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)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.В. Паршаков

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**Оглавление**

[1 Введение 3](#_Toc122792379)

[2 Постановка и анализ задачи 4](#_Toc122792380)

[2.1 Описание предмета проектирования 5](#_Toc122792381)

[2.2 Выбор инструментов и средств реализации 6](#_Toc122792382)

[2.3 Назначение плагина 6](#_Toc122792383)

[3 Обзор аналогов 7](#_Toc122792384)

[4 Описание реализации 8](#_Toc122792385)

[4.1 Диаграмма классов 8](#_Toc122792386)

[5 Описание программы для пользователя 13](#_Toc122792387)

[6 Тестирование программы 16](#_Toc122792388)

[6.1 Функциональное тестирование 16](#_Toc122792389)

[6.2 Модульное тестирование 18](#_Toc122792390)

[6.3 Нагрузочное тестирование 19](#_Toc122792391)

[Заключение 22](#_Toc122792392)

[Список использованных источников 23](#_Toc122792393)

# 1 Введение

Автоматизация моделирования имеет огромное значение для развития науки, техники и производства в современном обществе. В настоящее время автоматизация – основной способ повышения производительности и эффективности труда инженерно-технических работников, занимающихся моделированием сложных устройств. Использование автоматизации в проектировании позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений и технологий в той или иной области техники. Она позволяет значительно повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением [1].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Раковины» для системы автоматизированного проектирования Компас 3D с помощью интегрированной среды разработки Visual Studio 2022 Сommunity. [2]

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

# 2 Постановка и анализ задачи

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой САПР Kompas 3D, строит модель Раковины. [3] Необходимо чтобы плагин позволял изменять входные параметры раковины, такие как:

* ширина раковины **W**;
* длина раковины ***D***;
* глубина раковины **H**;
* диаметр сливного отверстия **R**;
* диаметр отверстия под кран **R1**;
* координата X отверстия под фильтр **C**;
* координата Y отверстия под фильтр **С**.

# 2.1 Описание предмета проектирования

Раковина — чашеобразное приспособление, которое используется для мытья рук или небольших предметов.

Изображение моделируемого объекта:

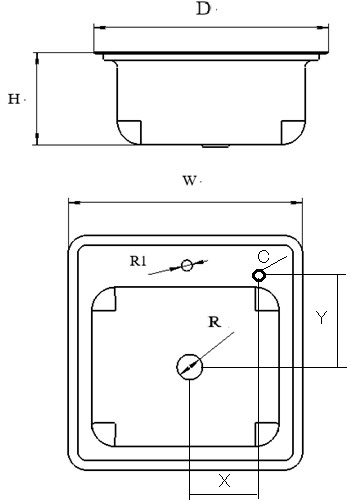


Рисунок 2.1 – Модель раковины

Измеряемые параметры для плагина:

* ***W*** – ширина раковины (мин – 450 мм, макс – 630 мм);
* ***D*** – длина раковины (мин – 450 мм, макс – 630 мм);
* ***H*** – глубина раковины (мин – 150 мм, макс – 210 мм);
* ***R*** – диаметр сливного отверстия (мин – 50 мм, макс – 70 мм);
* ***R1*** – диаметр отверстия под кран (мин – 20мм, макс – 30 мм);
* ***X*** – координата X отверстия под фильтр С;
* ***Y*** – координата Y отверстия под фильтр С;
* Глубина раковины ***H*** зависит от длины раковины ***D*** в соотношении D/3;
* Длина раковины **D** зависит от ширины раковины **W** в соотношении 1:1;

# 2.2 Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2022 с использованием .NET 6 [2], библиотеки для Kompas 3D [5].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [7] версии 3.13.3.

Для реализации пользовательского интерфейса использовалась система WindowsForms [8].

# 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием раковины различных размеров. Благодаря данному расширению, покупатели при покупке раковин могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 3 Обзор аналогов

Онлайн 3D - конфигуратор раковины и оборудования для ванной комнаты от компании KALDEWEI.

Конфигуратор продукции KALDEWEI позволяет моделировать ванны, эмалированные душевые поверхности, раковины и многое другое, с возможностью выбирать различные настройки, такие как цвет изделия, его изменяемые параметры (длина, ширина, высота), модель раковины, а также наличие отверстия под кран разного диаметра.

Интерфейс программы показан на рисунке 3.1.

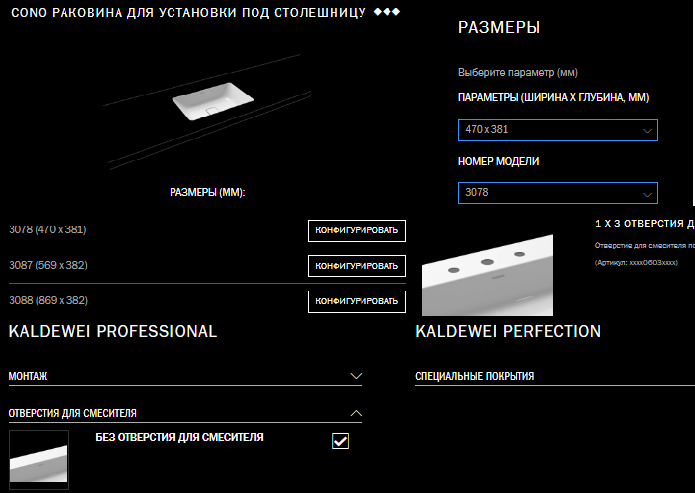


Рисунок 3.1 –Интерфейс программы

# 4 Описание реализации

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML.

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML – моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот. [9]

При использовании UML были простроена диаграмма классов.

# 4.1 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. [9]

На рисунке 4.1 представлена диаграмма классов.

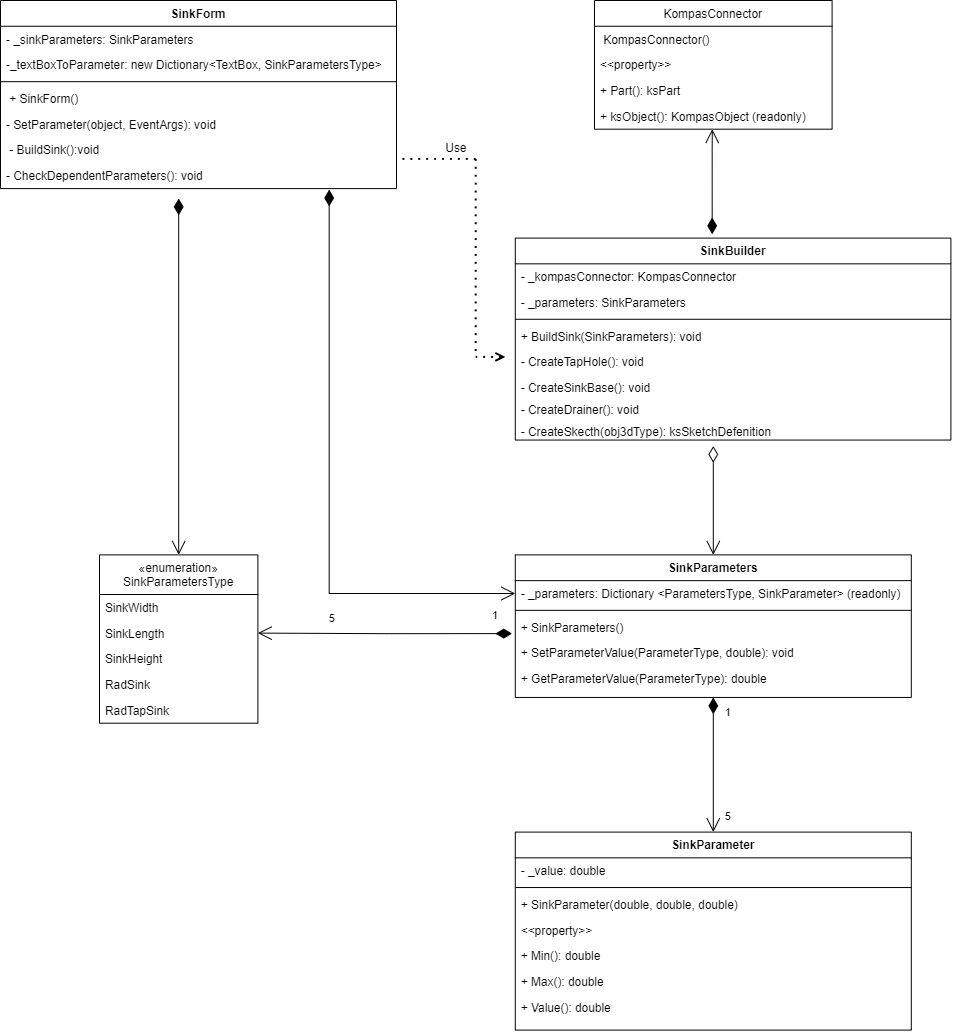


Рисунок 4.1 – Изначальная диаграмма классов

При начальном проектировании системы были созданы классы, представленные на рисунке 4.1, в процессе проектирования были изменены существующие классы, а также добавлены новые.

В итоговом проекте созданы следующие классы и методы, которые отображены на итоговой диаграмме классов (рисунок 4.2).

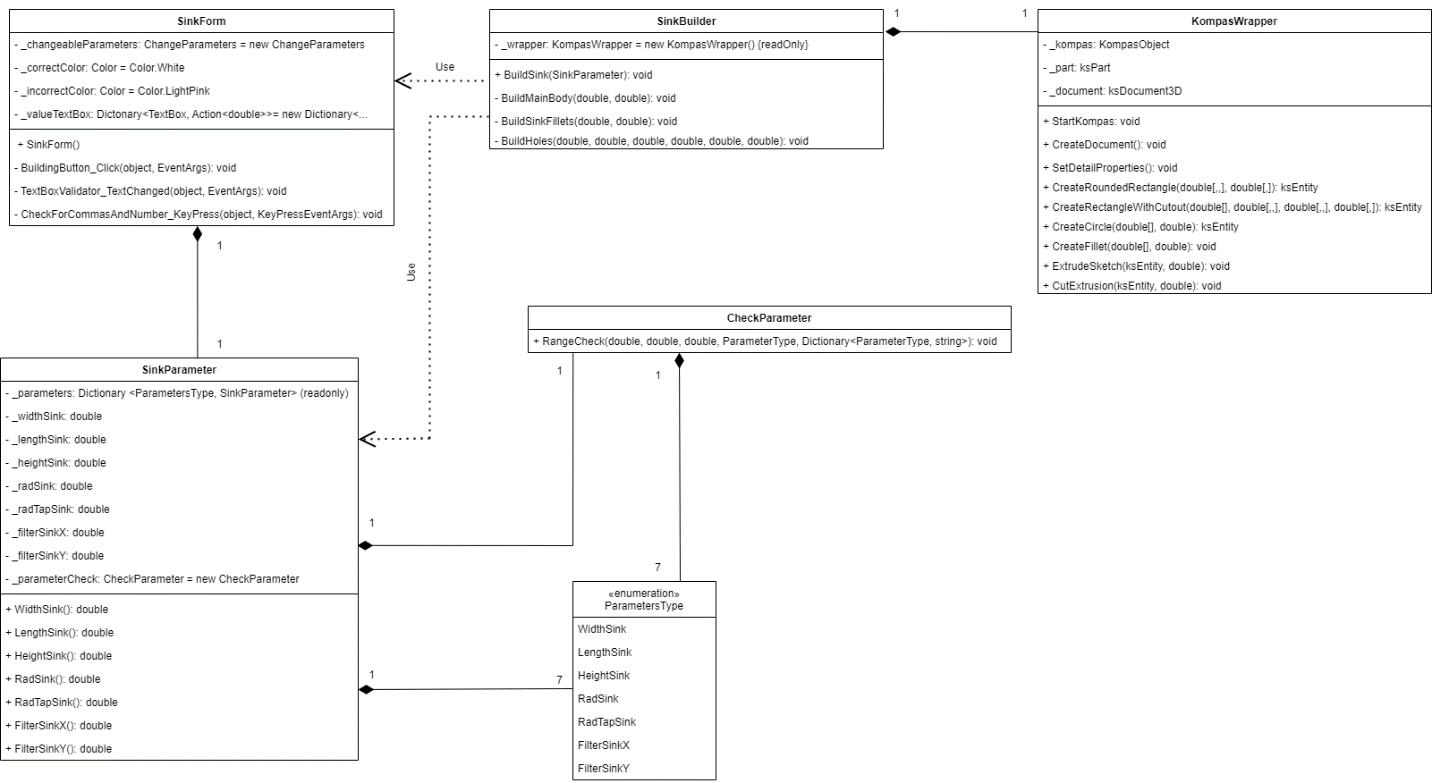


Рисунок 4.2 – Итоговая диаграмма классов

Был добавлен класс KompasWrapper для построения детали «раковины».

В классе SinkParameter появились 2 новых метода \_filterSinkX и \_filterSinkY, которые необходимы для выполнения дополнительного задания.

Также был создан класс CheckParameters который проверяет входит ли введенное значение в ряд допустимых.

Далее в таблицах 3.1 – 3.5 представлено описание классов.

Таблица 3.1 – Описание полей, методов, сущностей класса «KompasWrapper».

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_kompas: KompasObject |  | Объект компас API |
| -\_part: ksPart |  | Деталь |
| -\_document: ksDocument3D |  | Документ-модель |
| + StartKompas() | void | Запуск компас-3D |
| + CreateDocument() | void | Создание документа в компас-3D |
| + SetDetailProperties() | void | Установка свойств детали |
| +CreateRoundedRectangle() | ksEntity | Создание скругленного прямоугольника |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| +CreateRectangleWithCutout() | ksEntity | Создание прямоугольника с вырезом |
| +CreateCircle | ksEntity | Создание окружности |
| +CreateFillet | void | Создание скруглений плоскости |
| +ExtrudeSketch | void | Выдавливание эскиза на определенное расстояние |
| +CutExtrusion | void | Вырезание выдавливанием по эскизу |

Таблица 3.2 – Описание полей, методов, сущностей класса «SinkBuilder».

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| * \_wrapper: KompasWrapper |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| * BuildMainBody() | void | Хранит данные о каждом параметре модели |
| +SinkBuild(SinkParameter) | void | Построение раковины по заданным параметрам |
| * BuildSinkFillets() | void | Построение скруглений раковины |
| * BuildHoles() | void | Построение отверстий раковины |

Таблица 3.3 – Описание полей, методов, сущностей класса «SinkParameter».

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_widthSink | double | Параметр ширины раковины |
| \_lengthSink | double | Параметр длины раковины |
| \_heightSink | double | Параметр глубины раковины |
| \_radSink | double | Параметр сливного отверстия раковины |
| \_radTapSink | double | Параметр отверстия под кран раковины |
| \_filterSinkX | double | Координата X отверстия под фильтр |
| \_filterSinkY | double | Координата Y отверстия под фильтр |
| Dictionary<ParameterType, Action<string>>: Parameters |  | Словарь перечисления параметров и ошибки |

Продолжение таблицы 3.3

|  |  |  |
| --- | --- | --- |
| CheckParameter: \_parameterCheck |  | Экземпляр класса CheckParameter |
| WidthSink | double | Возвращает и устанавливает значение ширины раковины |
| LengthSink | double | Возвращает и устанавливает значение длины раковины |
| HeightSink | double | Возвращает и устанавливает значение глубины раковины |
| RadSink | double | Возвращает и устанавливает значение сливного отверстия раковины |
| RadTapSink | double | Возвращает и устанавливает значение отверстия под кран раковины |
| FilterSinkX | double | Возвращает и устанавливает значение координаты X отверстия под фильтр раковины |
| FilterSinkY | double | Возвращает и устанавливает значение координаты Y отверстия под фильтр раковины |

Таблица 3.4 – Описание полей, методов, сущностей класса «CheckParameter».

|  |  |  |
| --- | --- | --- |
| Название метода/поля |  | Описание |
| + RangeCheck(double, double, double, SinkParametersType, Dictionary<SinkParametersType, string>) |  | Проверка диапазона входных значений параметров |

Таблица 3.5 – Описание полей, методов, сущностей класса «SinkForm».

|  |  |
| --- | --- |
| Название метода/поля | Описание |
| SinkParameters: \_changeableParameter | Экземпляр класса SinkParameter. |
| Color: \_correctColor | Цвет корректно заполненного поля. |
| Color: \_incorrectColor | Цвет не корректно заполненного поля. |
| Dictionary<TextBox, Action<double>>: \_valueTextBox | Словарь, связывающий параметр раковины и его текстбокс. |

Продолжение таблицы 3.5

|  |  |
| --- | --- |
| TextBoxValidator\_TextChanged(object, EventArgs) | Валидация текстбоксов. |
| CheckForCommasAndNumbers\_KeyPress(object, KeyPressEventArgs) | Проверка содержания текстбоксом только одной запятой и цифры. |
| BuildingButton\_Click(object, EventArgs) | Обработка нажатия на кнопку Build button. |

# 5 Описание программы для пользователя

Макет пользовательского интерфейса представляет собой форму для ввода параметров раковины. Построение модели осуществляется путем нажатия на кнопку «Построить».

На рисунке 5.1 представлен макет пользовательского интерфейса.

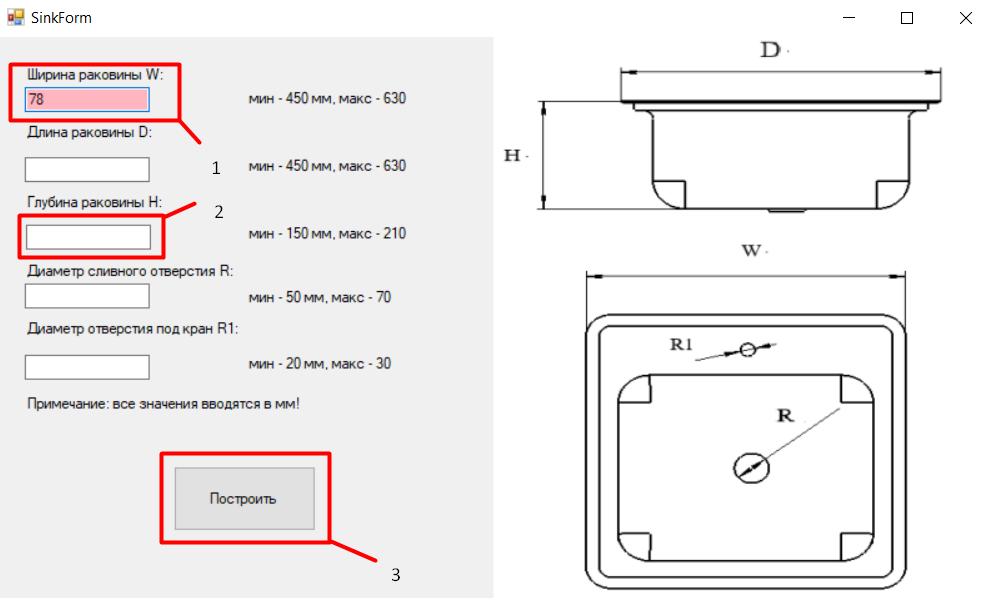


Рисунок 5.1 – Макет пользовательского интерфейса

Описание пользовательского интерфейса:

1 – ошибка при введении значений, не входящих в диапазон;

2 – поле для ввода параметров раковины в мм;

3 – кнопка для построения модели раковины в Компас-3D.

В ходе разработки были произведены некоторые изменения в пользовательском интерфейс. Итоговый макет пользовательского интерфейса представлен на рисунке 5.2.

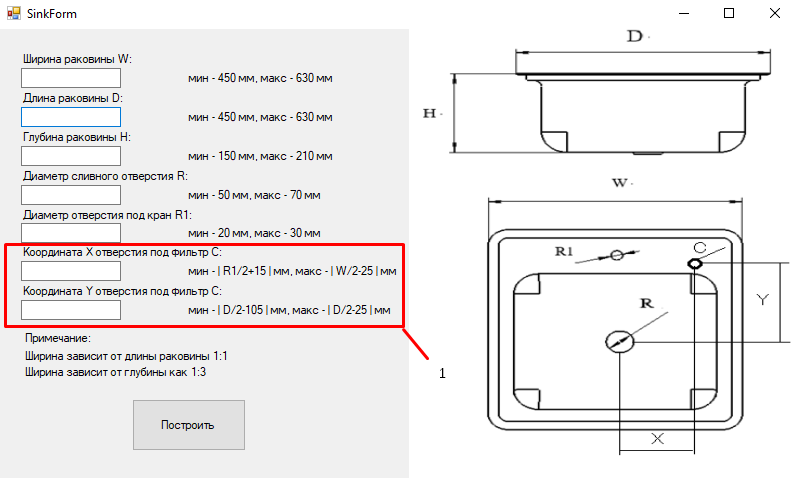


Рисунок 5.2 – Итоговый макет пользовательского интерфейса.

Описание нововведений в пользовательском интерфейсе:

1 – новые два поля ввода координат X и Y для построения отверстия под фильтр С.

После ввода необходимых параметров, построить деталь в САПР Kompas 3D можно с помощью кнопки «Построить». Раковина, построенная по заданным параметрам в САПР Kompas 3D, представлена на рисунке 5.3.

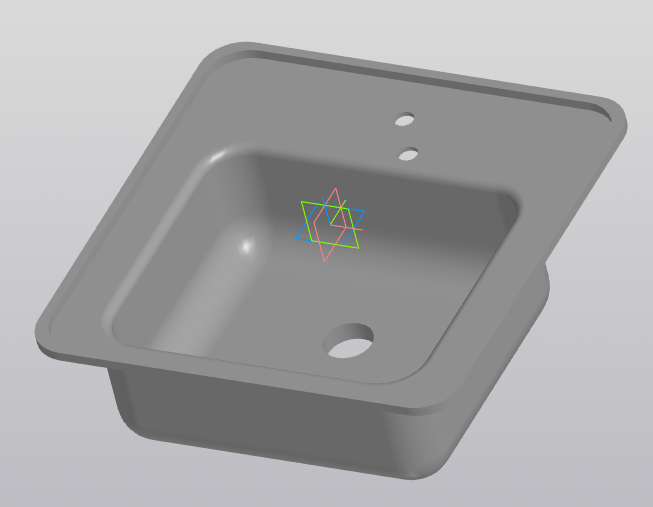


Рисунок 5.3 — Раковина, построенная по заданным параметрам в САПР Kompas 3D.

# 6 Тестирование программы

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

# 6.1 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Раковина», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [10]

Проведено тестирование максимальных, средних и минимальных параметров модели.

На рисунке 6.1 представлена проверки размеров модели с минимальным введенными параметрами в САПР Kompas 3D.

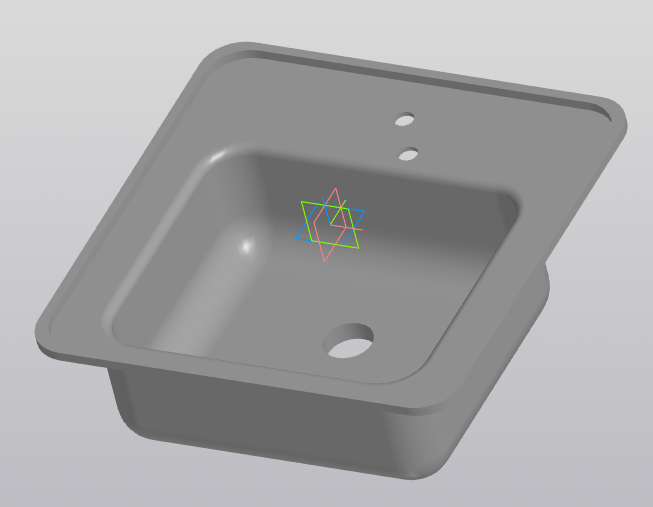


Рисунок 6.1 – Модель с минимальными введенными параметрами в Kompas 3D

Ниже на рисунках 6.2 и 6.3 представлены проверки размеров модели со средними и максимальными введенными параметрами в САПР Kompas 3D.

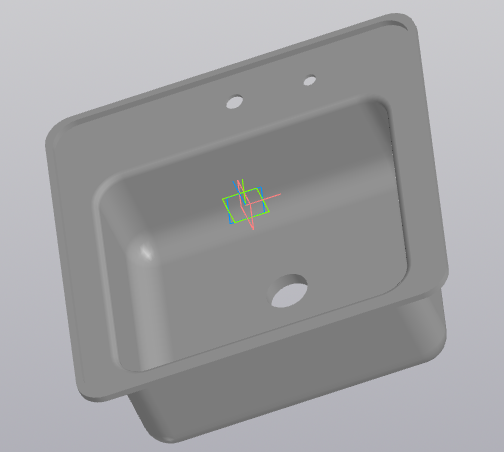


Рисунок 6.2 — Модель со средними параметрами в Kompas 3D

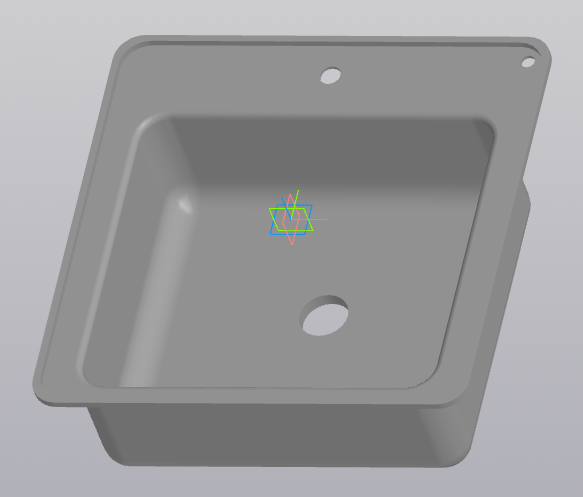


Рисунок 6.3 — Модель с максимальными параметрами в Kompas 3D

# 6.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование [11], проверялись открытые поля и методы. На рисунке 6.4 представлено тестирование класса проекта: Model. Степень покрытия проекта — сто процентов.

  
Рисунок 6.4 – Тестирование класса

Перечень тестов для Model, а также и их описание представлено в таблицах 6.1 - 6.14.

Таблица 6.1 – Тестовые случаи метода Test\_WidthSink\_GetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| WidthSink | 450 | Проверка корректного присваивания значения свойства WidthSink |
| LengthSink | 450 | Проверка корректного присваивания значения свойства LengthSink |

Таблица 6.2 – Тестовые случаи метода Test\_WidthSink\_UnCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| WidthSink | 430 | Проверка некорректного присваивания значения свойства WidthSink |
| WidthSink | 650 | Проверка некорректного присваивания значения свойства WidthSink |

Таблица 6.3 – Тестовые случаи метода Test\_LengthSink\_GetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| LengthSink | 470 | Проверка корректного получения значения свойства LengthSink |

Таблица 6.4 – Тестовые случаи метода Test\_LengthSink\_SetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| LengthSink | 450 | Проверка корректного присваивания значения свойства LengthSink |
| WidthSink | 450 | Проверка корректного присваивания значения свойства WidthSink |

Таблица 6.5 – Тестовые случаи метода Test\_LengthSink\_UnCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| LengthSink | 420 | Проверка некорректного присваивания значения свойства LengthSink |
| LengthSink | 670 | Проверка некорректного присваивания значения свойства LengthSink |

Таблица 6.6 – Тестовые случаи метода Test\_HeightSink\_GetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| LengthSink | 450 | Проверка корректного получения значения свойства LengthSink |
| HeightSink | 150 | Проверка корректного получения значения свойства HeightSink |

Таблица 6.7 – Тестовые случаи метода Test\_HeightSink\_SetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| LengthSink | 450 | Проверка корректного присваивания значения свойства LengthSink |
| HeightSink | 150 | Проверка корректного присваивания значения свойства HeightSink |

Таблица 6.8 – Тестовые случаи метода Test\_HeightSink\_UnCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| HeightSink | 216 | Проверка некорректного присваивания значения свойства HeightSink |
| HeightSink | 149 | Проверка некорректного присваивания значения свойства HeightSink |

Таблица 6.9 – Тестовые случаи метода Test\_RadSink\_GetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadSink | 55 | Проверка корректного получения значения свойства RadSink |

Таблица 6.10 – Тестовые случаи метода Test\_RadSink\_SetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadSink | 65 | Проверка корректного присваивания значения свойства RadSink |

Таблица 6.11 – Тестовые случаи метода Test\_RadSink\_UnCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadSink | 40 | Проверка некорректного присваивания значения свойства RadSink |

Продолжение таблицы 6.11

|  |  |  |
| --- | --- | --- |
| RadSink | 90 | Проверка некорректного присваивания значения свойства RadSink |

Таблица 6.12 – Тестовые случаи методы Test\_RadTapSink\_GetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadTapSink | 22 | Проверка корректного получения значения свойства RadTapSink |

Таблица 6.13 – Тестовые случаи метода Test\_RadTapSink\_SetCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadTapSink | 28 | Проверка корректного присваивания значения свойства RadTapSink |

Таблица 6.14 – Тестовые случаи метода Test\_RadTapSink\_UnCorrectValue

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметр | Описание |
| RadTapSink | 18 | Проверка некорректного присваивания значения свойства RadTapSink |
| RadTapSink | 35 | Проверка некорректного присваивания значения свойства RadTapSink |

# 

# 6.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [12]. Тестирование производилось на ПК со следующей конфигурацией:

* ЦП Ryzen 3 3.1ГГц;
* 8 ГБ ОЗУ;
* графический процессор объемом памяти 2 ГБ.

На графике, изображенном на рисунке 6.6 по горизонтали отмечено - количество построенных деталей, по вертикали - количество потребляемой оперативной памяти. На графике, изображенном на рисунке 6.7, по вертикали отмечено – время в секундах, по горизонтали – количество построенных деталей.

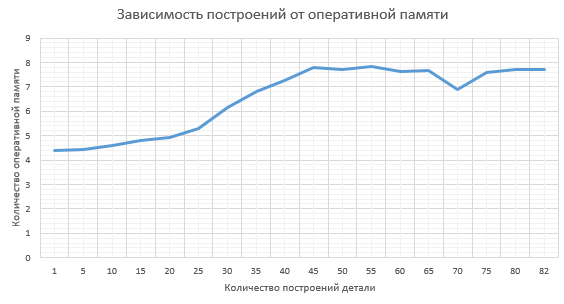


Рисунок 6.6 – График зависимости памяти от количества деталей

Из графика видно, что зависимость является линейной пока оперативная память на загружена практически полностью. На графике имеются падения, то есть происходит уменьшение нагрузки на оперативную память. Это обусловлено устройством оперативной памяти, что для её регенерации периодически приостанавливается обращение, это снижает среднюю скорость обмена и понижает нагрузку [12].

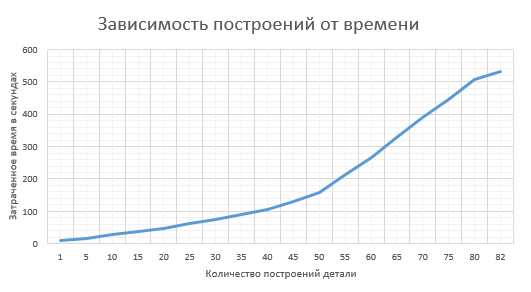


Рисунок 6.7 – График зависимости времени от количества построенных деталей.

Из графика видно, что зависимость количества необходимого времени от числа деталей является линейной с изменением угла наклона в течение теста. Это связано с тем, что при большой загрузке оперативной памяти и центрального процессора падает скорость построения одной детали. Поэтому увеличивается коэффициент наклона прямой.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API, функциональное и нагрузочное тестирование и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D модели «Раковина» в Kompas 3D, и проведено функциональное и нагрузочное тестирование плагина.

# Список использованных источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 09.12.2022).
2. Visual Studio [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 24.12.2022).
3. Раковина — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Раковина (дата обращения 20.10.2022).
4. KALDEWEI. [Электронный ресурс]. – Режим доступа: https://www.kaldewei.ru/ (дата обращения 20.10.2022).
5. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 20.10.2022).
6. КОМПАС-3D для разработчиков [Электронный ресурс]. – URL: https://kompas.ru/solutions/developers/ (дата обращения: 20.10.2022).
7. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 11.12.2022).
8. Что такое Windows Presentation Foundation (WPF) [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022 (дата обращения: 11.12.2022).
9. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 11.12.2022).
10. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 11.12.2022).
11. Юнит-тестирование для чайников [Электронный ресурс]. – URL: https://habr.com/ru/post/169381/ (дата обращения: 11.12.2022).
12. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: https://habr.com/ru/company/jugru/blog/329174/ (дата обращения: 11.12.2022).