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 «КРУЖКА» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 581

\_\_\_\_\_\_\_\_\_ Зафитумбу А.

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

Проверил:

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024

**РЕФЕРАТ**

Пояснительная записка, 42 страницы, 28 рисунков, 13 таблиц, 12 источников.

Ключевые слова: САПР, КОМПАС-3D, Плагин для САПР, Плагин отвёртки, C#, Windows Forms.

Объектом исследования являются технологии разработки плагинов для САПР.

Предметом исследования является применение технологий разработки плагинов, для автоматизации построения отвёрток разных размеров и параметров в САПР КОМПАС-3D.

Цель работы: создание программы для автоматизации построения отвёртки в САПР КОМПАС-3D.

Для создания использовались Microsoft Visual Studio 2022 (Windows Forms), .NET Framework 4.7.2, NUnit 3.14.0, NUnit3TestAdapter 3.17.0, StyleCop.Analyzers 1.1.118, StyleCop.Analyzers.Unstable 1.2.0.556, ReSharper, Fine Code Coverage, GitHub.

В результате работы было создано приложение Windows Forms, взаимодействующее с САПР КОМПАС-3D.

Областью применения являются предприятия, связанные с моделированием кружки.

**СОДЕРЖАНИЕ**

[1 ВВЕДЕНИЕ 4](#_Toc23763)

[2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 5](#_Toc22434)

[3 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 7](#_Toc31758)

[4 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 9](#_Toc363)

[5 НАЗНАЧЕНИЕ ПЛАГИНА 10](#_Toc17893)

[6 ОБЗОР АНАЛОГОВ 10](#_Toc13689)

[7 ОПИСАНИЕ РЕАЛИЗАЦИИ 13](#_Toc16292)

[8 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ 22](#_Toc9887)

[9 ТЕСТИРОВАНИЕ ПЛАГИНА 27](#_Toc28841)

[9.1 Функциональное тестирование 27](#_Toc10064)

[9.2 Модульное тестирование 32](#_Toc20472)

[9.3 Нагрузочное тестирование 37](#_Toc3928)

[10 ЗАКЛЮЧЕНИЕ 40](#_Toc1958)

[11 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 41](#_Toc5152)

**1 ВВЕДЕНИЕ**

САПР – организационно-техническая система, входящая в структуру проектной организации и осуществляющая проектирование при помощи комплекса средств автоматизированного проектирования. [1]

API (Application Programming Interface) — набор правил и протоколов, с помощью которых различные программные приложения могут взаимодействовать друг с другом и обмениваться данными, повышая тем самым функциональность и эффективность работы. [2]

Для разработки плагина для САПР прежде всего необходимо выбрать объект проектирования, подходящую для выбранного объекта САПР, средства разработки плагина (язык программирования и дополнительные средства разработки, выбор может быть основан на наличии API для выбранной САПР на конкретном языке).

Плагин автоматизации построения кружки необходим и может быть использован на предприятиях, занимающихся моделированием кружки, поскольку он упростит процесс моделирования и снизит нагрузку на моделлеров.

**2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ**

Этапы проведения работ по разработке плагина «Кружка» для САПР «Компас 3D» приведены в таблице 2.1.

Таблица 2.1 − Этапы проведения работ по разработке плагина «Кружка» для САПР «Компас 3D».

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этап | Состав работ | Наименование документа | Обозначение | Разработано согласно | Сроки выполнения |
| 1 | Создание технического задания | Техническое задание | - | ГОСТ 34.602-2020 | Не позднее 8.10.2024 |
| 2 | Создание проекта системы | Проект системы | - | ОС ТУСУР 01-2021 | Не позднее 29.10.2024 |
| 3 | Реализация плагина | Программный код | - | RSDN Magazine #1-2004 | Не позднее 10.12.2024 |
| Документ с тремя вариантами дополнительной функциональности плагина для согласования |
| Модульные тесты |

Таблица 2.1 − Продолжение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этап | Состав работ | Наименование документа | Обозначение | Разработано согласно | Сроки выполнения |
| 4 | 1. Доработка плагина 2. Создание пояснительной записки | Программный код | - | 1. RSDN Magazine #1-2004 2. ОС ТУСУР 01-2021 | Не позднее 31.12.2024 |
| Модульные тесты |
| Пояснительная записка |

Основной проблемой на всех этапах проекта оказалась связь между вторым и третьим этапами. Это связано с недостаточными знаниями в области проектирования систем и допущением множества неточностей. В результате на этапе реализации плагина пришлось вносить изменения, что повлияло на окончательную версию UML-диаграммы классов. Хотя я описал эту проблему, известно, что изменения в финальной версии диаграммы по сравнению с версией на этапе проектирования не являются редкостью. В реальных проектах такие изменения чаще всего вызваны потребностями заказчика, а не недостатком опыта разработчиков в создании диаграмм.

Кроме того, на этапе выбора предмета проектирования недостаточно изучались особенности API КОМПАС-3D. Хотя это не привело к серьезным трудностям в написании кода, данный факт увеличил временные затраты по сравнению с первоначальными планами. Вывод однозначен: необходимо более тщательно изучать выбранные инструменты разработки, чтобы избежать ситуаций, связанных с непредусмотрительностью разработчиков.

# 3 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Кружка — это большой стакан с ручкой, предназначенный для напитков. Она может иметь различные формы и размеры, иногда оснащается крышкой. В русском языке термин "кружка" также обозначает меру объема. Кружки часто используются для чая, кофе и других горячих или холодных напитков.[5]



Рисунок 3.1 − Модель кружки

***Изменяемые параметры для предмета проектирования*** (также все обозначения показаны на рисунке 3.1):

* Диаметр кружки D1 (100 мм – 150 мм);
* Диаметр полости кружки D2 (9/10 от диаметра D1);
* Диаметр основания полости кружки D3 (9/10 от диаметра D4);
* Диаметр основания кружки D4 (70 мм – 100 мм);
* Радиус R1 от кривизны кружки до верха кружки (300 мм – 350 мм);
* Радиус R2 кривизны кружки от основания (5/8 от радиуса R1);
* Радиус R3 1-й кривизны запястья кружки (10 мм – 20 мм);
* Радиус R4 2-й кривизны запястья кружки (Два раза больше чем радиус R3: R4 = 3\*R3);
* Радиус R5 3-й кривизны запястья кружки (75 мм – 85 мм);
* Кружка имеет высоту L (100 мм – 150 мм);
* Ручка кружки имеет длину *l* (3/4 от длины L).

**4 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ**

При создании плагина использовались следующие инструменты:

− WindowsForms и .NET Framework 4.7.2;

− GitHub;

− ReSharper;

− Fine Code Coverage;

− StyleCop.Analyzers 1.1.118;

− StyleCop.Analyzers.Unstable 1.2.0.556;

− NUnit 3.14.0;

− NUnit3TestAdapter 3.17.0.

Плагин был создан на технологии Windows Forms, поддерживающей широкий набор функций для разработки приложений, включая элементы управления, графику, привязку данных и ввод пользователя [4], а также .NET Framework 4.7.2, программной платформе основанной на сервероцентрической модели.

GitHub − платформа с возможностями хранения, распространения и совместной работы над написанием кода. Git − система управления версиями, которая интеллектуально отслеживает изменения в файлах. [5]

ReSharper − расширение для Microsoft Visual Studio, помогающее программировать эффективнее. Позволяет исследовать, улучшать, писать и обслуживать код. [6]

Fine Code Coverage − расширение для Microsoft Visual Studio, визуализирующий покрытие кода модульными тестами. [7]

StyleCop − средство для контроля кода, автоматически находящее синтаксические ошибки. [8]

NUnit − фреймфорк для модульного тестирования всех языков .Net. [9]

# 5 НАЗНАЧЕНИЕ ПЛАГИНА

Назначение разрабатываемого плагина заключается в быстром моделировании различных видов кружек. Благодаря этому расширению производители кружек смогут наглядно увидеть спроектированную модель и, при необходимости, адаптировать её под свои требования.

**6 ОБЗОР АНАЛОГОВ**

Первым аналогом является плагин «Mug Creator» для Blender, который позволяет интуитивно и эффективно моделировать кружки в 3D. Этот плагин доступен бесплатно, что делает его доступным для всех пользователей Blender, как новичков, так и опытных. Он предлагает такие функции, как создание индивидуальных форм кружек, добавление ручек и применение реалистичных текстур. Этот плагин выделяется своей простотой в использовании и удобным интерфейсом, позволяя пользователям сосредоточиться на креативности, не отвлекаясь на сложные инструменты. Кроме того, плагин включает в себя расширенные опции, такие как возможность добавления геометрических деталей и настройки параметров рендеринга для достижения фотореалистичного вида. Пользователи также могут воспользоваться встроенными учебниками, которые направляют их через процесс моделирования, что делает обучение более плавным. Этот плагин является прямым инструментом для тех, кто хочет быстро и точно создавать модели кружек в Blender. Интерфейс взаимодействия представлен на рисунке 6.1.

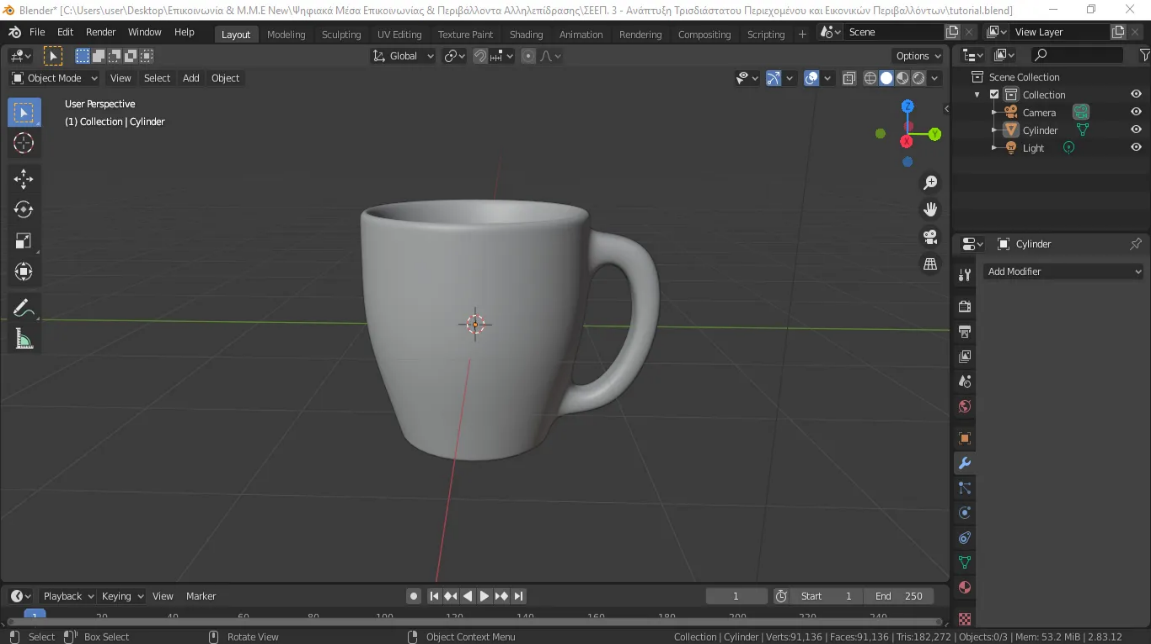


Рисунок 6.1 – Пользовательский интерфейс «Mug Creator» для Blender

Вторым аналогом является плагин «Coffee Cup» для SketchUp, который позволяет легко и быстро моделировать кружки в 3D. Этот плагин доступен бесплатно и подходит как для новичков, так и для опытных пользователей. Он предлагает интуитивно понятные инструменты, такие как «Перо», «Прямоугольник», «Дуга» и «Круг», что упрощает процесс создания формы кружки.

Пользователи могут начать с рисования половины 2D-кружки и затем использовать инструмент «Follow Me» для создания объемной модели. Плагин также предоставляет возможность добавления ручек с помощью простых геометрических форм, что делает процесс моделирования более гибким и доступным. Интерфейс взаимодействия представлен на рисунке 6.2.



Рисунок 6.2 – пользовательский интерфейс ‘‘SketchUp’’

Третьим аналогом является плагин «Coffee Mug» для Fusion 360, который позволяет легко и быстро моделировать кружки в 3D. Этот плагин доступен в рамках подписки на Fusion 360, что делает его доступным для пользователей, уже использующих данное программное обеспечение. Он предлагает интуитивно понятные инструменты для создания формы кружки, включая команды «Revolve» и «Loft», что упрощает процесс моделирования.

Процесс начинается с создания цилиндра, который служит основой для тела кружки. Затем с помощью команды «Shell» можно сделать внутреннюю часть пустой. Для ручки кружки используется команда «Spline» для рисования профиля, после чего применяется команда «Loft» для соединения ручки с телом кружки. Интерфейс взаимодействия представлен на рисунке 6.3.



Рисунок 6.3 – пользовательский интерфейс Fusion 360

# 7 ОПИСАНИЕ РЕАЛИЗАЦИИ

UML − это стандартный язык визуального моделирования, предназначенный для следующего использования:

− моделирование бизнеса и подобных процессов;

− анализ, проектирование и внедрения программных систем.

UML − это общий язык для бизнес-аналитиков, архитекторов и разработчиков программного обеспечения, используемый для описания, спецификации, проектирования и документирования существующих или новых бизнес-процессов, структуры и поведения артефактов программных систем.[12]

UML диаграмма классов после проектирования для плагина «Кружка» представлена на рисунке 7.1.

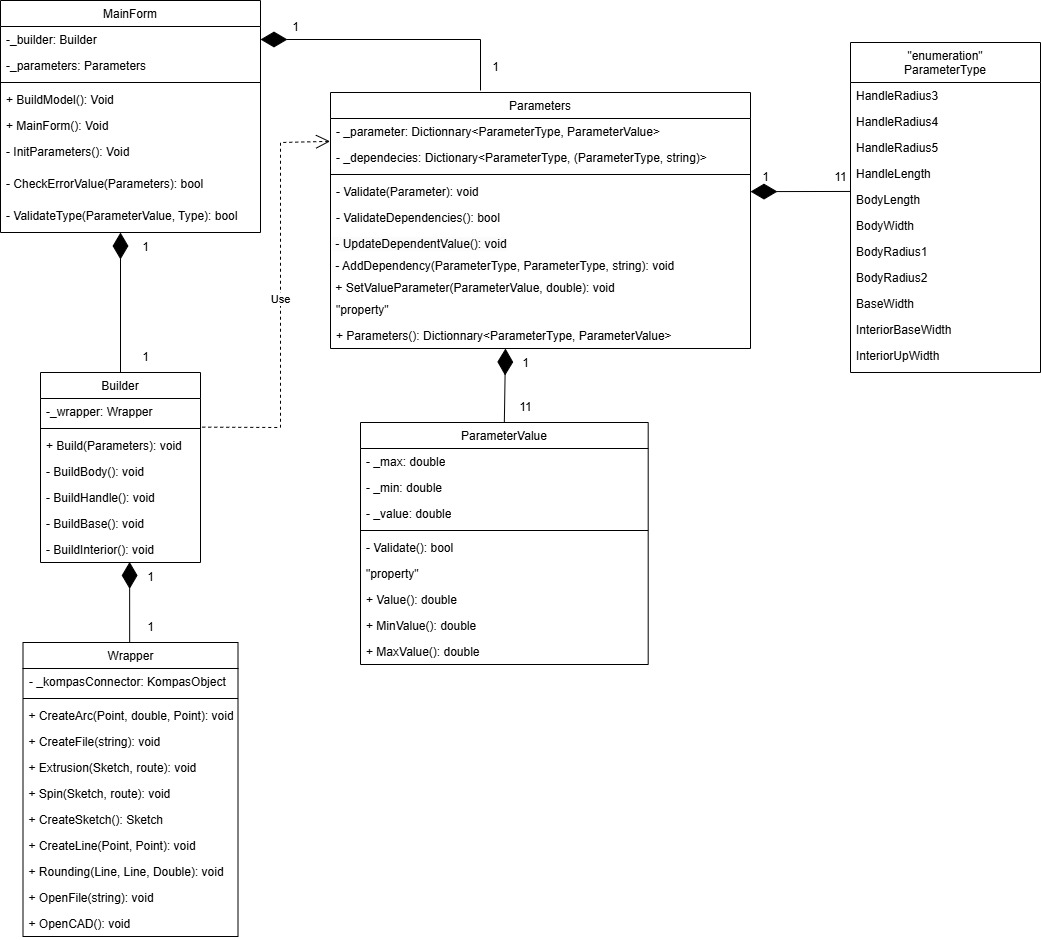


Рисунок 7.1 − UML диаграмма классов после проектирования для плагина «Кружка»

UML диаграмма классов после реализации плагина «Кружка» представлена на рисунке 7.2.

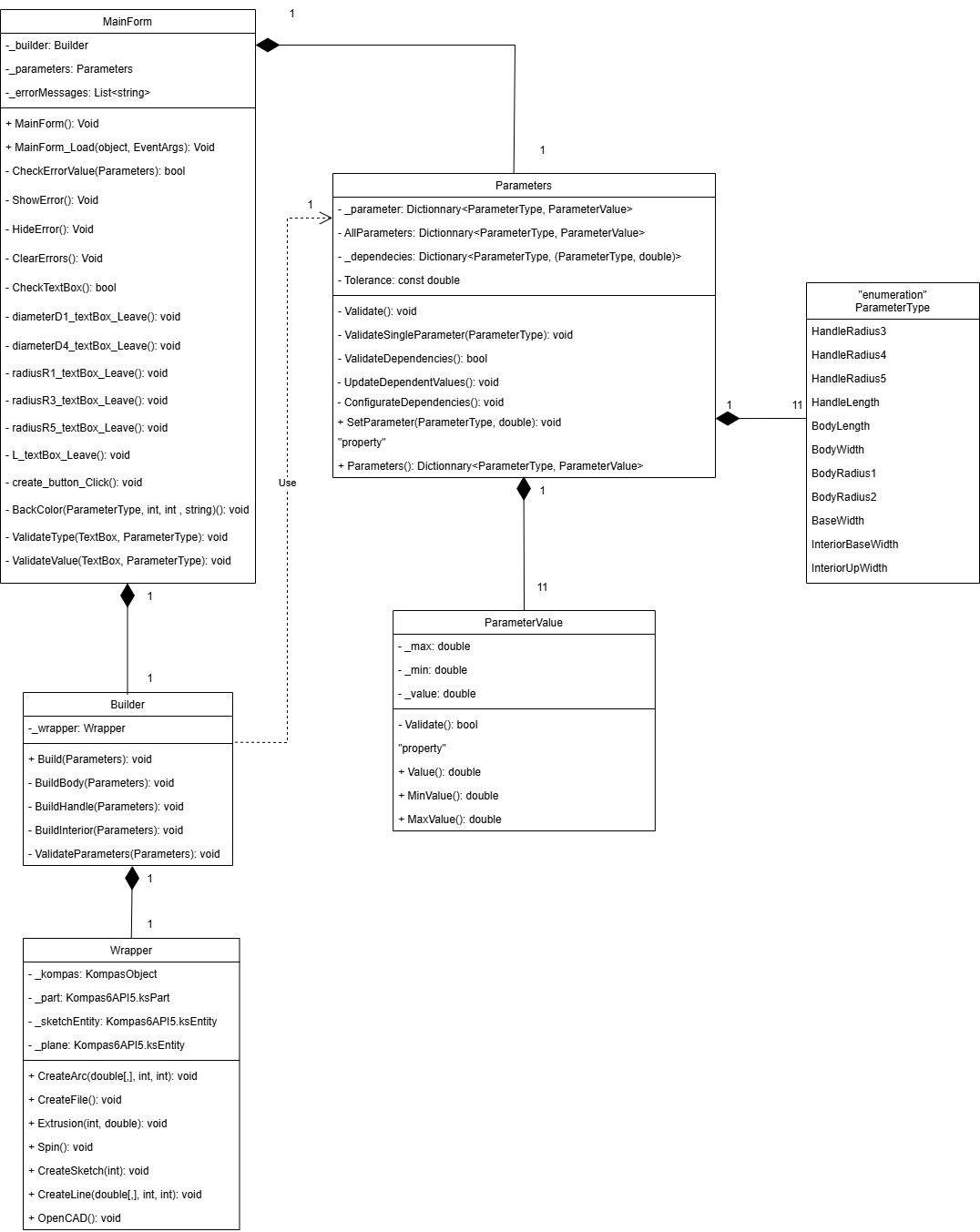


Рисунок 7.2 − UML диаграмма классов после реализации плагина «Кружка»

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

Таблица 7.1 − Свойства класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранит в себе объект построения |
| \_parameters | Parameters | Хранит в себе параметры для объекта построения |
| \_errorMessages | List<string> | Поле хранящее в себе сообщения ошибок |

Таблица 7.2 − Методы класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| MainForm | − | Конструктор MainForm |
| MainForm\_Load |  | Инициализация параметров при загрузке формы |
| diameterD1\_textBox\_TextLeave | − | Обработчик события покидания поля ввода для диаметра кружки (D1) |
| diameterD4\_textBox\_ TextLeave | − | Обработчик события покидания поля ввода для диаметра основания (D4) |
| radiusR1\_textBox\_ TextLeave | − | Обработчик события покидания поля ввода для радиуса тела (R1) |
| radiusR3\_textBox\_ TextLeave | − | Обработчик события покидания поля ввода для радиуса тела (R3) |

Таблица 7.2 − Окончание таблицы 7.2.

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| radiusR5\_textBox\_ TextLeave | − | Обработчик события покидания поля ввода для радиуса тела (R5) |
| L\_textBox\_Leave | − | Обработчик события покидания поля ввода для высоты кружки (L) |
| create\_button\_Click | − | Запуск построения модели по заданным параметрам |
| BackColors | TextBox, ParameterType, int, int, string | Устанавливает цвета для всех текст боксов по результатам проверки текст ошибки для установки правильного Тултипа |
| ValidateValue | TextBox, ParameterType | Вызов валидации параметров |
| ValidateType | TextBox, ParameterType | Проверка введённых данных по формату |

Таблица 7.3 − Свойства класса Builder

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

Таблица 7.4 − Методы класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| Build | Parameters | Построение модели по заданным параметрам |
| BuildBody | − | Построение стержня кружки |
| BuildHandle | − | Построение ручки кружки |
| BuildInterior | − | Построение внутренней части кружки |
| ValidateParameters | Parameters | Валидация параметров перед построением |

Таблица 7.5 − Свойства класса ParameterValue

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_max | double | Максимально допустимое значение параметра |
| \_min | double | Минимально допустимое значение параметра |
| \_value | double | Значение параметра |

Таблица 7.6 − Методы класса ParameterValue

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| MinValue | − | double | Минимально допустимое значение параметра |

Таблица 7.6 − Окончание таблицы 7.6.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Value | − | double | Значение параметра |
| MaxValue | − | double | Максимально допустимое значение параметра |
| Validate | − | bool | Проверка правильности все значения |

Таблица 7.7 − Свойства класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_kompas | KompasObject | Ссылка на обьект компаса |
| \_part | ksPart | Поле для хранения выбранной 3d детали |
| \_sketchEntity | ksEntity | Поле для хранения выбранного эскиза |
| \_plane | ksEntity | Поле для хранения выбранной плоскости |

Таблица 7.8 − Методы класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateArc | double[,], int, int | Создание дуги по двум точкам |
| CreateCircle | double, int, int | Рисует окружность |
| CreateLine | double[,], int, int | Создание линии |

Таблица 7.9 − Методы класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateSketch | int | Создание эскиза |
| Spin | − | Вращение эскиза |
| Extrusion | int, double | Выдавливание эскиза |
| OpenFile | − | Открытие файла |
| OpenCAD | − | Открытие Компас3D |

Таблица 7.10 − Свойства класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_parameters | Dictionary  <ParameterType, ParameterValue> | Хранит в себе словарь с параметрами объекта построения |
| \_dependencies | Dictionary  <ParameterType, (ParameterType, double)> | Хранит в себе словарь с параметрами, зависящими от объекта построения |
| Tolerance | const double | Поле для хранения допустимой погрешности |

Таблица 7.11 − Методы класса Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип данных | Описание |
| Parameters | − | Dictionary  <ParameterType, Parameter> | Хранит в себе словарь с параметрами объекта построения |

Таблица 7.11 − Окончание таблицы 7.11

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип данных | Описание |
| ValidateDependecies | − | bool | Выполнить валидацию зависимых параметров. |
| Parameters | − | Dictionnary<ParameterType, ParameterValue> | Конструктор класса |
| Validate | Parameter | − | Выполнить валидацию параметров. |
| ValidateSingleParameter | ParameterType | − | Метод для проверки отдельного параметра |
| SetParameters | ParameterType, double | − | Выполнить валидацию и задание каждого значения параметров. |
| UpdateDependentValue | − | − | Автоматически обновляет зависимые значения |
| ConfigureDependencies | ParameterType, ParameterType, double | − | Метод для настройки зависимостей между параметрами |

В отличии от диаграммы классов проекта системы диаграмма классов после реализации плагина имеет следующие отличия:

− MainForm: добавление переменной errorMessages, которая регистрирует все ошибки при вводе данных; добавление аргументов CheckErrorValues, ShowErrors, HideErrors, ClearErrors, CheckTextBox для управления отображением ошибок на интерфейсе; BuildModel заменили на creatButton\_Click; Validate, ValidateValue, SetColors получили входные параметры;

− Parameters: добавление двух новых переменных: AllParameters (словарь всех параметров) и Tolerance (для допустимых отклонений значений); AddDependency был изменен на ConfigureDependencies; SetValueParameter изменен на SetParameters; добавлена функция ValidateSingleParameter, которая валидирует каждый параметр по отдельности;

− Builder: BuildBase был удален и перенесен в BuildBody; добавлен аргумент ValidateParameters, который валидирует параметры перед их построением;

− Wrapper: были убраны ненужные поля \_document3D, \_document2D, \_sketchDef; было добавлено необходимое поле \_plane, а также переопределены типы у полей на ksPart у \_part и ksEntity у \_sketchEntity и \_plane; CreateLine, CreateArc, изменены входные параметры для избавления от повторений в коде;

**8 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ**

При запуске приложения открывается форма для заполнения параметров объекта (рисунок 8.1).

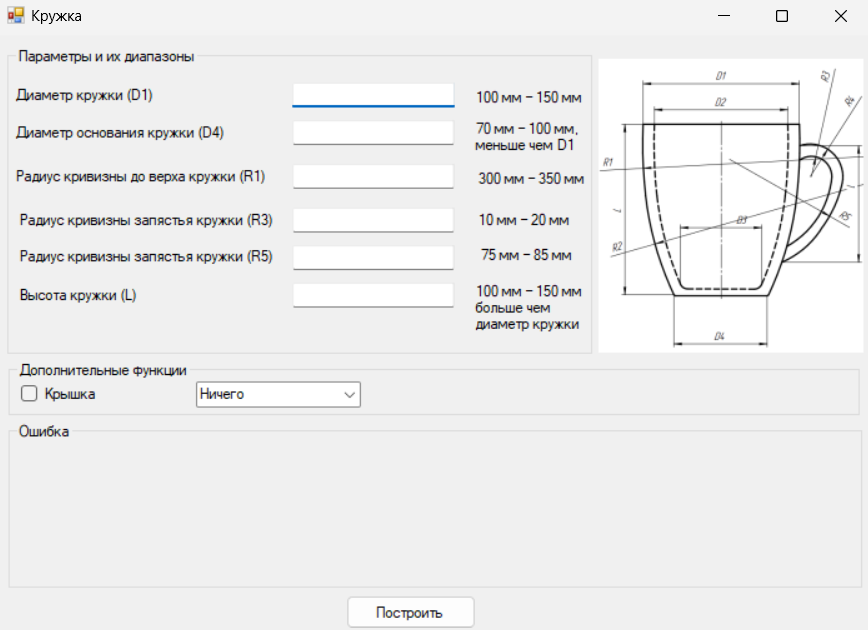


Рисунок 8.1 – Начальная форма в момент запуска приложения

При нажимании на кнопку с незаполненными или неверно заполненными полями будет показано MessageBox с ошибкой (см. рис. 8.2), при правильном же заполнении откроется КОМПАС-3D и начнётся построение модели по заданным параметрам.

На рисунке 8.2 мы можем посмотреть MessageBox который выдает когда нажимаем кнопку построения но все текстовые полы пусты.

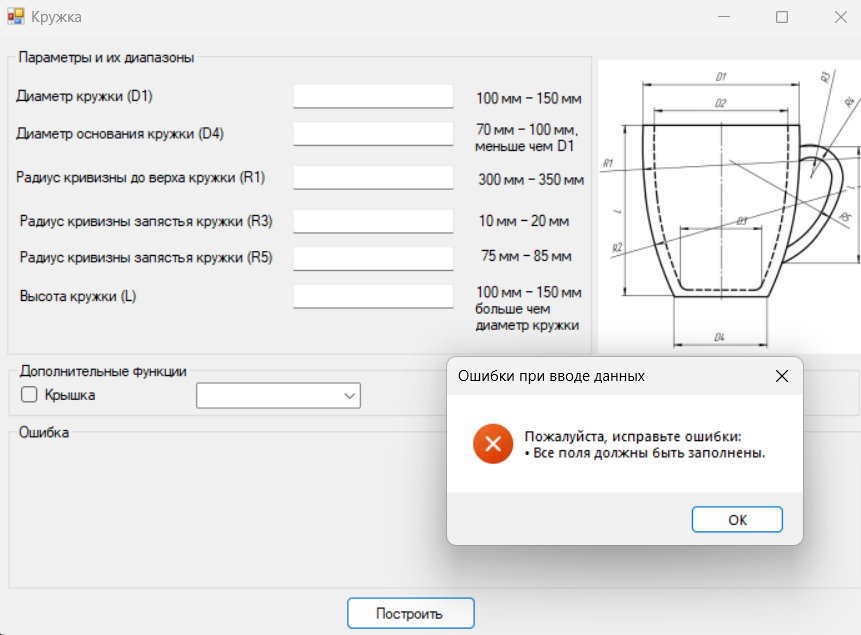


Рисунок 8.2 — Макет пользовательского интерфейса с неправильно введенным значением параметра

Создание дополнительных функций не является обязательным, мы можем посмотреть на рисунках 8.3, 8.4 и 8.5, сообщения, отправленные по интерфейсу, и какие дополнительные функции используются в новой модели.

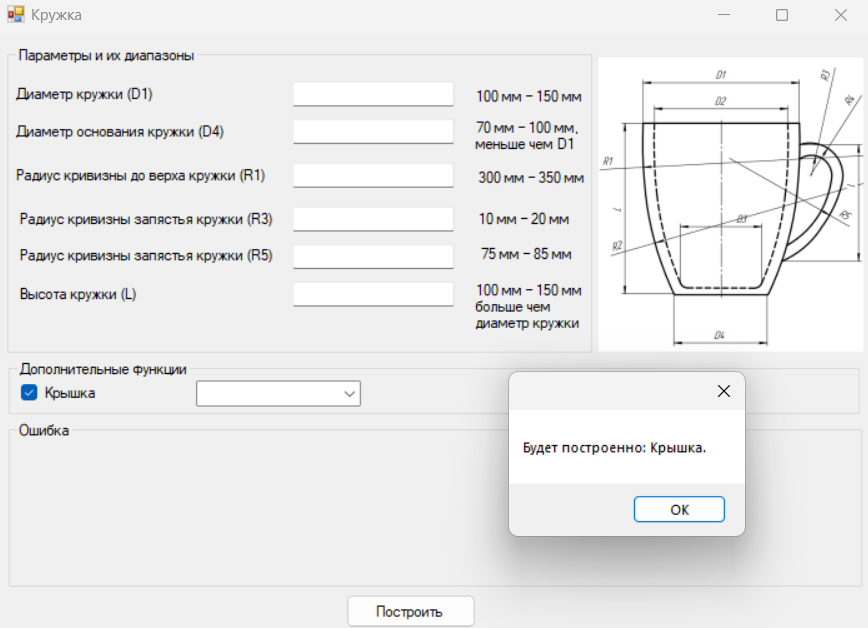


Рисунок 8.3 — Макет пользовательского интерфейса с сообщением при выборе строительства крышки

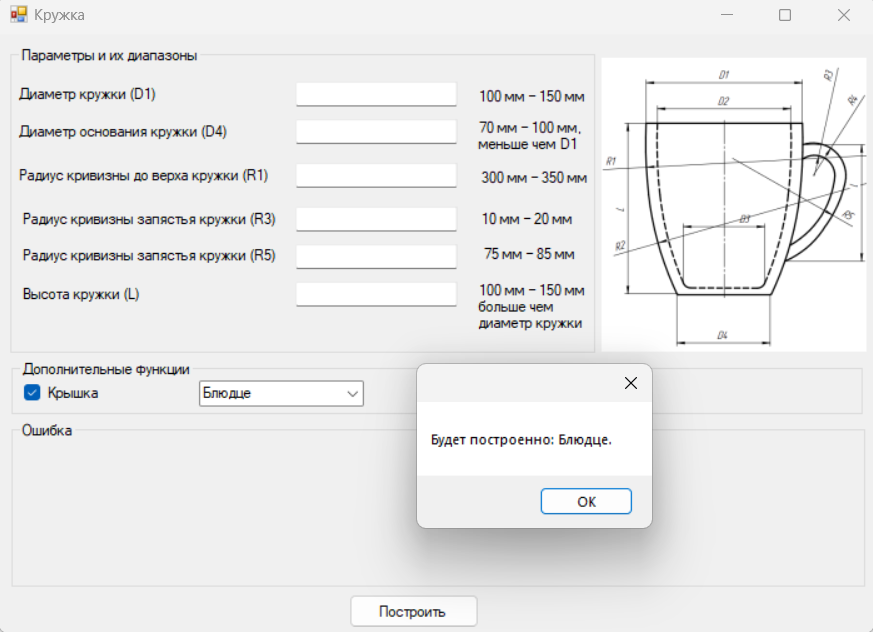


Рисунок 8.4 — Макет пользовательского интерфейса с сообщением при выборе строительства блюдце

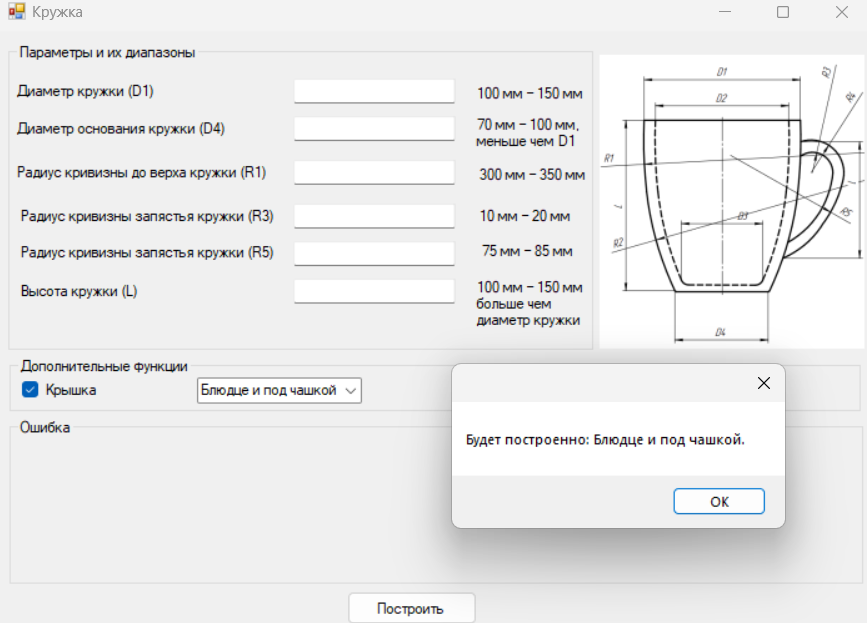


Рисунок 8.5 — Макет пользовательского интерфейса с сообщением при выборе строительства блюдце и под чашкой

Программа оснащена функциями проверки введенных пользователем чисел, в частности выводом на интерфейс минимальных и максимальных значений каждого параметра. На рисунке 8.6 мы можем посмотреть ошибку, когда мы вводим неправильное значение в один из параметров.

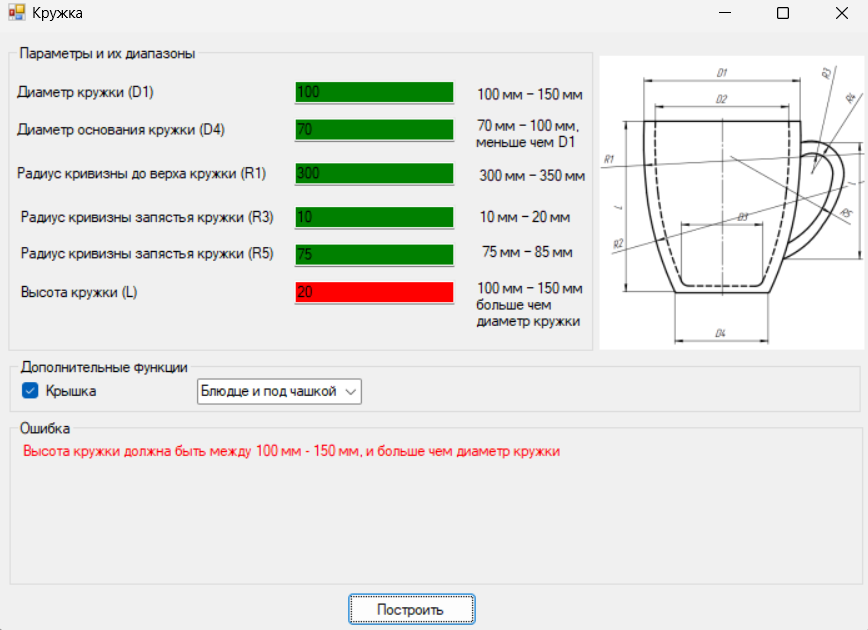


Рисунок 8.6 — Макет пользовательского интерфейса с неправильно введенным значением параметра

На рисунке 8.7 мы можем увидеть список ошибок, когда все значения неверны.

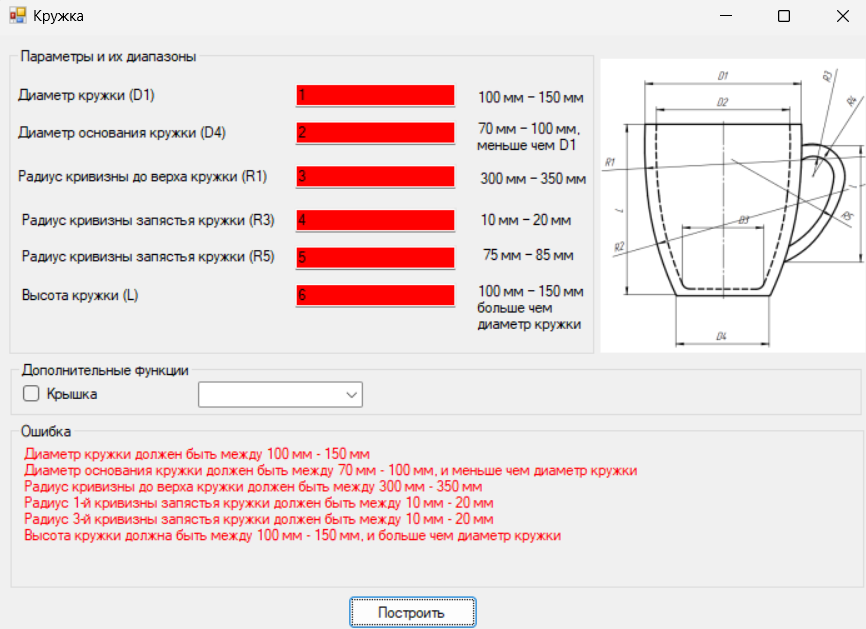


Рисунок 8.7 — Макет пользовательского интерфейса с списком ошибок

На рисунке 8.8 мы можем посмотреть окно сообщения, когда нажимаем кнопку «Построить», хотя ошибки все еще есть.

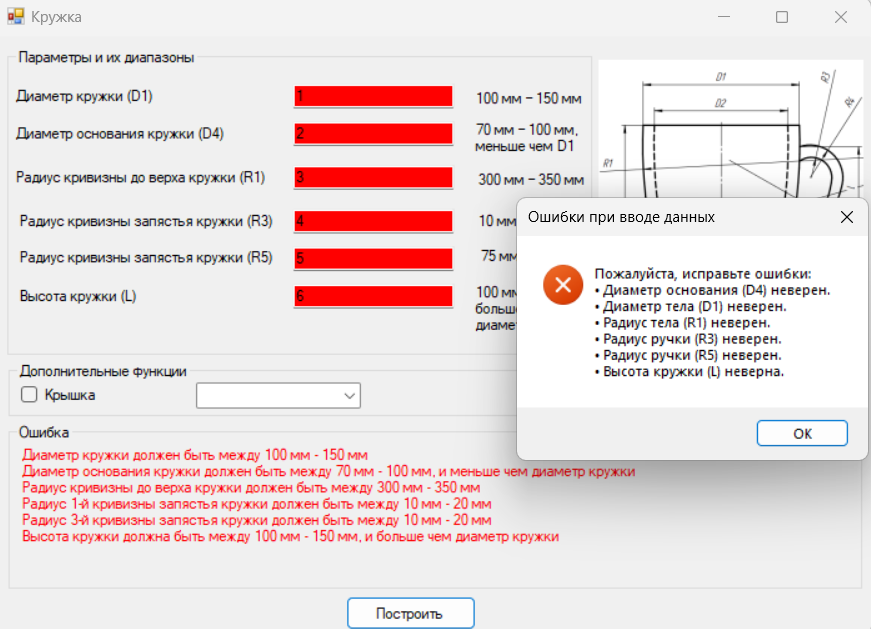


Рисунок 8.8 — Макет пользовательского интерфейса с неправильно введенными значениями параметров

**9 ТЕСТИРОВАНИЕ ПЛАГИНА**

## 9.1 Функциональное тестирование

Помимо проверок минимального и максимального значений, при вводе значений выполняются и другие проверки. В пользовательском интерфейсе (см. рисунок 8.1) в разделе «Параметры и их диапазоны» видно, что D4 должен быть меньше D1, а L должен быть больше D1, в противном случае будет выведена ошибка , как показано на рисунке 9.1. и мы не можем построить модель, как показано на рисунке 9.2.

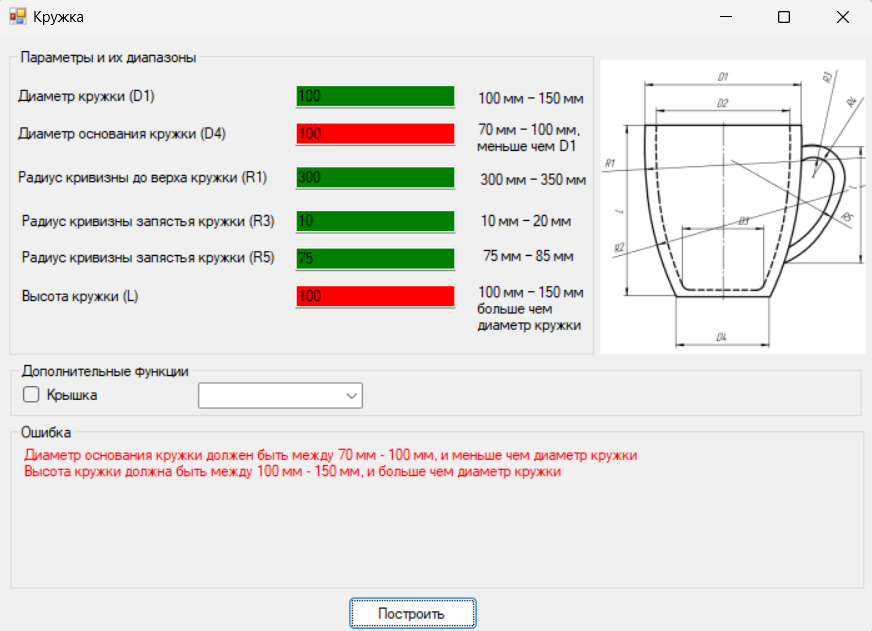


Рисунок 9.1 — Макет пользовательского интерфейса с ошибками ввода

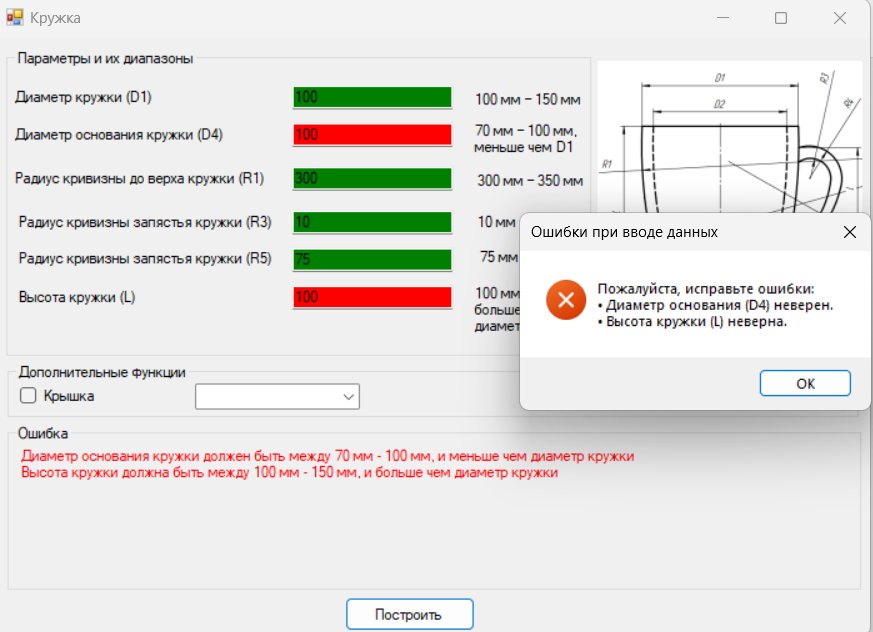


Рисунок 9.2 — Макет пользовательского интерфейса с сообщением при нажатии кнопки «Построить»

На рисунке 9.3 мы видим, когда все введенные пользователем значения верны. Таким образом, можно построить модель.

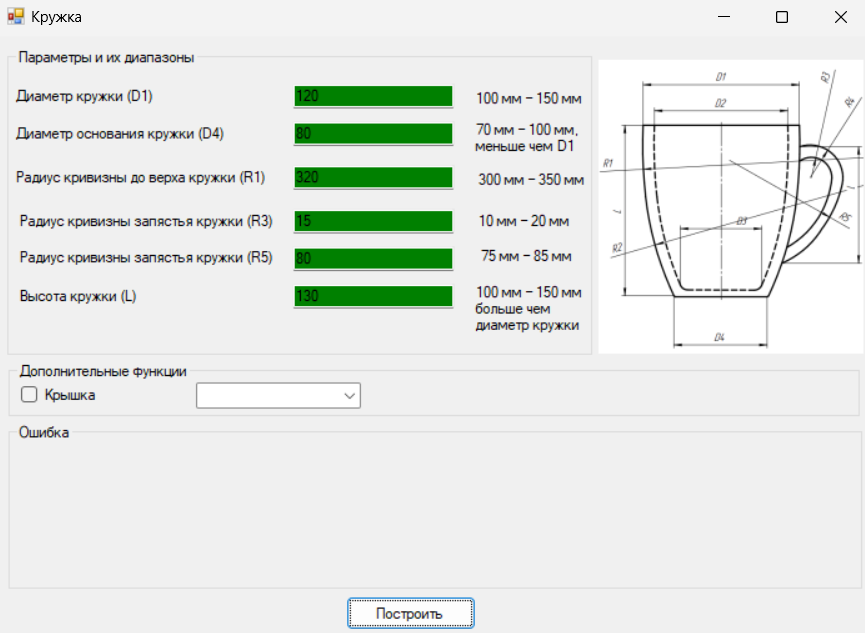


Рисунок 9.3 — Макет пользовательского интерфейса со всеми правильными значениями (стандартные параметры)

На рисунке 9.4 представлен результат построения модели с стандартными параметрами.

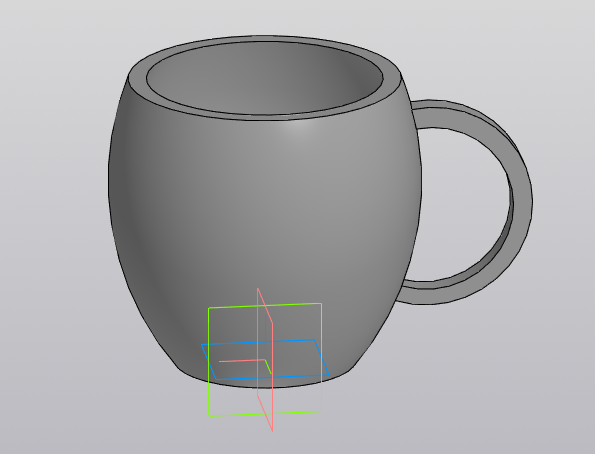


Рисунок 9.4 – Модель по стандартным параметрам

На рисунке 9.5 представлено заполнение формы минимальными параметрами.

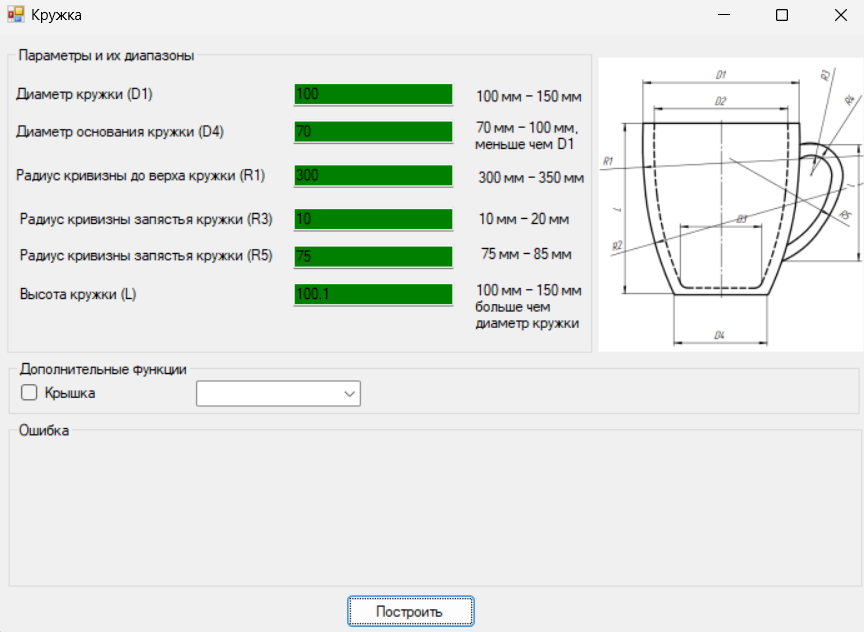


Рисунок 9.5 – Минимальные параметры

На рисунке 9.6 представлен результат построения модели с минимальными параметрами.

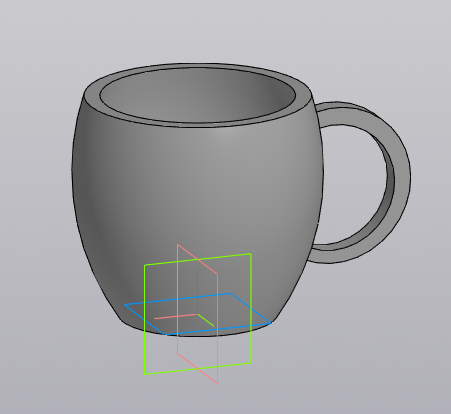


Рисунок 9.6 – Модель по минимальным параметрам

На рисунке 9.7 представлено заполнение формы максимальными параметрами.

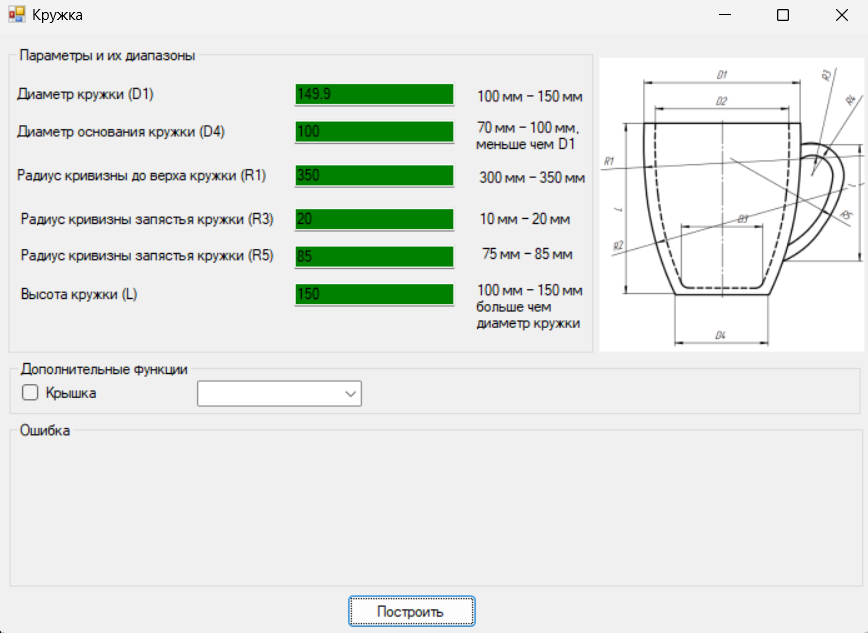


Рисунок 9.7 – Максимальные параметры

На рисунке 9.8 представлен результат построения модели с максимальными параметрами.

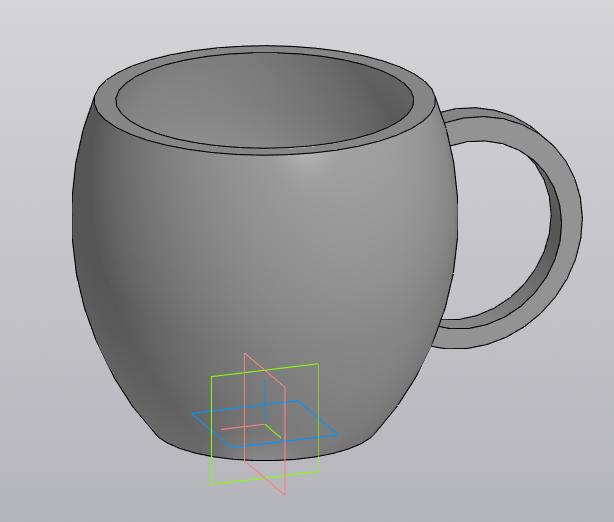


Рисунок 9.8 – Модель по максимальным параметрам

На рисунке 9.9 показано заполнение формы со стандартными параметрами и дополнительными функциями.

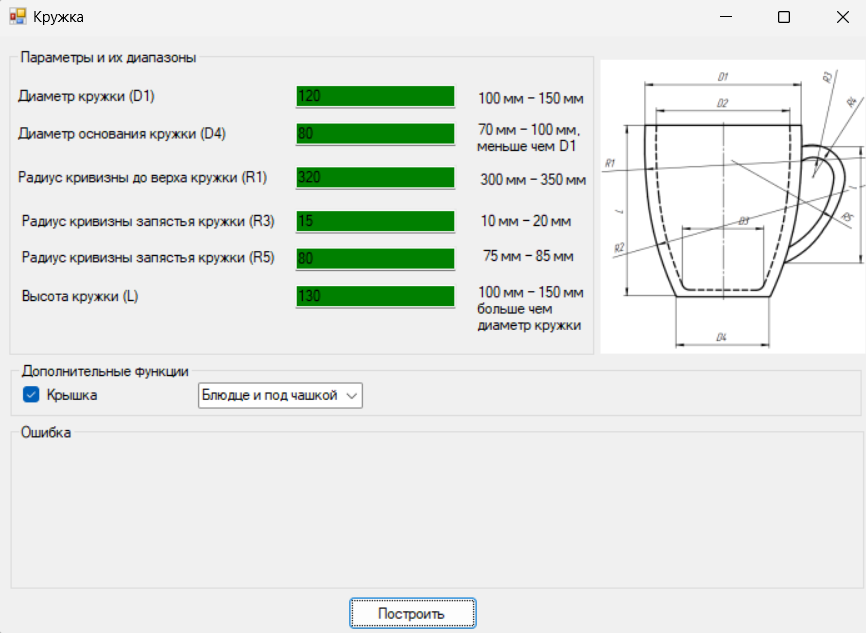


Рисунок 9.9 – Стандартные параметрыи и дополнительные функции

На рисунке 9.10 представлен результат построения модели с стандартными параметрами и дополнительными функциями.

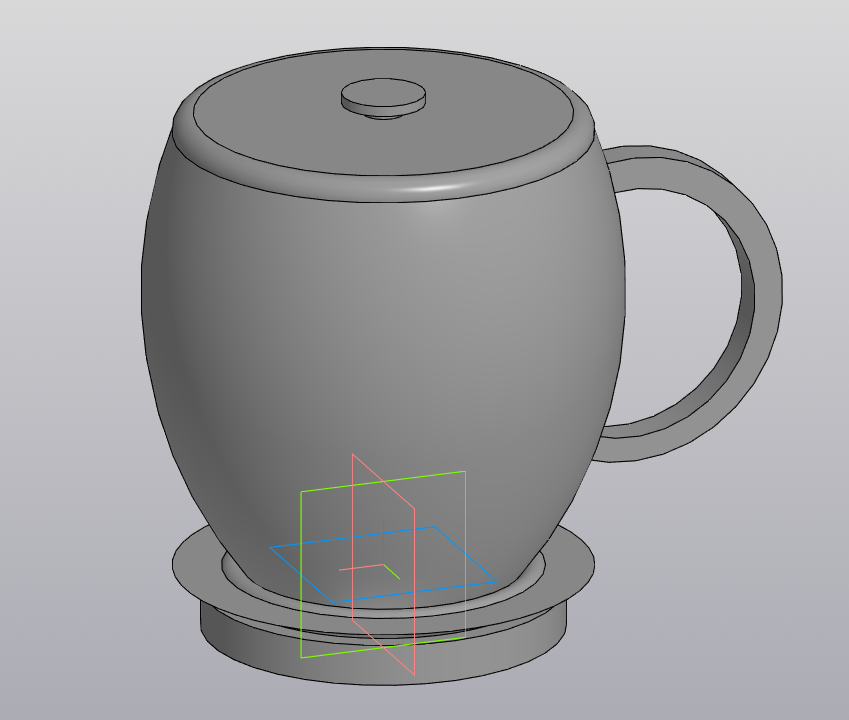


Рисунок 9.10 – Модель по стандартным параметрам и дополнительным функциям

Обратите внимание, что дополнительные функции могут быть выбраны пользователем произвольно. Он может выбирать, какие функции добавлять, а какие нет, поскольку добавление дополнительных функций не является обязательным.

## 9.2 Модульное тестирование

На рисунке 9.11 представлено количество написанных Unit-тестов, а также что их выполнение происходит корректно.

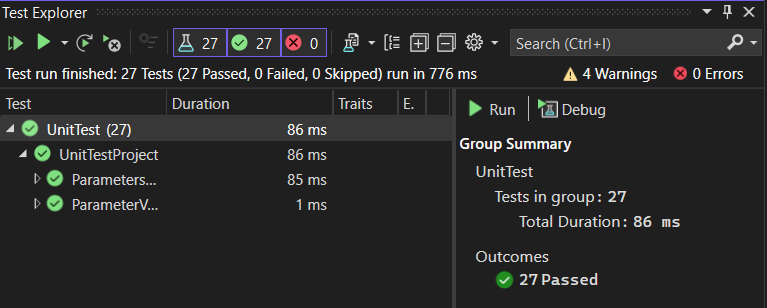


Рисунок 9.11 – Количество написанных Unit-тестов

Необходимо было написать тесты для 2-ух классов: Parameters и Parameter. В таблице 9.1 представлены все написанные тесты и их описание.

Таблица 9.1 – Unit-тесты

|  |  |
| --- | --- |
| Название теста | Описание теста |
| Validate\_BodyWidth\_WithinRange\_ShouldPass | Проверяет ширину корпуса, если значение находится в пределах допустимого диапазона. |
| Validate\_BodyWidth\_OutOfRange\_ShouldThrowException | Проверяет ширину корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| Validate\_BaseWidth\_WithinRange\_ShouldPass | Проверяет ширину основания, если значение находится в пределах допустимого диапазона. |
| Validate\_BaseWidth\_OutOfRange\_ShouldThrowException | Проверяет ширину основания, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| Validate\_BodyRadius1\_WithinRange\_ShouldPass | Проверяет радиус корпуса, если значение находится в пределах допустимого диапазона. |
| Validate\_BodyRadius1\_OutOfRange\_ShouldThrowException | Проверяет радиус корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |

Продолжение таблицы 9.1

|  |  |
| --- | --- |
| Название теста | Описание теста |
| Validate\_HandleRadius3\_WithinRange\_ShouldPass | Проверяет радиус ручки, если значение находится в пределах допустимого диапазона. |
| Validate\_HandleRadius3\_OutOfRange\_ShouldThrowException | Проверяет радиус ручки, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| Validate\_HandleRadius5\_WithinRange\_ShouldPass | Проверяет радиус ручки, если значение находится в пределах допустимого диапазона. |
| Validate\_HandleRadius5\_OutOfRange\_ShouldThrowException | Проверяет радиус ручки, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| Validate\_BodyLength\_WithinRange\_ShouldPass | Проверяет длину корпуса, если значение находится в пределах допустимого диапазона. |
| Validate\_BodyLength\_OutOfRange\_ShouldThrowException | Проверяет длину корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| Validate\_InteriorBaseWidth\_Dependency\_ShouldPass | Проверяет зависимость ширины внутреннего основания. |
| Validate\_InteriorBaseWidth\_Dependency\_ShouldThrowException | Проверяет зависимость ширины внутреннего основания, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |

Продолжение таблицы 9.1

|  |  |
| --- | --- |
| Название теста | Описание теста |
| Validate\_AllParameters\_ValidValuesAndDependencies\_ShouldPass | Проверяет все параметры с корректными значениями и зависимостями. |
| BodyWidth\_Initialization\_ShouldPass | Проверяет инициализацию ширины корпуса. Значения находятся в допустимом диапазоне. |
| BodyWidth\_InvalidValue\_ShouldThrowException | Проверяет ширину корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| BaseWidth\_Initialization\_ShouldPass | Проверяет инициализацию ширины основания. Значения находятся в допустимом диапазоне. |
| BaseWidth\_InvalidValue\_ShouldThrowException | Проверяет ширину основания, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| BodyRadius1\_Initialization\_ShouldPass | Проверяет инициализацию радиуса корпуса. Значения находятся в допустимом диапазоне. |
| BodyRadius1\_InvalidValue\_ShouldThrowException | Проверяет радиус корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| HandleRadius3\_Initialization\_ShouldPass | Проверяет инициализацию радиуса ручки. Значения находятся в допустимом диапазоне. |
| HandleRadius3\_InvalidValue\_ShouldThrowException | Проверяет радиус ручки, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |

Продолжение таблицы 9.1

|  |  |
| --- | --- |
| Название теста | Описание теста |
| HandleRadius5\_Initialization\_ShouldPass | Проверяет инициализацию радиуса ручки. Значения находятся в допустимом диапазоне. |
| HandleRadius5\_InvalidValue\_ShouldThrowException | Проверяет радиус ручки, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| BodyLength\_Initialization\_ShouldPass | Проверяет инициализацию длины корпуса. Значения находятся в допустимом диапазоне. |
| BodyLength\_InvalidValue\_ShouldThrowException | Проверяет длину корпуса, если значение выходит за пределы диапазона. Ожидается исключение ArgumentException. |
| InteriorUpWidth\_ValidDependency\_ShouldPass | Проверяет зависимость ширины внутренней верхней части корпуса. |
| InteriorBaseWidth\_ValidDependency\_ShouldPass | Проверяет зависимость ширины внутреннего основания. |
| BodyRadius2\_ValidDependency\_ShouldPass | Проверяет зависимость радиуса корпуса. |
| HandleRadius4\_ValidDependency\_ShouldPass | Проверяет зависимость радиуса ручки. |
| HandleLength\_ValidDependency\_ShouldPass | Проверяет зависимость длины ручки. |

На рисунке 9.12 также представлен скриншот плагина, измеряющего процент покрытия модульными тестами

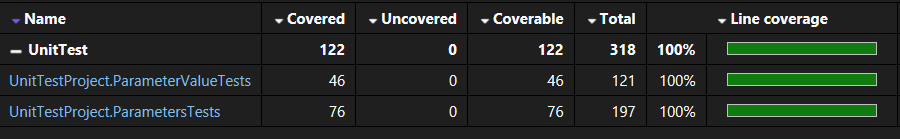


Рисунок 9.12 – Результаты плагина

## 9.3 Нагрузочное тестирование

На рисунке 9.13 представлен график зависимости памяти ОЗУ от построения модели, а на рисунке 9.14 представлен график зависимости времени от построения модели.

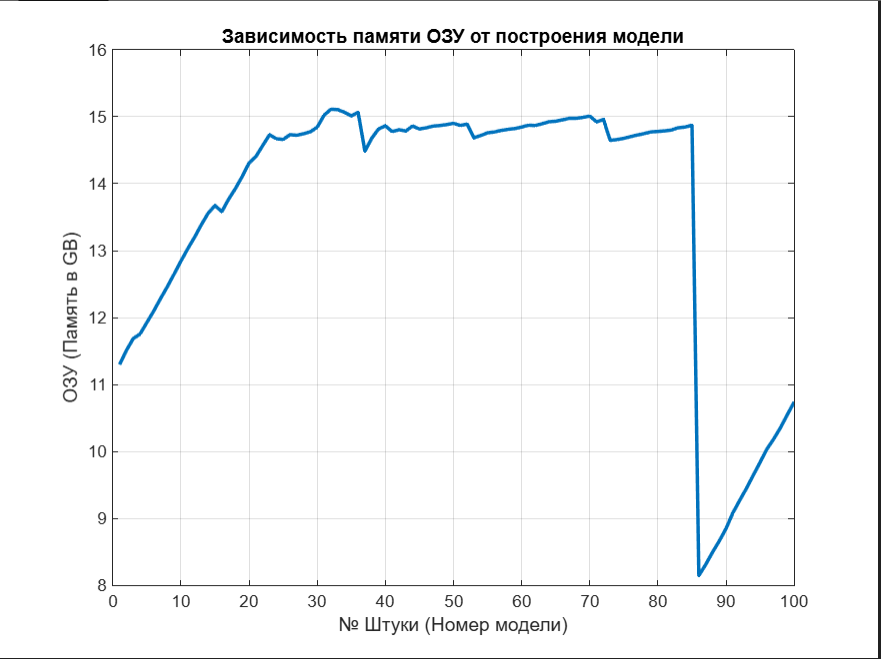


Рисунок 9.13 – График зависимости памяти ОЗУ от количества построенных моделей

Из представленного графика 9.13 можно сделать следующий вывод: на начальном этапе (до ~20 моделей) наблюдается стабильный рост объема используемой оперативной памяти. В этом диапазоне происходит плавное увеличение ОЗУ, предположительно, за счет загрузки исходных данных и минимальных вычислительных операций. Начиная с 20-й модели, использование памяти стабилизируется на уровне ~14.5 ГБ и сохраняет относительно постоянное значение до ~80-й модели. Это может быть связано с тем, что система достигает своего оптимального режима работы, где дополнительно выделяемая память минимальна. Однако после 80-й модели происходит резкий спад до уровня ~8 ГБ. Такое поведение может указывать на сброс оперативной памяти или освобождение зарезервированного объема, что может быть связано с изменением подхода системы к управлению памятью или завершением определенных операций. В заключении можно отметить, что данное распределение ОЗУ не связано со сложностью моделей, а, скорее всего, с особенностями управления памятью в используемой программе или ограничениями вычислительных ресурсов.

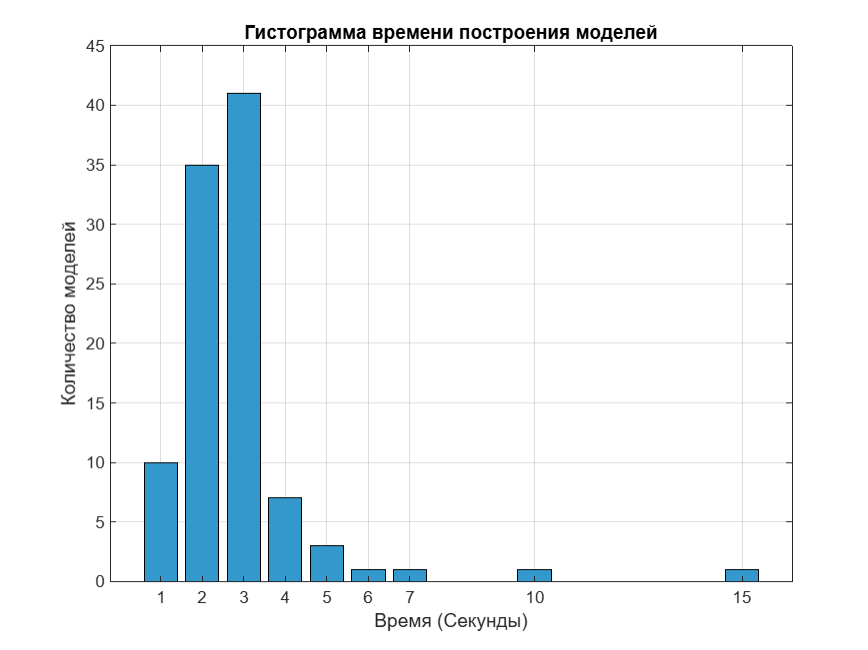


Рисунок 9.14 – График гистограммы построения модели

По гистограмме 9.14 можно сделать вывод, что основное время построения моделей находится в диапазоне от 2 до 3 секунд, что составляет наибольшее количество моделей. Это указывает на то, что процесс построения большинства моделей является достаточно простым и не требует значительных вычислительных ресурсов. Модели с временем построения менее 2 секунд также занимают заметную долю, что может свидетельствовать о минимальных требованиях к ресурсам для этих задач. Модели, построение которых заняло от 4 до 5 секунд, можно связать с увеличением загрузки системы, например, выполнением фоновых задач ОС, которые влияют на скорость выполнения процесса. Временные значения, превышающие 7 секунд, являются редкостью и, вероятно, обусловлены исключительными случаями, такими как сбои в системе или необходимость выполнения более сложных вычислений. В целом, можно сделать вывод, что процесс моделирования оптимизирован и стабилен для большинства задач, укладываясь в промежуток 2-3 секунды.

**10 ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторных работ был создан плагин для КОМПАС-3D, который автоматически генерирует кружку на основе параметров, задаваемых пользователем. Каждая из лабораторных работ была направлена на поэтапное достижение этой цели. Выбор объекта моделирования и системы автоматизированного проектирования (САПР) способствовал углублению знаний в соответствующих областях, а техническое задание помогло задать чёткий вектор работы и внести необходимые коррективы в процесс разработки. Этап проектирования системы позволил заранее предусмотреть множество нюансов, минимизировав вероятность крупных изменений на стадии реализации.

Процесс написания кода включал изучение работы с API КОМПАС-3D и создание пользовательского интерфейса, что потребовало внимания к мелким деталям. Неожиданным открытием стало осознание недостатков первоначального подхода к валидации параметров, что привело к пересмотру решений и улучшению конечного результата. Также был получен полезный опыт работы с инструментами для стандартизации кода, такими как StyleCops, что значительно повысило читаемость и качество написанного кода, облегчая его дальнейшую поддержку и доработку.

**11 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 23501.101-87 «Системы автоматизированного проектирования. Основные положения» (дата обращения 13.12.2024)
2. API [Электронный ресурс]. − Режим доступа <http://habrahabr.ru/post/201700/> (дата обращения 14.10.2024).
3. ГОСТ 2417-75 «Кружки мерные металлические» [Электронный ресурс]. − Режим доступа <https://www.normacs.ru/Doclist/doc/35H9.html> (дата обращения 14.10.2024).
4. Windows Forms [Электронный ресурс]. − Режим доступа <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-9.0> (дата обращения 13.12.2024)
5. Github [Электронный ресурс]. − Режим доступа <https://docs.github.com/ru/get-started/start-your-journey/about-github-and-git> (13.12.2024)
6. ReSharper [Электронный ресурс]. − Режим доступа <https://www.jetbrains.com/ru-ru/resharper/> (13.12.2024)
7. Fine Code Coverage [Электронный ресурс]. − Режим доступа <https://marketplace.visualstudio.com/items?itemName=FortuneNgwenya.FineCodeCoverage> (13.12.2024)
8. StyleCop [Электронный ресурс]. − Режим доступа <https://andrey.moveax.ru/post/net-standard-using-style-cop> (13.12.2024)
9. NUnit [Электронный ресурс]. − Режим доступа <https://nunit.org/> (13.12.2024)
10. Разъёмные соединения [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/application/machinery/threaded-connection/> (дата обращения 05.10.2024)
11. Валы и механические передачи 3D. [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/application/machinery/gear-cutting/> (дата обращения 05.10.2024)
12. Введение в UML от создателей языка руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. - 2-е изд. - М. : ДМК Пресс, 2012. - 494 с. : ил. - (Классика программирования). - Предм. указ.: с. 483- 493. - ISBN 978-5-94074-644-7 [Электронный ресурс] : − Режим доступа [https://old-vt.chuvsu.ru/RmtLrn](https://old-vt.chuvsu.ru/RmtLrn/%D0%9A%D0%B0%D1%84%D0%B5%D0%B4%D1%80%D0%B0%20%D0%9A%D0%A2%20(%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D1%8B%20%D1%81%20%D0%BB%D0%B8%D1%82%D0%B5%D1%80%D0%BE%D0%B9%20%D0%9A%D0%A2)/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%20%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%98%D0%A1_%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%20%D0%90%D0%A5/UML_%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D1%8B.pdf) (дата обращения 05.10.2024).