Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОННИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Разработка плагина «Мастер шестерёнок» для САПР «AutoCAD»

Пояснительная записка по дисциплине «Основы разработки САПР»

Выполнил:

Студент гр. 580-2

\_\_\_\_\_\_\_\_\_ Г.П. Лубов

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

Проверил:

доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев

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

Оглавление

[Введение 3](#_Toc154603034)

[1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_Toc154603035)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 5](#_Toc154603036)

[3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 6](#_Toc154603037)

[4 НАЗНАЧЕНИЕ ПЛАГИНА 7](#_Toc154603038)

[5 ОБЗОР АНАЛОГОВ 8](#_Toc154603039)

[6 ОПИСАНИЕ РЕАЛИЗАЦИИ 11](#_Toc154603040)

[7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 20](#_Toc154603041)

[8 ТЕСТИРОВАНИЕ ПЛАГИНА 23](#_Toc154603042)

[8.1 Функциональное тестирование 23](#_Toc154603043)

[8.2 Модульное тестирование 28](#_Toc154603044)

[8.3 Нагрузочное тестирование 31](#_Toc154603045)

[Заключение 33](#_Toc154603046)

[Список использованных источников 34](#_Toc154603047)

# 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.

Проектирование шестерёнок является трудоёмкой задачей ввиду большого количества моделируемых элементов. В данной работе рассматривается разработка плагина для системы автоматизированного проектирования (САПР) Autodesk AutoCAD, позволяющего автоматизировать проектирование шестерёнок.

# 1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

Целью работы является разработка плагина «Мастер шестерёнок» для системы автоматизированного проектирования (САПР) AutoCAD. Система должна быть выполнена в качестве встроенного плагина AutoCAD, который запускается непосредственно из САПР. У плагина должны быть изменяемые параметры: внешний диаметр шестерни, внутренний диаметр отверстия, толщина, высота зуба и количество зубьев. Все размеры должны указываться в системе СИ. В плагине должны проходить проверки значений, вводимых пользователем. Реализуемый плагин должен обеспечивать обработку ошибочных ситуаций, возникающих в процессе работы. При нажатии на кнопку «Построить» должна проходить проверка правильности ввода данных. Если данные некорректные, то под кнопкой построения в строке состояния должно появиться сообщение о том, какие параметры введены неверно, а также поля ввода для параметров должны подсветиться красным цветом.

В рамках проекта были поставлены задачи:

1. Составить ТЗ (25.09.2023 – 07.10.2023);
2. Составить проект системы (09.10.2023 – 20.11.2023);
3. Реализовать систему (21.11.2023 – 18.12.2023);
4. Реализовать дополнительную функциональность, выбранную преподавателем (18.12.2023 – 28.12.2023);
5. Написать пояснительную записку (18.12.2023 – 28.12.2023).

В результате анализа были отмечены следующие пункты:

* При написании плагина возникнет трудность с использованием AutoCAD API ввиду отсутствия обучающих материалов и опыта работы с ним;
* Есть вероятность столкнуться с технической сложностью моделирования большого количества зубьев через API;
* Ресурсоёмкость приложения может оказаться слишком большой ввиду моделирования большого количество твёрдых тел и произведении операций над ними.

# 2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Предметом проектирования является зубчатая прямозубая шестерня.

Параметры шестерни:

* Внешний радиус по вершинам D (1 — 1000 мм, но не менее d);
* Радиус посадочного отверстия d (1 — 999 мм, но не более, чем 9D/10);
* Толщина шестерни (1 — 1000 мм);
* Высота зуба h (0,5 — 999 мм, но не более, чем радиус посадочного отверстия);
* Количество зубьев (3 — 1000 штук).

На рисунке 2.1 показаны чертёж моделируемой шестерни.



Рисунок 2.1 – Чертёж шестерни

В качестве дополнительной функциональности были выбраны следующие пункты:

* Рефакторинг пользовательского интерфейса;
* Внедрение системы журналирования для приложения.

# 3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ

В процессе разработки плагина использовались следующие технологии:

* Windows Presentation Foundation (WPF);
* Object ARX;
* StyleCop.

Windows Presentation Foundation (WPF) — это фреймворк для создания оконных приложений для Windows. Преимуществом данной технологии является встроенная поддержка высокоуровневого паттерна проектирования MVVM [1].

В качестве платформы был использована .NET Framework. Платформа .NET Framework — это технология, которая поддерживает создание и выполнение веб-служб и приложений Windows [2].

Object ARX – это официальный API для системы автоматизированного проектирования AutoCAD. Написан на C++, но совместим с C#. Предоставляет классы и методы для построения моделей в приложении [3].

StyleCop – это инструмент статического анализа кода с открытым исходным кодом от Microsoft, который проверяет код C# на соответствие рекомендуемым стилям кодирования StyleCop и подмножеству руководящих указаний Microsoft по проектированию .NET Framework. Не позволяет скомпилировать приложение до тех пор, пока не будут соблюдены правила оформления кода [4].

# 4 НАЗНАЧЕНИЕ ПЛАГИНА

Данный плагин позволяет сократить трудоёмкость моделирования шестерёнок за счёт автоматизированного проектирования шестерёнок по определённым пользователем параметрам.

# 5 ОБЗОР АНАЛОГОВ

Среди прямых аналогов на сегодняшний день присутствую следующие решения:

* Цилиндрическое зубчатое зацепление – инструмент в программе Autodesk Inventor, позволяющий создавать зубчатые передачи ременного, цепного, червячного, цилиндрического и конического зацепления косозубого и прямозубого типов по заданным параметрам [6]. С помощью данного инструмента можно в автоматизированном режиме получить готовый узел зубчатой передачи по заданным параметрам. В данный момент данный программный продукт на территории РФ не распространяется. Пользовательский интерфейс инструмента представлен на рисунке 2.
* Autodesk Fusion 360 – система автоматизированного проектирования, включающая в себя модули автоматизированного геометрического моделирования, инженерных расчётов, производства, проектирования печатных плат и автоматизации формирования конструкторской документации. Преимуществом данного программного продукта является простой пользовательский интерфейс и возможность работы в облаке с конструкторской документацией [7]. С помощью данного программного продукта можно вручную создать модель шестерни. В данный момент этот программный продукт на территории РФ не распространяется. Пользовательский интерфейс программы представлен на рисунке 3.;

Kompas-3D – система автоматизированного проектирования отечественной разработки, включающая в себя модули автоматизированного геометрического моделирования, инженерных расчётов, производства, проектирования печатных плат, автоматизации и формирования конструкторской документации [8]. Изначально система была ориентирована на создание конструкторской документации в соответствии с ЕСКД, ЕСТД, СПДС и международными стандартами. Преимуществом данного программного обеспечения является его доступность в РФ, в отличие от импортных САПР. С помощью данного программного продукта можно вручную создать модель шестерни. Пользовательский интерфейс программы представлен на рисунке 5.1.

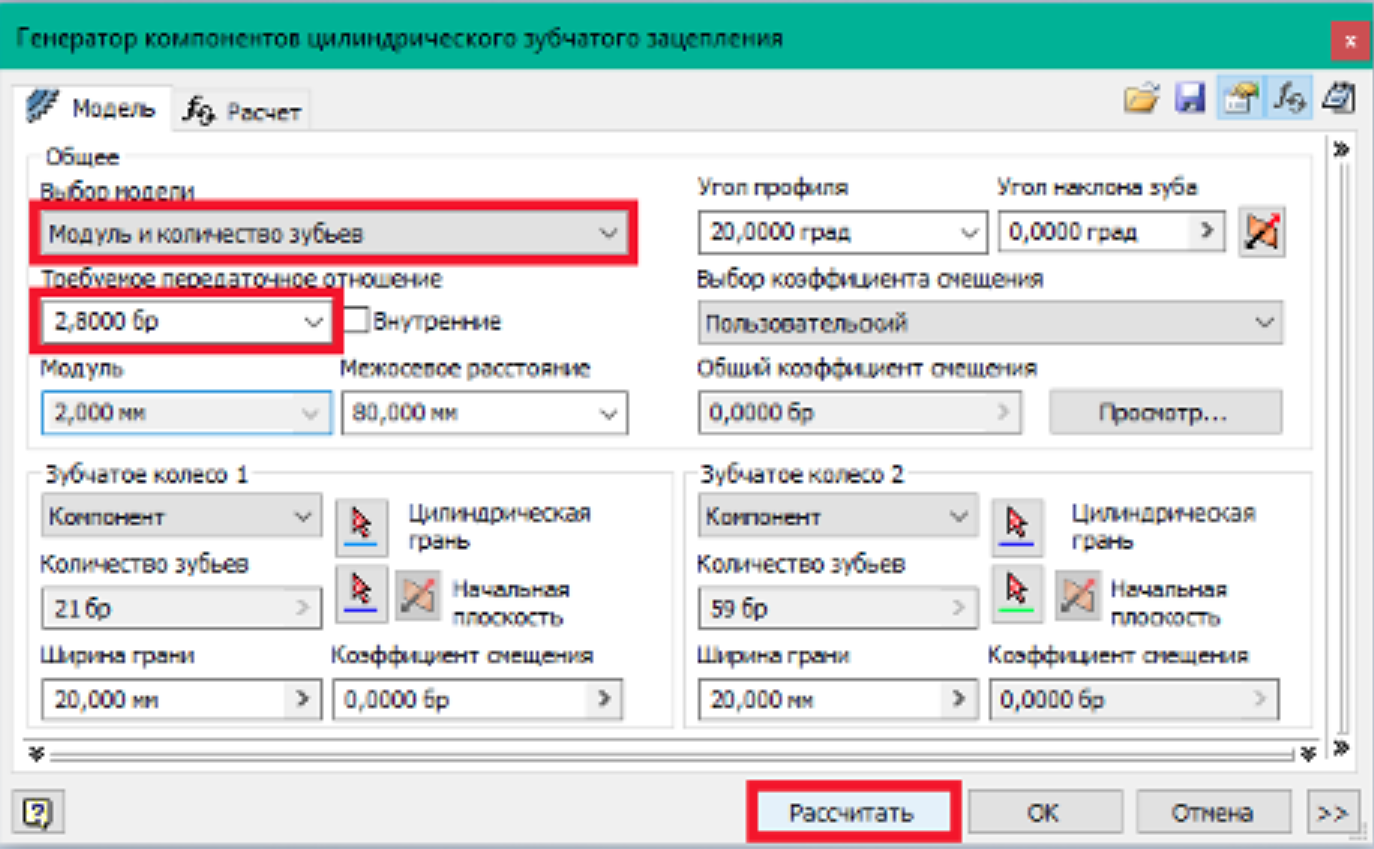


Рисунок 5.1 – Пользовательский интерфейс инструмента “Цилиндрическое зубчатое зацепление” в программе Autodesk Inventor

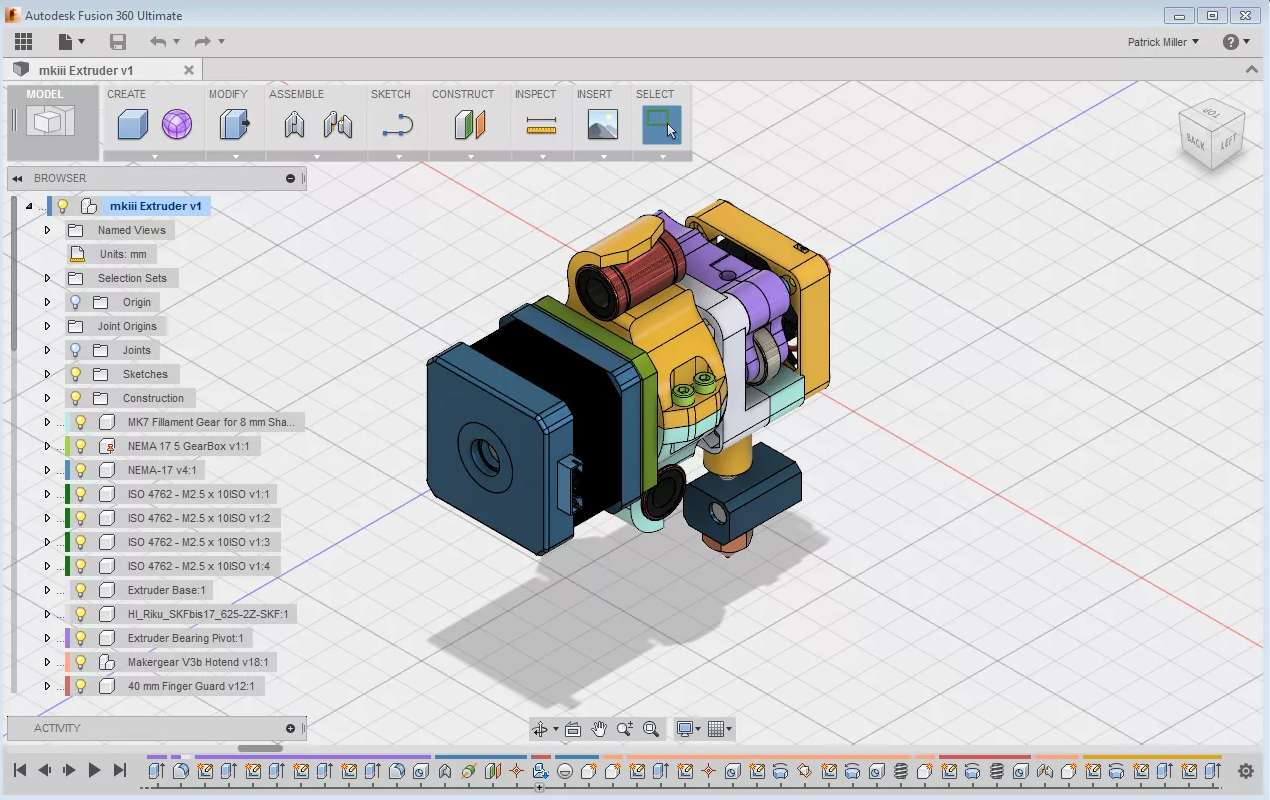


Рисунок 5.2 – Пользовательский интерфейс Autodesk Fusion 360

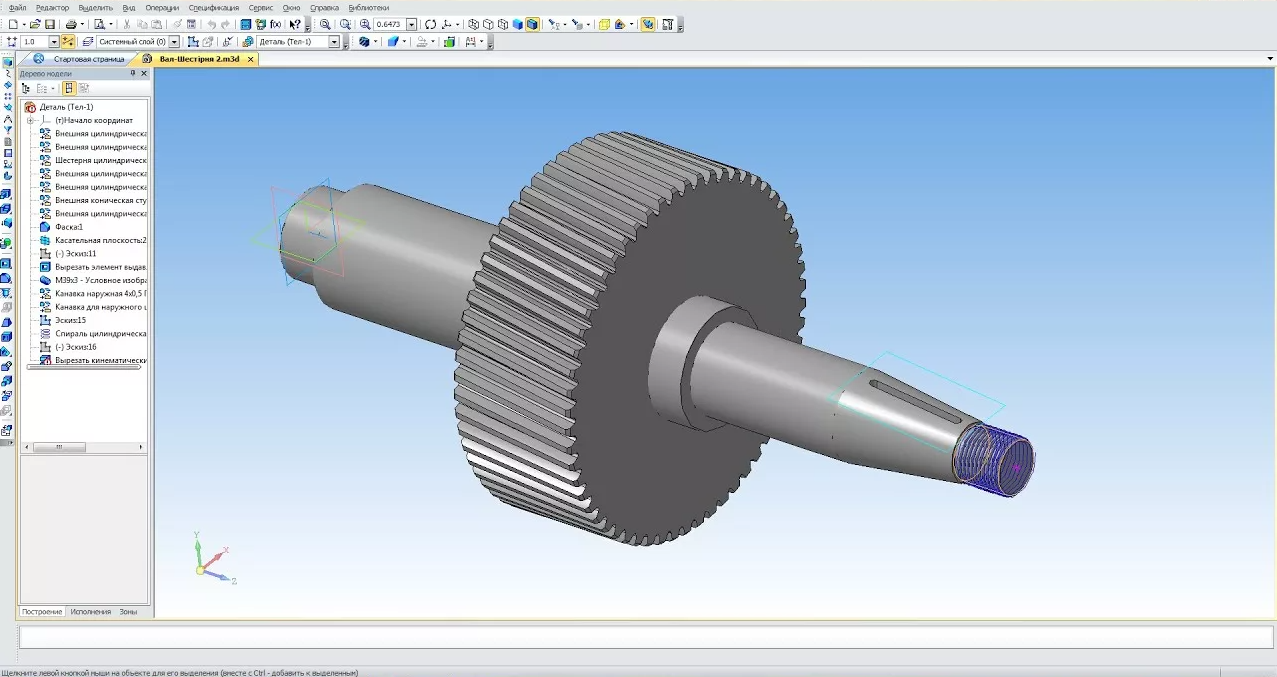


Рисунок 5.3 – пользовательский интерфейс Kompas-3D

# 6 ОПИСАНИЕ РЕАЛИЗАЦИИ

На рисунке 6.1 показана UML диаграмма классов после проектирования.

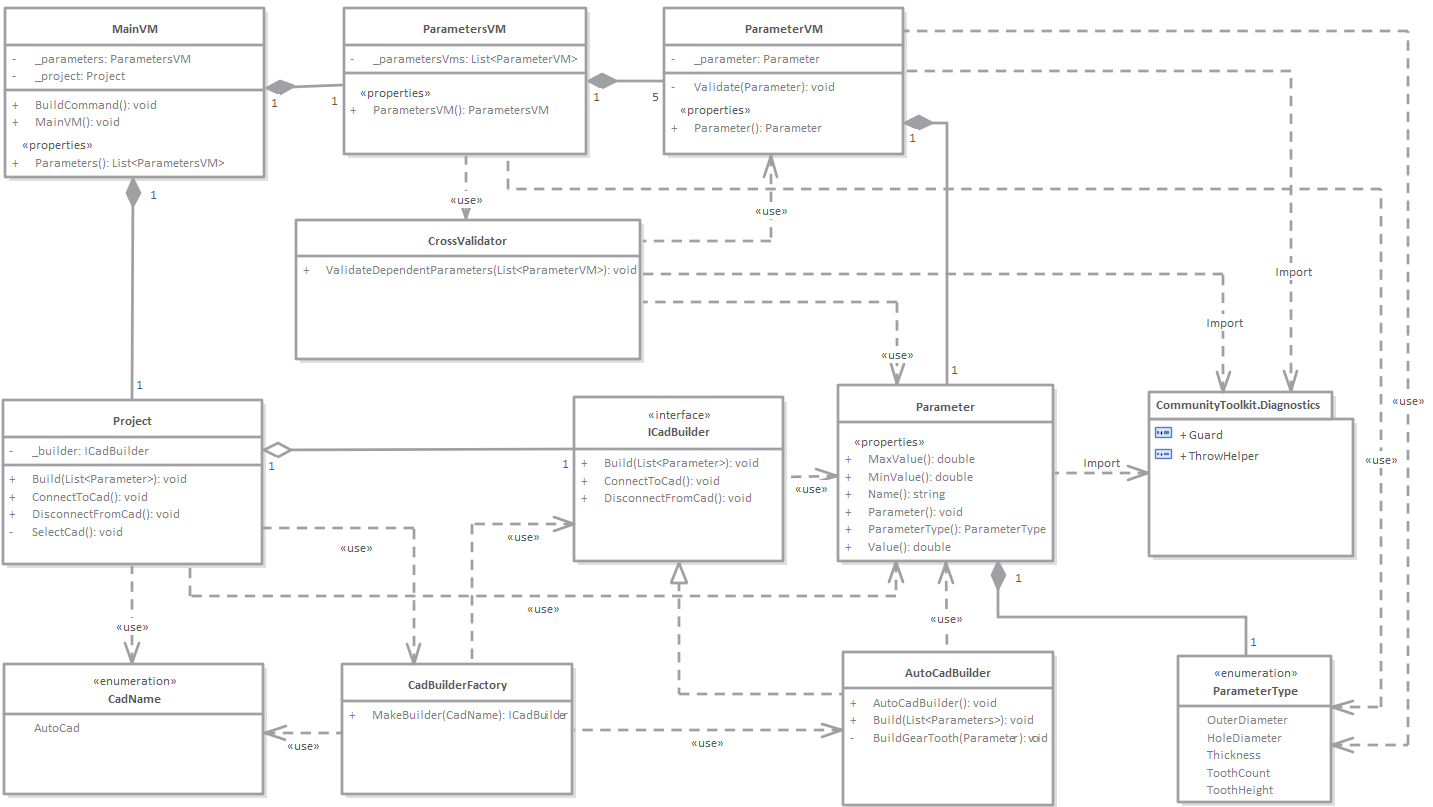


Рисунок 6.1 – UML диаграмма классов до реализации программы

Архитектура приложения реализована по паттерну MVVM.

Класс MainVM представляет собой объект, через который будет осуществляться обработка пользовательского ввода и передача его в модель (таблица 6.1).

Таблица 6.1 – Описание класса MainVM

|  |  |
| --- | --- |
| Название | Описание |
| \_project | Хранит экземпляр проекта |
| \_parameters | Хранит параметры шестерни |
| BuildCommand | Вызывает команду построения шестерни |
| MainVM | Конструктор |
| ParamatersVM | Хранит все параметры шестерни |

Класс Parameters представляет группу параметров моделируемого объекта (таблица 6.2).

Таблица 6.2 – Описание класса ParametersVM

|  |  |
| --- | --- |
| Название | Описание |
| \_parametersVms | Параметры шестерни |
| ParametersVM | Конструктор |

Класс ParameterVM является представлением параметра, который отвечает за первичную валидацию параметра и дальнейшую отправку этого параметра на уровень модели (таблица 6.3).

Таблица 6.3 – Описание класса ParameterVM

|  |  |
| --- | --- |
| Название | Описание |
| Parameter | Хранит параметр |
| \_parameter | Хранит параметр |
| Validate | Выполняет валидацию текущего фрагмента |

Класс CrossValidator осуществляет перекрёстную валидацию между зависимыми параметрами (таблица 6.4).

Таблица 6.4 – Описание класса CrossValidator

|  |  |
| --- | --- |
| Название | Описание |
| ValidateDependentParameters | Валидирует зависимые параметры |

Класс Project является главным классом модели. Через него происходит взаимодействие с САПР, выполняется подключение, отключение, выбор целевой САПР и выполняется построение модели (таблица 6.5).

Таблица 6.5 – Описание класса Project

|  |  |
| --- | --- |
| Название | Описание |
| \_builder | Хранит экземпляр обёртки над построителем |
| Build | Выполняет построение модели |
| ConnectToCad | Выполняет подключение к САПР |
| DisconnectFromCad | Выполняет отключение от САПР |
| SelectCad | Выбирает конкретную САПР для использования при построении |

Интерфейс ICadBuilder абстрагирует Project от конкретной реализации построителя (таблица 6.6).

Таблица 6.6 – Описание интерфейса ICadBuilder

|  |  |
| --- | --- |
| Название | Описание |
| Build | Выполняет построение модели в САПР |
| ConnectToCad | Выполняет подключение к САПР |
| DisconnectFromCad | Выполняет отключение от САПР |

Класс Parameter выполняет хранение информации об одном из параметров модели (таблица 6.7).

Таблица 6.7 – Описание класса Parameter

|  |  |
| --- | --- |
| Название | Описание |
| MaxValue | Хранит максимальное допустимое значение параметра |
| MinValue | Хранит минимальное допустимое значение параметра |
| Name | Хранит название параметра |
| Parameter | Конструктор |
| Value | Значение параметра |
| Type | Тип параметра |

Перечисление CadName отображает названия САПР, с которыми в настоящий момент времени может работать плагин (таблица 6.8).

Таблица 6.8 – Описание перечисления CadName

|  |  |
| --- | --- |
| Название | Описание |
| AutoCad | Перечисление для Autodesk AutoCAD |

Класс CadBuilderFactory предоставляет конкретный экземпляр ICadBuilder классу Project. Данное решение применено для упрощения модификации плагина в будущем на тот случай, если потребуется подключить этот плагин к другой САПР (таблица 6.9).

Таблица 6.9 – Описание класса CadBuilderFactory

|  |  |
| --- | --- |
| Название | Описание |
| MakeBuilder | Создаёт экземпляр конкретного построителя модели в соответствующей САПР |

Класс AutoCadBuilder предоставляет конкретную реализацию построителя для САПР AutoCAD (таблица 6.10).

Таблица 6.10 – Описание класса AutoCadBuilder

|  |  |
| --- | --- |
| Название | Описание |
| AutoCadBuilder | Конструктор |
| Build | Выполняет построение модели в AutoCAD |
| BuildGearTooth | Выполняет построение одного зуба в AutoCAD |

Библиотека CommunityToolkit.Guard предоставляет реализацию методов валидации без необходимости реализовывать их самостоятельно (рсиунок 6.11).

Таблица 6.11 – Описание библиотеки CommunityToolkit.Guard

|  |  |
| --- | --- |
| Название | Описание |
| Guard | Реализует методы валидации значений на минимум, максимум, т.п. |
| ThrowHelper | Реализует методы для облегчения выбрасывания исключений |

Запуск плагина предполагается выполнять из САПР через командную строку.

В итоговом проекте созданы следующие и методы, которые отображены на итоговой диаграмме классов (рисунок 6.2).

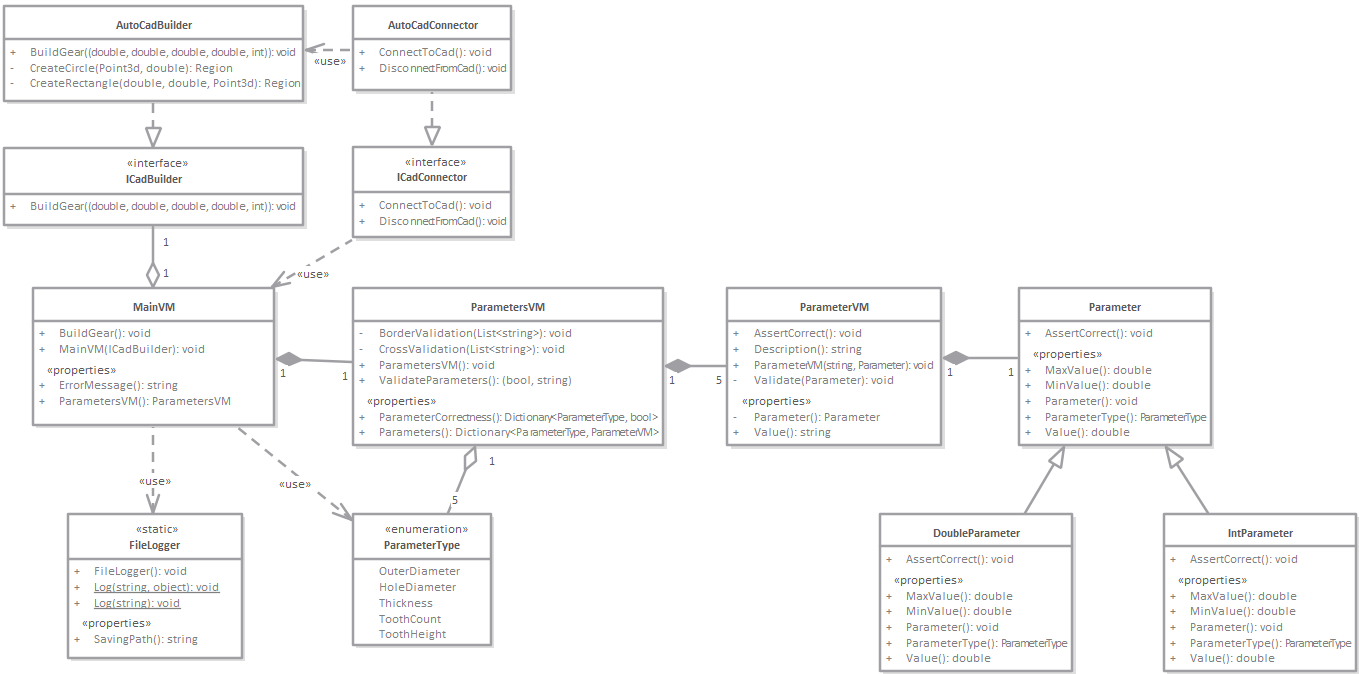


Рисунок 6.2 – UML диаграмма классов после реализации программы

При реализации системы были сделаны следующие архитектурные изменения:

1. Ответственность за валидацию пользовательского ввода передана классу ParametersVM, потому что класс CrossValidator нельзя было бы повторно использовать без класса ParametersVM;
2. Ответственность запуска построения модели теперь лежит на MainVM, потому что класс Project (который ранее отвечал за это) имел единственную ответственность в виде построения и был излишним;
3. Ответственность по подключению к САПР и выполнению построения модели разделена на 2 интерфейса: ICadConnector и ICadBuilder и 2 класса: AutoCadConnector и AutoCadBuilder соответственно, что обеспечивает гибкость реализации;
4. Ввиду простоты реализации было решено отказаться от пакета CommunityToolkit.Diagnostics;
5. Ввиду добавления функциональности, был добавлен класс FileLogger, осуществляющий журналирование действий в программе.

Ключевым классом приложения является MainVM (таблица 6.12).

Таблица 6.12 – Описание класса MainVM

|  |  |
| --- | --- |
| Название | Описание |
| ErrorMessage | Хранит строку состояния с ошибками валидации |
| BuildGear | Вызывает построение шестерни |
| MainVM | Конструктор |
| ParamatersVM | Хранит все параметры шестерни |

Для валидации параметров используется класс ParametersVM (таблица 6.13).

Таблица 6.13 – Описание класса ParametersVM

|  |  |
| --- | --- |
| Название | Описание |
| BorderValidation() | Выполняет валидацию параметра по допустимым диапазонам значений |
| CrossValidation() | Выполняет кросс-валидацию параметров |
| ValidateParameters() | Выполняет полную валидацию параметров |
| ParametersVM() | Конструктор |
| ParameterCorrectness | Отображает текущее состояние корректности введённых параметров |
| Parameters | Хранит параметры |

Класс ParameterVM является представлением параметра, который отвечает за первичную валидацию параметра и дальнейшую отправку этого параметра на уровень модели (таблица 6.14).

Таблица 6.14 – Описание класса ParameterVM

|  |  |
| --- | --- |
| Название | Описание |
| Parameter | Хранит параметр |
| Value | Обеспечивает доступ к значению параметра |
| Description | Описание параметра |
| AssertCorrect | Выполняет валидацию текущего параметра по допустимому диапазону значений и по принадлежности к целевому типу данных |
| ParameterVM | Конструктор |

Класс Parameter хранит значение параметра и выполняет валидацию (таблица 6.15.

Таблица 6.15 – Описание класса Parameter

|  |  |
| --- | --- |
| Название | Описание |
| MinValue | Хранит минимально допустимое значение для параметра |
| Value | Хранит текущее значение параметра |
| MinValue | Хранит максимальное значение для параметра |
| AssertCorrect | Выполняет валидацию текущего параметра по допустимому диапазону значений и по принадлежности к целевому типу данных |
| Parameter | Конструктор |
| ParameterType | Определяет принадлежность параметра к конкретному параметру шестерни |

Перечисление ParameterType содержит все параметры моделируемой шестерни.

Класс AutoCadBuilder выполняет построение шестерни (таблица 6.16).

Таблица 6.16 – Описание класса AutoCadBuilder

|  |  |
| --- | --- |
| Название | Описание |
| BuildGear | Выполняет построение шестерни |
| BuildRectangle | Выполняет построение эскиза четырёхугольника |
| BuildCircle | Выполняет построение эскиза окружности |

Класс FileLogger отвечает за журналирование действий программы в файл (таблица 6.17).

Таблица 6.17 – Описание класса FileLogger

|  |  |
| --- | --- |
| Название | Описание |
| Log | Выполняет журналирование сообщения |
| SavingPath | Хранит путь для сохранения журнала действий программы |
| FileLogger | Конструктор |

Интерфейсы ICadBuilder и ICadConnector необходимы для обеспечения более гибкой архитектуры приложения и упрощения его модификации в будущем.

Класс AutoCadConnector выполняет подключение к текущему файлу AutoCAD, из которого была выполнена команда запуска плагина.

Классы IntParameter и DoubleParameter являются наследниками класса Parameter. Такой поход упрощает валидацию введённых параметров за счёт использования полиморфизма.

# 7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ

Для начала работы с плагином пользователь должен загрузить файл библиотеки коннектора c расширением “.dll”. После загрузки файла пользователю станет доступна команда “GearMaster”, которая запускает плагин.

После выполнения команды, открывается окно плагина с установленными по умолчанию значениями параметров и пользователю становится доступны поля для ввода данных. Пользовательский интерфейс плагина представлен на рисунке 7.1.

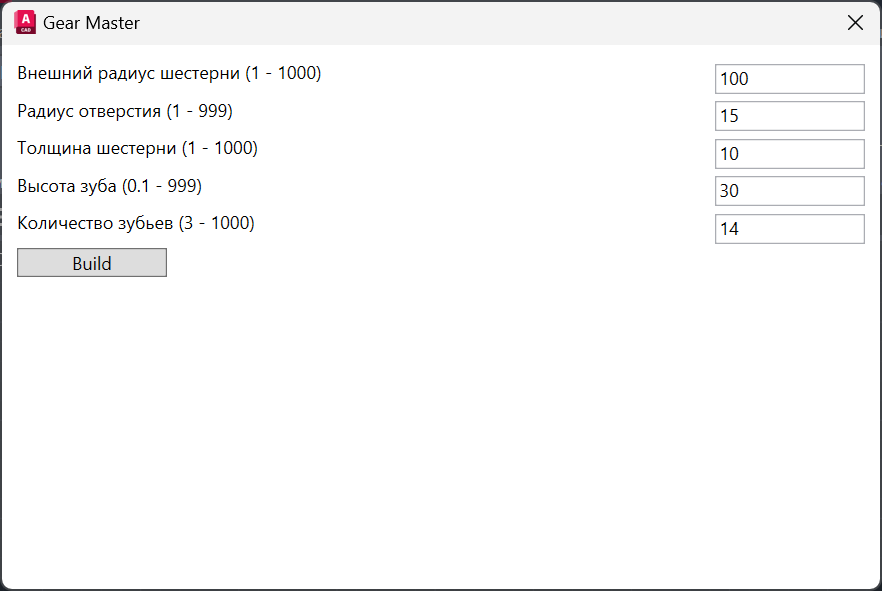


Рисунок 7.1 – Пользовательский интерфейс плагина

Пользовательский интерфейс плагина состоит из пяти параметров, доступных для ввода пользовательских данных, кнопки построения и строки состояния, которая сообщает о наличии ошибок пользовательского ввода.

В плагине реализована проверка пользовательского ввода на минимально и максимально допустимые значения параметров, а также перекрёстная проверка между параметрами.

При вводе некорректных пользовательских данных, соответствующие поля подсвечиваются красным цветом, а в строке состояния под кнопкой построения появляются сообщения об ошибках.

Если все данные были введены корректно, то при нажатии кнопки построения происходит построение модели в файле, из которого была загружена библиотека с плагином. Пользовательский интерфейс с неверно введёнными параметрами представлен на рисунке 7.2.

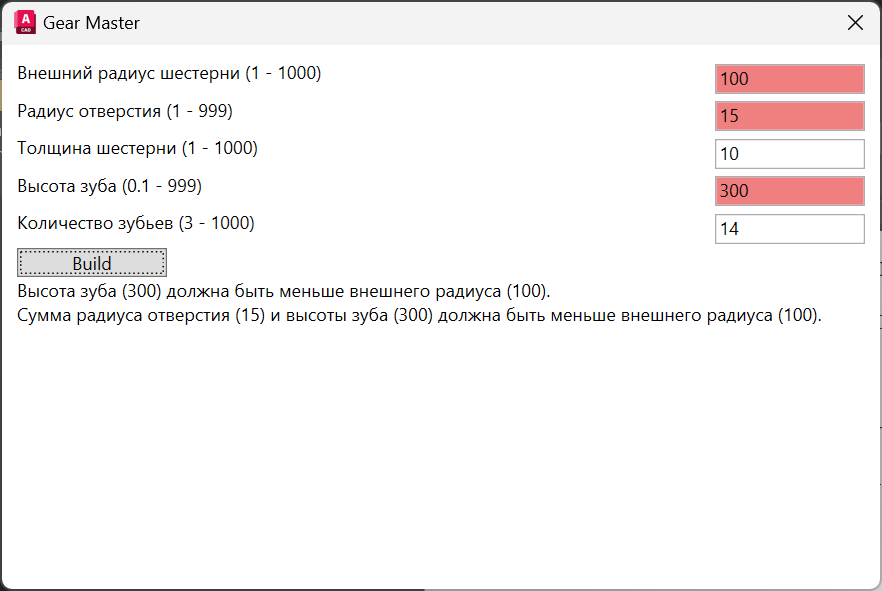


Рисунок 7.2 – Интерфейс с неверно введёнными параметрами

Корректное построение модели представлено на рисунке 7.3.

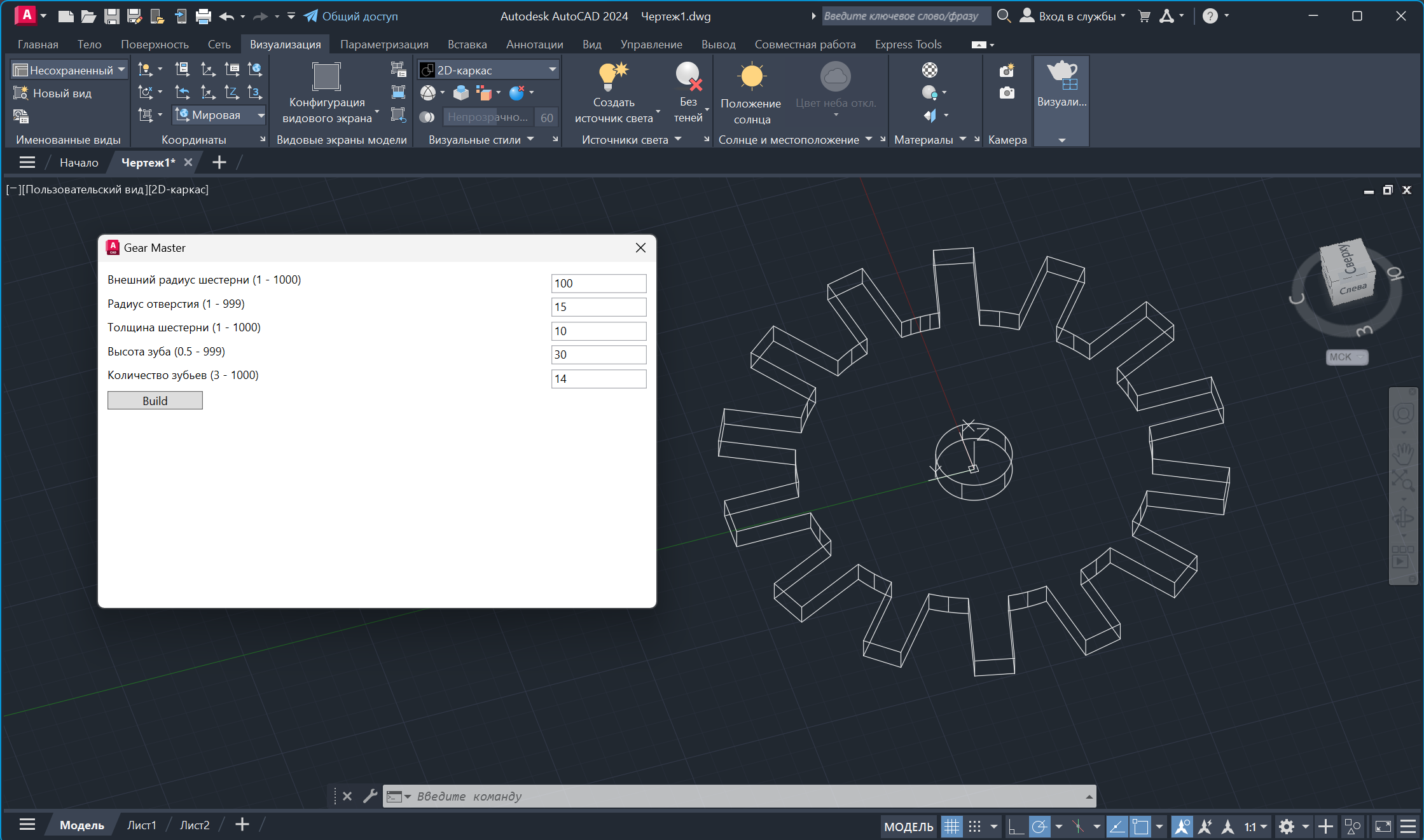


Рисунок 7.3 – Интерфейс с корректно введёнными параметрами и построенной по ним шестернёй

# 8 ТЕСТИРОВАНИЕ ПЛАГИНА

Тестирование позволяет убедиться в работоспособности программы, выявить ошибки при изменении функциональности.

## 8.1 Функциональное тестирование

Вывод различных сообщений об ошибке показан на рисунках 8.1 – 8.3.

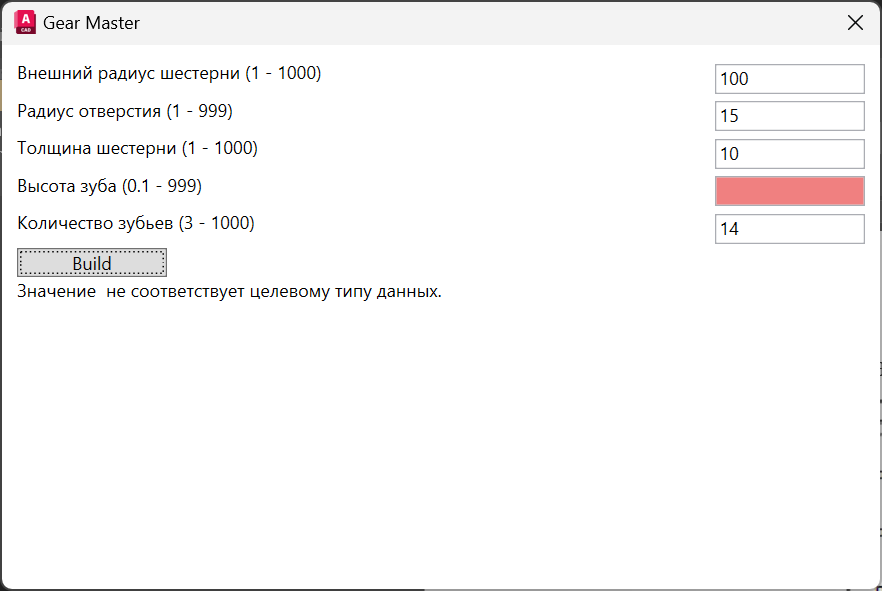


Рисунок 8.1 – Вывод сообщения при пустом текстовом поле

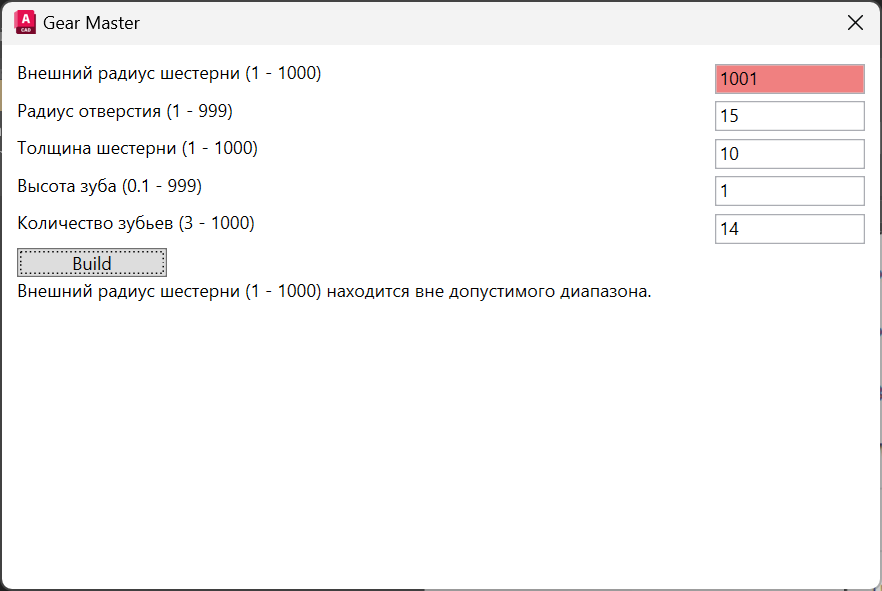


Рисунок 8.2 – Вывод сообщения при значениях, не входящих в допустимый диапазон

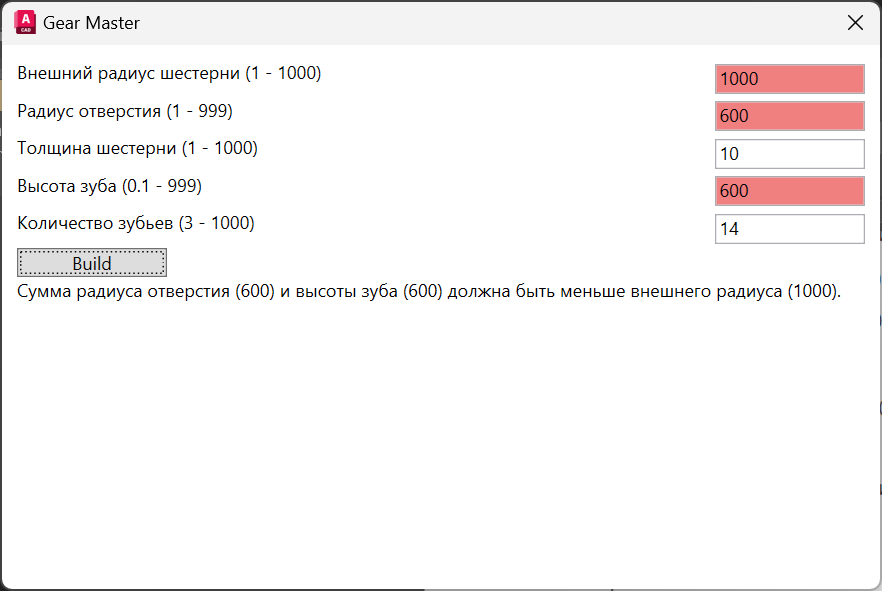


Рисунок 8.3 – Вывод сообщения при значениях, не подходящих условиям в созависимых полях

Минимальные значения:

1. Внешний радиус шестерни = 1 мм;
2. Диаметр отверстия = 1 мм;
3. Толщина шестерни = 1 мм;
4. Высота зуба = 0.5 мм;
5. Количество зубьев = 3 мм.

Максимальные значения:

1. Внешний радиус шестерни = 1000 мм;
2. Диаметр отверстия = 999 мм;
3. Толщина шестерни = 1000 мм;
4. Высота зуба = 999 мм;
5. Количество зубьев = 1000 мм.

При запуске программы значения параметров устанавливаются минимально допустимыми. На рисунках 8.4-8.5 представлены модели с минимально введёнными параметрами.

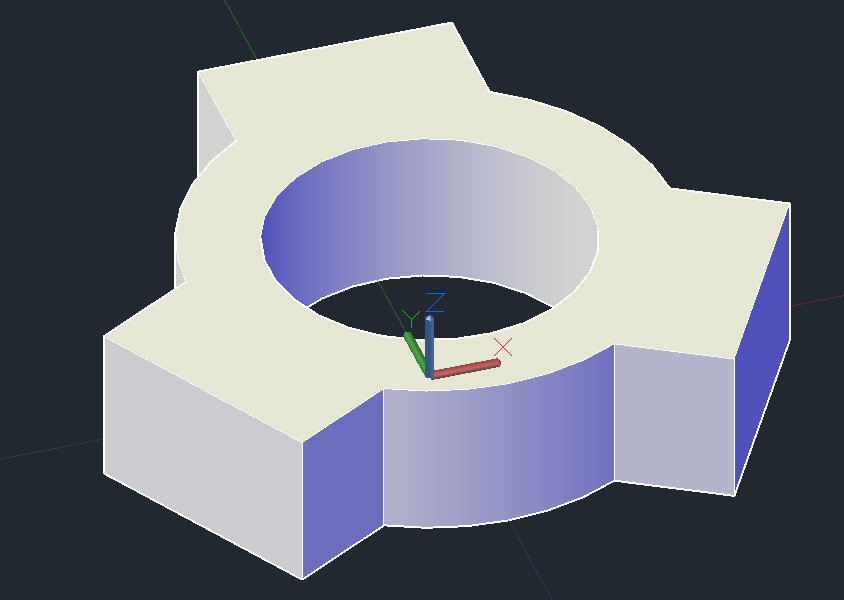


Рисунок 8.4 – Модель шестерни с минимальными параметрами

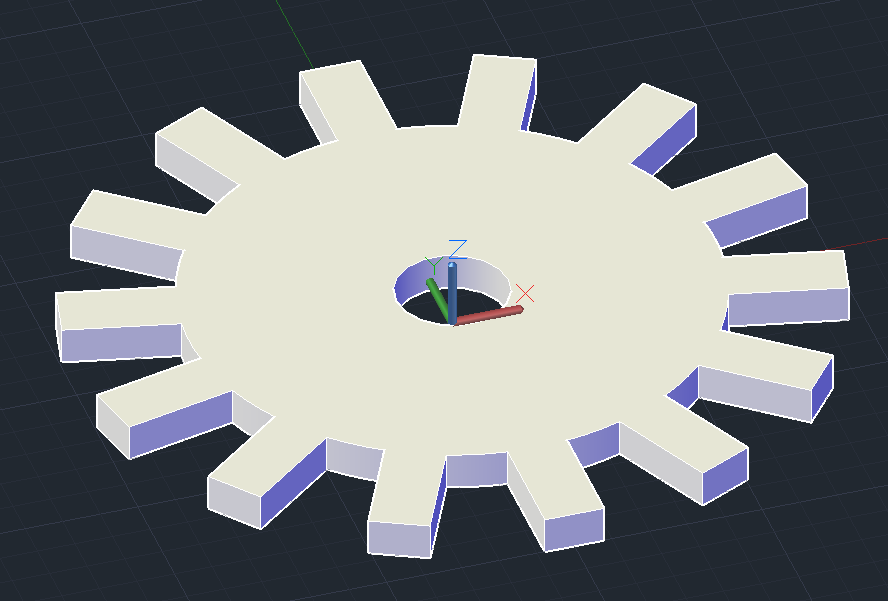


Рисунок 8.5 – Модель шестерни с параметрами по умолчанию

На рисунке 8.6 представлена модель с максимально введёнными параметрами. Данная модель слаба похожа на шестерню, потому из-за большого количества зубьев и слишком большой их высоты.

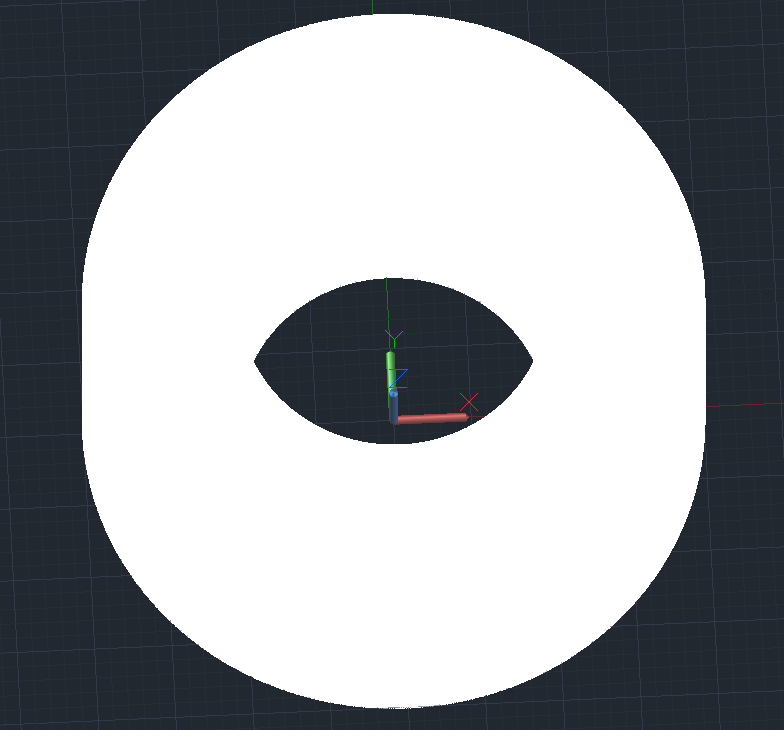


Рисунок 8.6 – Модель шестерни с максимально введёнными параметрами

## 8.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи фреймворка для модульного тестирования NUnit версии 3.13 было проведено модульное тестирование открытых свойств и методов. Были протестированы следующие классы: ParametersVM, ParameterVM, Parameter, IntParameter, DoubleParameter. Покрытие модели тестами составило 100%, что показано на рисунке 8.7.

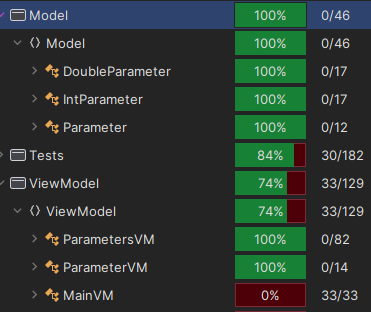


Рисунок 8.7 – Покрытие кода тестами

Тестирование класса Parameter осуществляется через тестирование его наследников, т.к. Parameter является абстрактным классом. Тесты для классов IntParameter и DoubleParameter являются идентичными по реализации и различаются только тестовыми данными (таблица 8.1).

Таблица 8.1 – Тесты классов IntParameter и DoubleParameter

|  |  |
| --- | --- |
| Название | Описание |
| AssertOnAssertCorrect\_CorrectValue | Тест валидации параметра при корректных входных данных |
| AssertOnAssertCorrect\_UnparsebleIncorrectValue | Тест валидации параметра при некорректных входных данных, которые нельзя привети к целевому типу данных |
| AssertOnAssertCorrect\_IncorrectArgumentValue | Тест валидации параметра при некорректных входных данных, которые находятся вне диапазона допустимых значений |

Тесты для класса ParametrVM представлены в таблице 8.2.

Таблица 8.2 – Тесты класса ParameterVM

|  |  |
| --- | --- |
| Название | Описание |
| AssertOnAssertCorrect\_CorrectValue | Позитивный тест валидации |
| AssertOnAssertCorrect\_IncorrectValue | Негативный тест валидации |
| AssertOnGetDescription\_CorrectValue | Позитивный тест геттера для описания параметра |
| AssertOnGetValue\_CorrectValue | Позитивный тест геттера для значения параметра |
| AssertOnParameterVM\_CorrectValue | Позитивный тест конструктора ParameterVM |
| AssertOnSetDescription\_CorrectValue | Позитивный тест сеттера описания параметра |
| AssertOnSetValue\_CorrectValue | Позитивный тест сеттера значения параметра |

Тесты класса ParametersVM представлены в таблице 8.3.

Таблица 8.3 – Тесты класса ParametersVM

|  |  |
| --- | --- |
| Название | Описание |
| AssertExportParameters\_CorrectValue | Проверяет корректность работы метода экспортирования параметров в целевых типах данных |
| AssertGetParameters\_CorrectValue | Проверяет геттер для свойства параметров |
| AssertGetParametersCorrectness\_CorrectValue | Проверяет геттер для свойства корректности параметров |
| AssertOnValidateParametersCorrect\_CorrectValue | Негативный тест на проверку валидации хранимых параметров |
| AssertOnValidateParameters\_CorrectValue | Позитивный тест на проверку валидации хранимых параметров |

## 8.3 Нагрузочное тестирование

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

* Процессор AMD Ryzen 7 5800H 3.8 GHz;
* 16 ГБ ОЗУ;
* Операционная система Windows 11 домашняя x64;
* Видеокарта AMD Radeon Vega 8 Graphics с объёмом памяти 2 ГБ.

Для нагрузочного тестирования был задан цикл построения детали. Для измерения времени был использован класс Stopwatch. Тестирование заключалось в построении 1000 шестерней со средними параметрами. На рисунке 15 и 16 показан результат данного тестирования.

Рисунок 8.8 – График зависимости времени построения от количества построенных моделей

Рисунок 8.9 – График зависимости загруженности памяти от количества построенных моделей

Тестирование длилось 37 минут, была построена 1000 моделей шестерней со средними параметрами.

Исходя из графика, представленного на рисунке 16, можно понять, что каждая последующая модель в среднем имеет большее время построения относительно предыдущей. Прослеживается зависимость линейного увеличения времени построения моделей в зависимости от общего количества построенных моделей в тесте.

Исходя из графика, представленного на рисунке 17, можно заметить линейное увеличение объёма занимаемой оперативной памяти в зависимости от количества моделей до тех пор, пока оперативная память не закончится. Однако, после заполнения её объёма, работа плагина не останавливается, потому что начинает использоваться файл подкачки, восполняющий необходимость в оперативной памяти за счёт использования постоянной памяти SSD M.2.

# Заключение

В процессе разработки приложения был создан плагин, позволяющий создавать 3D-модели шестерней в САПР Autodesk AutoCAD.

Для разработки были изучены:

* Библиотека StyleCop для форматирования кода;
* Программное обеспечение Enterprise Architect для создания UML-диаграмм и планирования архитектуры приложения;
* AutoCAD API для программного взаимодействия с САПР AutoCAD;
* Фреймворк WPF для создания оконных приложений;
* Библиотека NUnit для написания Unit-тестов.

# Список использованных источников

1. Microsoft Learn [электронный ресурс]. – URL: <https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/overview/?view=netdesktop-8.0>
2. Microsoft Learn [электронный ресурс]. – URL: https://learn.microsoft.com/ru-ru/dotnet/framework/get-started/overview
3. Приложения ObjectARX. [Электронный ресурс]: официальный сайт Autodesk AutoCAD 2024. URL: <https://help.autodesk.com/view/OARX/2024/RUS/?guid=GUID-3FF72BD0-9863-4739-8A45-B14AF1B67B06>.
4. StyleCop. [Электронный ресурс]: официальный маркетплейс Visual Studio. URL: [https://marketplace.visualstudio.com/items?itemName=ChrisDahlberg.StyleCop](https://vk.com/away.php?to=https%3A%2F%2Fmarketplace.visualstudio.com%2Fitems%3FitemName%3DChrisDahlberg.StyleCop&cc_key=).
5. Autodesk Appstore [электронный ресурс]. – URL: <https://apps.autodesk.com/ACD/ru/Detail/Index?id=5237307566585001661&appLang=en&os=Win32_64>
6. Обучающая статья “Autodesk Inventor. Построение зубчатой передачи”. [Электронный ресурс]. Режим доступа: свободный (дата обращения 09.10.23), <https://www.pointcad.ru/novosti/autodesk-inventor.-postroenie-zubchatoj-peredachi>
7. Официальный сайт САПР Autodesk Fusion 360. [Электронный ресурс]. Режим доступа: свободный (дата обращения 09.10.23), <https://www.autodesk.com/products/fusion-360/overview?term=1&tab=subscription>
8. Официальный сайт САПР Kompas-3D. [Электронный ресурс]. Режим доступа: свободный (дата обращения 09.10.23), <https://kompas.ru/kompas-3d/about/>