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы. Рассмотрим диаграмму классов плагина на рисунке 5.1.

Рисунок 5.1 – Диаграмма классов плагина

Класс Parameter хранит информацию об одном параметре модели.

Класс ParametersModel хранит словарь со всеми параметрами модели, а также использует методы расчета максимальных значений параметров высоты и длины.

Класс BuilderModel хранит параметры модели, введенные пользователем. Используются методы для построения 3D модели, данные методы используют параметры модели.

Класс Manager хранит объект класса построителя 3D модели (BuilderModel), использует методы для начала работы с САПР.

Класс ModelParametersForm является формой пользовательского интерфейса. Реализует методы, используемые для взаимодействия с пользователем.

Во время разработки проекта, диаграмма классов подверглась изменениям:

1. Добавлена дополнительная функциональность: построение ручки от дверцы. В классе Builder добавлен новый метод CreateDoorHandle.
2. Добавлены новые методы в Builder для реализации выдавливания и закругления: Extrusion и Fillet.
3. В MainForm создан словарь \_correctValue для проверки параметров вносимые пользователем.
4. В классе ModelParameters убрана реализация словаря и методы пересчёта зависимого параметра.
5. Добавлен метод FillTextBoxes в класс MainForm для заполнения полей параметрами по умолчанию, также реализован метод GetNewModelParameters для создания новых параметров при изменении зависимого параметра и для уменьшения дублирования кода.
6. В классе ModelParameters убрана реализация словаря и методы пересчёта зависимого параметра.

Диаграмма, полученная после этапа реализации, изображена на рисунке 5.2.

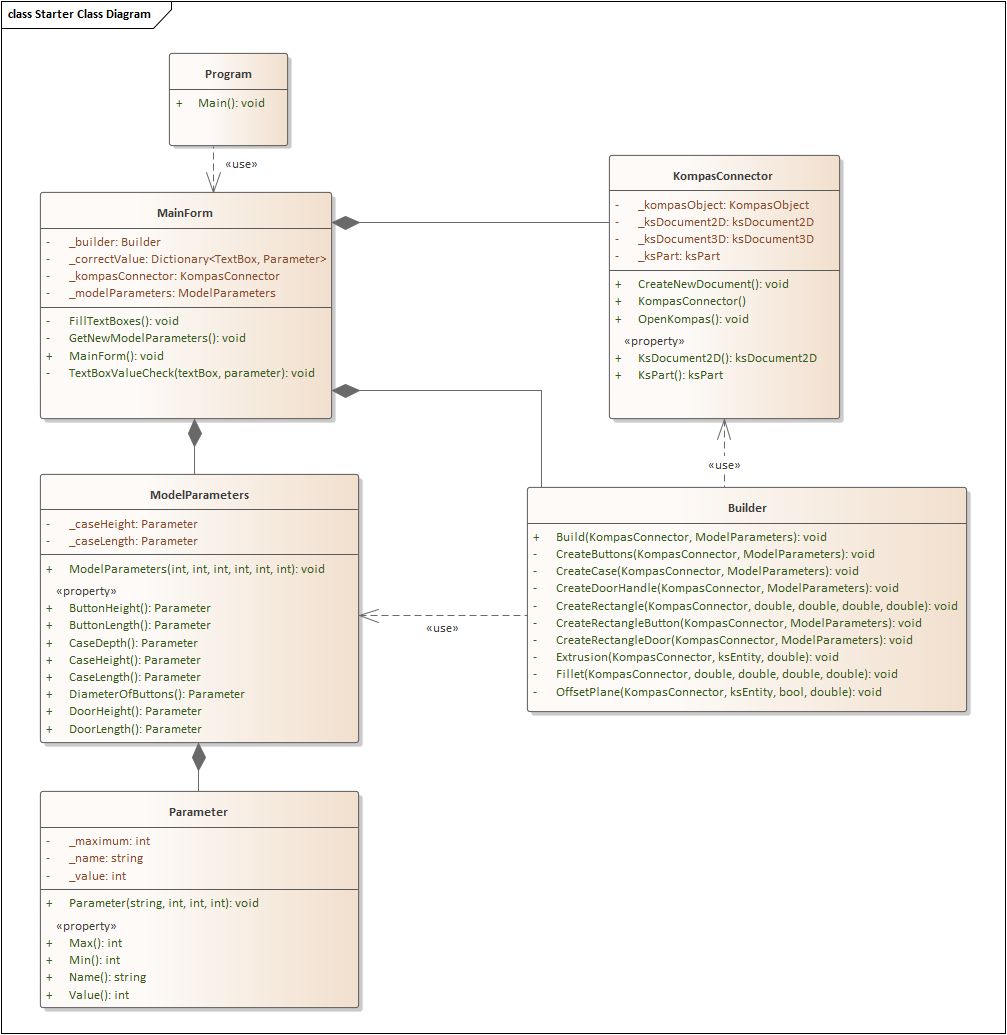


Рисунок 5.2 – Диаграмма классов плагина после реализации и добавления дополнительной функциональности

# 6 Пользовательский интерфейс

Пользовательский интерфейс [6] – интерфейс, обеспечивающий передачу информации между пользователем – человеком и программно-аппаратными компонентами компьютерной сети.

На рисунке 6.1 представлен макет пользовательского интерфейса плагина.

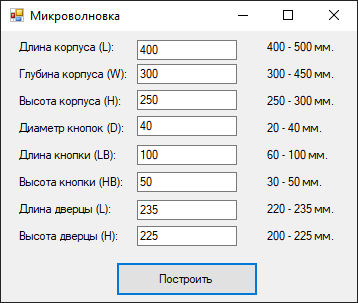


Рисунок 6.1 – Макет пользовательского интерфейса

Для каждого параметра предусмотрено два элемента:

- TextBox принимает значение параметра, вводимое пользователем. Окрашивается в красный цвет, при вводе некорректного значения. Если одно из значений некорректно, кнопка «Построить» становится неактивной. На рисунке 6.2 и 6.3 представлены макеты обработки некорректного значения параметра.

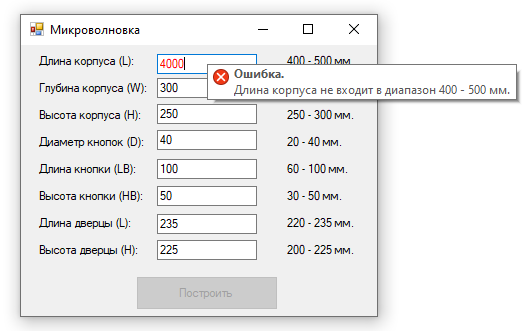


Рисунок 6.2 – Макет пользовательского интерфейса при некорректном вводе.

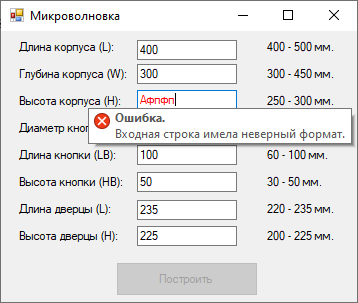


Рисунок 6.3 – Макет пользовательского интерфейса при некорректном вводе.

* Lable указывающий интервал возможных значений вводимого параметра.

После запуска для ввода доступно восемь параметров:

* Длина корпуса (L);
* Глубина корпуса (W);
* Высота корпуса (H);
* Диаметр кнопок (D);
* Длина кнопки (LB);
* Высота кнопки (HB);
* Длина дверцы (L);
* Высота дверцы (H);

Все поля заданы значениями параметров по умолчанию.

После клика по элементу «Построить» происходит открытие САПР КОМПАС – 3D и построение 3D модели.

# 7 Тестирование

# 7.1 Функциональное тестирование

Функциональное тестирование – это тестирование функциональности и поведения программы на соответствие требованиям функциональной спецификации. Функциональная спецификация определяет, что именно делает ПО, какие задачи оно решает. В данном случае будет проверяться правильность построения детали при различных входных параметрах, проведём тестирование максимальных, минимальных и параметров по умолчанию.

Построение модели с минимальными параметрами:

* Длина корпуса – 400;
* Глубина корпуса – 300;
* Высота корпуса – 250;
* Диаметр кнопок – 20;
* Длина кнопки – 60;
* Высота кнопки – 30;
* Длина дверцы – 220;
* Высота дверцы – 200;

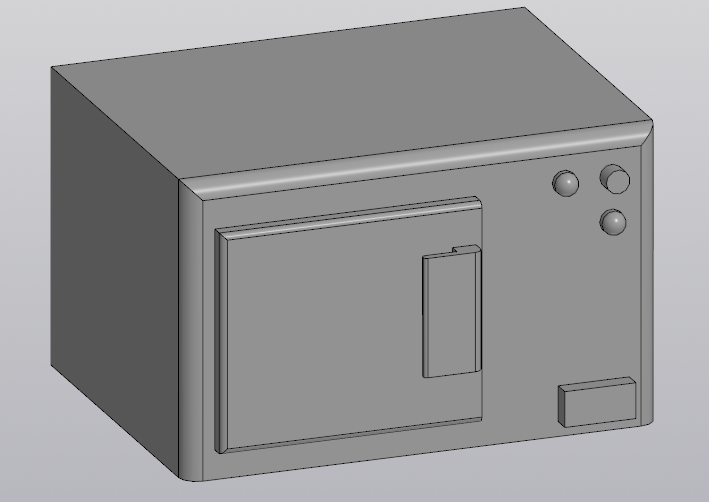
Модель микроволновки с минимальными параметрами показана на рисунке 7.1.

Рисунок 7.1 – Модель микроволновки с минимальными возможными параметрами

Построение модели с максимальными параметрами:

* Длина корпуса – 500;
* Глубина корпуса – 450;
* Высота корпуса – 300;
* Диаметр кнопок – 40;
* Длина кнопки – 100;
* Высота кнопки – 50;
* Длина дверцы – 335;
* Высота дверцы – 275;

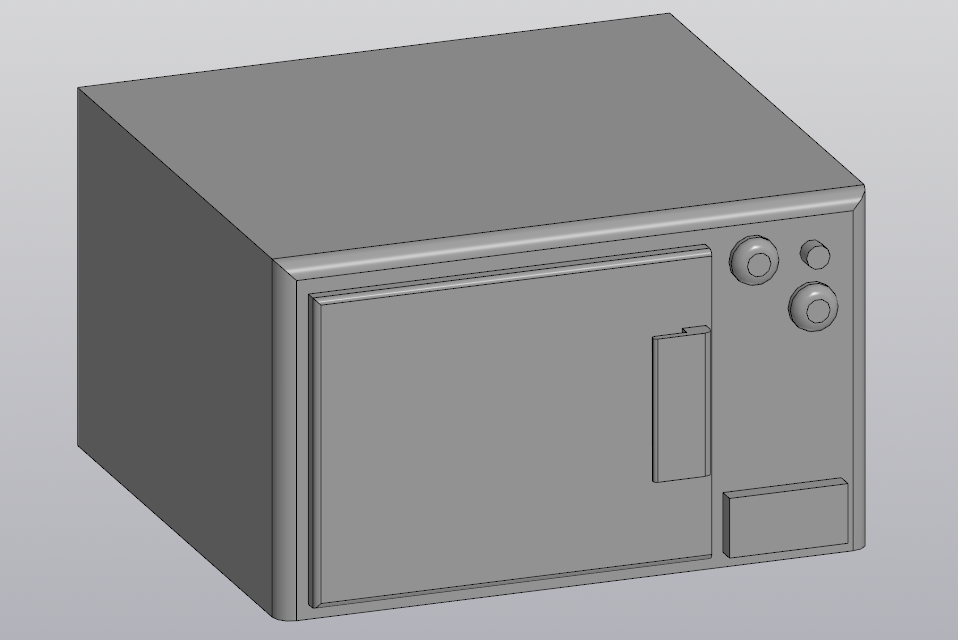
Модель микроволновки с максимальными параметрами показана на рисунке 7.2.

Рисунок 7.2 – Модель микроволновки с максимальными возможными параметрами

Построение модели с параметрами по умолчанию:

* Длина корпуса – 400;
* Глубина корпуса – 300;
* Высота корпуса – 250;
* Диаметр кнопок – 40;
* Длина кнопки – 100;
* Высота кнопки – 50;
* Длина дверцы – 235;
* Высота дверцы – 225;

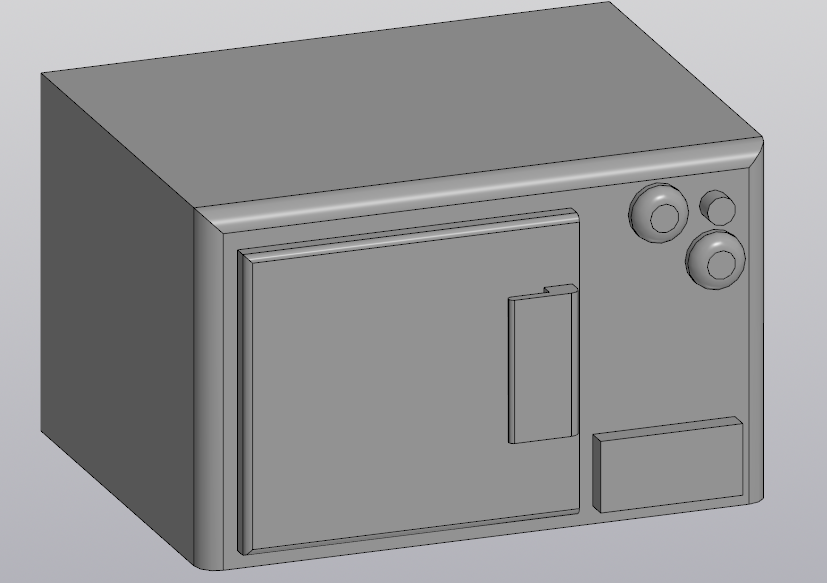
Модель микроволновки с максимальными параметрами показана на рисунке 7.3

Рисунок 7.3 – Модель микроволновки с параметрами по умолчанию

# 7.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio было проведено модульное тестирование. Модульное тестирование — тестирование отдельного элемента изолированно от остальной системы. Модульное тестирование предназначено для проверки правильности работы отдельно взятого элемента. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других элементов, тестируемый элемент должен быть максимально изолирован, то есть не использовать объекты и методы других классов [7].

На основе таблицы приведенных в приложении А тестовых сценариев (таблица А.1), проводилось тестирование корректности входных параметров 3D-модели.

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.13.1 для языков платформы .NET.

Результаты всех модульных тестов приведены на рисунке 7.3.

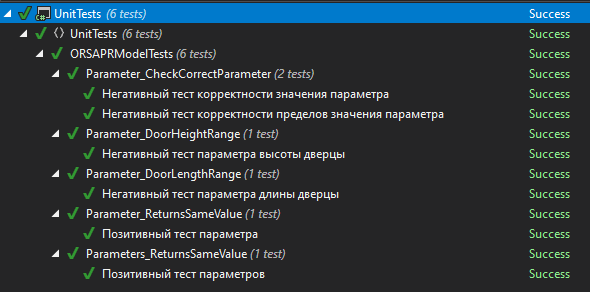


Рисунок 7.3 – Диалоговое окно пройденных тестов

Результат покрытия моделей Parameter и ModelParameters тестами приведен на рисунке 7.4. Цикломатическая сложность равна 79.

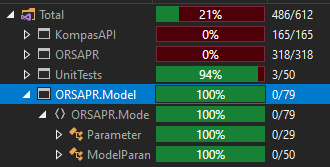


Рисунок 7.4 – Результат покрытия тестами

7.3 Нагрузочное тестирование

Нагрузочное тестирование – подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству) [8].

Нагрузочное тестирование проводилось на компьютере со следующими характеристиками:

* Процессор: AMD Ryzen 7 2700 4 ГГц;
* Оперативная память: 16 Гбайт DDR4 3200 ГГц;
* Графический процессор: NVIDIA GeForce GTX 2060;
* Операционная система: Windows 10 Pro 64-битная.

При попытке последовательного построения 200 моделей, «КОМПАС-3D» остановил дальнейшие построения на 170 модели, потребление ОЗУ достигло 16 Гбайт, потребление видеопамяти достигло 5.3 Гбайт.

Зависимость потребления ОЗУ от количества построенных моделей представлена рисунке. 7.5.

Рисунок 7.5 – Потребление ОЗУ от количества построенных моделей

Потребление оперативной памяти резко возросло после построения 30 моделей. Пик потребления был достигнут на 87 построении и составил 2665 МБ.

Зависимость времени построения от количества построенных моделей представлена рисунке. 7.6.

Рисунок 7.6 – Время построения от количества построенных моделей

Время построения не превышает 5 секунд вплоть до 169 построения. Последующее 170 построение заняло около 10 секунд из-за нехватки оперативной памяти и следующие построения модели не состоялись.

# Заключение

При выполнения лабораторных работ были изучены основные этапы проектирования программного продукта и его реализации, предметная область объекта проектирования, SDK «КОМПАС-3D». Было составлено техническое задание, разработан проект системы, составлена UML диаграмма классов, разработан макет пользовательского интерфейса, выполнено тестирование проекта.

В результате работы был разработан и реализован плагин для САПР «КОМПАС-3D», выполняющий построение 3D-модели микроволновки по задаваемым параметрам.

# Список литературы

КОМПАС – 3D [Электронный ресурс]. – URL: https://kompas.ru/ (дата обращения: 28.04.2021);

Базовые интерфейсы API системы КОМПАС [Электронный ресурс]. – URL: https://it.wikireading.ru/23741 (дата обращения: 28.04.2021);

Программа моделирования «SketchUp» [Электронный ресурс]. – URL: https://www.sketchup.com/ (дата обращения: 28.04.2021);

Микроволновая печь. [Электронный ресурс] – URL: https://ru.wikipedia.org/wiki/Микроволновая\_печь (дата обращения: 28.04.2021);

Мартин Фаулер UML Основы, 3-е издание. Краткое руководство по стандартному языку объектного моделирования – 2018 – 192 с.

Интерфейс пользователя [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Интерфейс\_пользователя (дата обращения: 28.04.2021);

Новые технологии в программировании: учебное пособие / А.А.Калентьев, Д.В.Гарайс, А.Е.Горяинов – Томск: Эль Контент, 2014.—176 с.

Нагрузочное тестирование [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Нагрузочное\_тестирование (дата обращения: 28.04.2021);

# Приложение А

(справочное)

Таблица А.1 – Тестовые сценарии

|  |  |  |
| --- | --- | --- |
| Тестовый метод | Входные параметры | Описание тестового случая |
| Parameters\_ReturnsSameValue() | DoorHeight – 200,  CaseDepth – 300,  DiameterOfButtons – 40, DoorLength – 220,  CaseLength – 400,  CaseHeight – 250,  ButtonLength – 100,  ButtonHeight – 50 | Позитивный тест параметров модели |
| Parameter\_DoorLengthRange() | CaseDepth – 300,  DiameterOfButtons – 30, CaseLength – 500,  CaseHeight – 250,  ButtonLength – 100,  ButtonHeight – 50, DoorLengthValue – 2000; | Негативный тест параметра длины дверцы |
| Parameter\_DoorHeightRange() | CaseDepth – 300,  DiameterOfButtons – 30, CaseLength – 500,  CaseHeight – 250,  ButtonLength – 100,  ButtonHeight – 50, DoorHeightValue – 1000; | Негативный тест параметра высоты дверцы |
| Parameter\_ReturnSameValue() | Name –"Параметр первый", Value – 250,  Min – 100,  Max – 500 | Позитивный тест параметра модели |
| Parameter\_CheckCorrectParameter() | Name –"Параметр первый", Value – 155,  Min – 90,  Max – 100 | Негативный тест корректности значения параметра |
| Parameter\_CheckCorrectParameter() | Name –"Параметр первый", Value – 155,  Min – 2000,  Max – 100 | Негативный тест корректности пределов значения параметра |