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 "ВАЛ-ШЕСТЕРНЯ" ДЛЯ САПР Компас-3D

Пояснительная записка по дисциплине

«ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_А.С.Избышев

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

Руководитель:

к.т.н., доцент каф. КСУП

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

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

Томск 2023

**Оглавление**

[1 Введение 3](#_Toc124456400)

[2 Постановка и анализ задачи 4](#_Toc124456401)

[3 Описание предмета проектирования 5](#_Toc124456402)

[5 Назначение плагина 8](#_Toc124456403)

[6 Обзор аналогов 9](#_Toc124456404)

[7 Описание реализации 10](#_Toc124456405)

[8 Диаграмма классов 11](#_Toc124456406)

[9 Описание программы для пользователя 15](#_Toc124456407)

[10 Функциональное тестирование 17](#_Toc124456408)

[11 Модульное тестирование 19](#_Toc124456409)

[12 Нагрузочное тестирование 20](#_Toc124456410)

[13 Заключение 21](#_Toc124456411)

[Список литературы 22](#_Toc124456412)

# 1 Введение

Автоматизация проектирования 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 проектированием 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.

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей (в том числе, деталей, формируемых из листового материала путём его гибки) и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы [1]. Параметрическая технология позволяет быстро получать модели типовых изделий на основе проектированного ранее прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства.

КОМПАС-3D широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение (транспортное, сельскохозяйственное, энергетическое, нефтегазовое, химическое и т.д.), приборостроение, авиастроение, судостроение, станкостроение, вагоностроение, металлургия, промышленное и гражданское строительство, товары народного потребления и т. д.

# 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 вал-шестерни. Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры вал-шестерни, такие как:

* диаметр профиля шестерни ***D1***;
* ширина шестерни ***W1***;
* диаметр соединяющего цилиндра ***D2***;
* диаметр основного цилиндра ***D3***;
* диаметр вращательного цилиндра ***D4***;
* длина вращательного цилиндра ***L1***;
* угол вращения шестерни ***R1****.*

# 3 Описание предмета проектирования

Вал-шестерня – комбинированный вид запчасти, состоящей непосредственно из самого вала и шестерни. Целевой задачей данной конструкции является передача крутящего усилия с одного вала на другой и поддержание шкивов, катков и прочих элементов в приводных механизмах, редукторах. Коническая вал-шестерня – это шестерня, в которой оси двух валов пересекаются, а зубчатая поверхность самой шестерни имеет коническую форму. Поверхность конических вал-шестерней представляет собой конус [2].

Изображение объекта представлено на рисунках 3.1-3.2:



Рисунок 3.1 – Чертёж конической вал-шестерни слева

Изменяемые параметры для плагина:

**W1** – ширина шестерни (70-200 мм);

**D1** – диаметр профиля шестерни (70-120 мм);

**D2** – диаметр соединяющего цилиндра (85-105 мм);

**D3** – диаметр основного цилиндра (80-100 мм);

**D4** – диаметр вращательного цилиндра (20-55 мм);

**L1** – длина вращательного цилиндра (30-75 мм);

**R1** – угол вращения шестерни (0-40 градусов).

Ширина шестерни W1 и диаметр конического профиля D1 ограничены так, чтобы сохранять форму конуса и не допускать искажений модели.

Диаметр D2 и D3 не должны превышать выхода за профиль шестерни, а также зависимы между собой и второе значение должно быть больше первого на 5 см для правильного соединения между ними.

Диаметр D4 ограничен значением 20 см и предельно допустимым для перехода к следующему элементу значением 55 см. Длина L1 имеет границы в пределах нормы.

Также для вращательного цилиндра установлена зависимость – длина L1 должна быть больше диаметра D4 как минимум на 10 мм.

Угол вращения R1 ограничен значениями от 0 до 40 градусов.

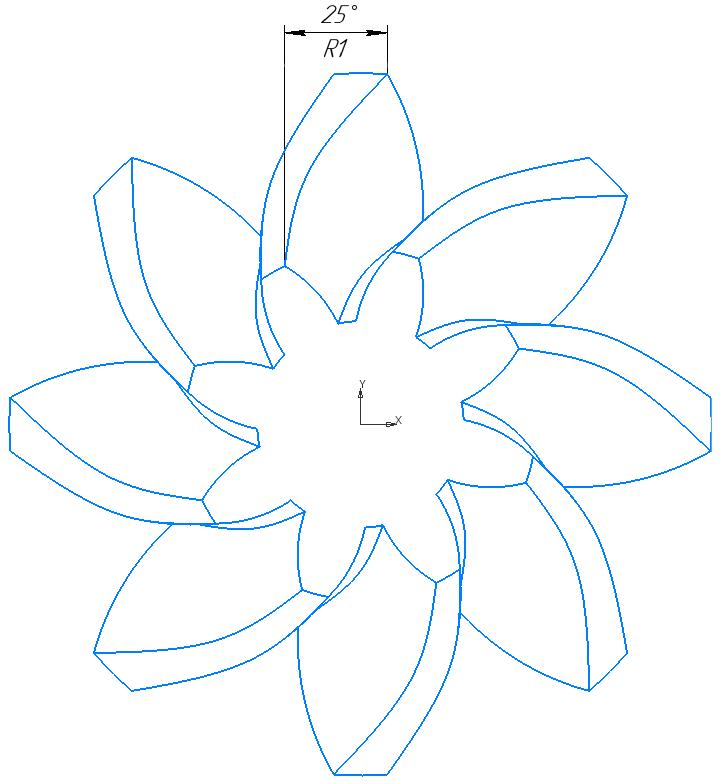


Рисунок 3.2 – Чертёж конической вал-шестерни спереди

**4 Выбор инструментов и средств реализации**

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2 [3], библиотеки для Kompas 3D [4].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [5] версии 3.13.3.

Для реализации пользовательского интерфейса использовалась система для построения настольных приложений WindowsForms [6].

# 5 Назначение плагина

Программа предназначена для автоматизации моделирования детали «Коническая вал-шестерня».

Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях с помощью цветового выделения и всплывающих подсказок.

При запуске моделирования с некорректными значениями программа выводит сообщение об ошибке и отменяет построение модели.

При правильно введенных значениях результатом работы программы будет созданная по ним модель конической вал-шестерни

# 6 Обзор аналогов

Онлайн 3D – конфигуратор шестерней Gear Generator[7].

Gear Generator – это онлайн генератор для создания различных контуров шестерней, которые можно сконвертировать в 3D-модель в формате DXF, либо использовать как 2D-скетч в формате SVG для дальнейшего выдавливания в Inventor, Solidworks, Fusion 360 и других программах, поддерживающих эту функцию.

Генератор включает в себя всевозможные настраиваемые параметры: центр шестерни, диаметр, количество зубьев, профиль зуба и их направление. Также можно создавать внутренние шестерни.

Интерфейс программы показан на рисунке 6.1:



Рисунок 6.1 –Интерфейс Gear Generator

# 7 Описание реализации

Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях.

Если пользователем введены недопустимые значения параметров, то построение модели не начнётся.

При правильно введенных значениях результатом работы программы будет созданная по ним модель конической вал-шестерни.

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии) 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 [8].

При использовании UML были простроена диаграмма классов.

# 8 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.

На рисунке 8.1 представлена диаграмма классов.

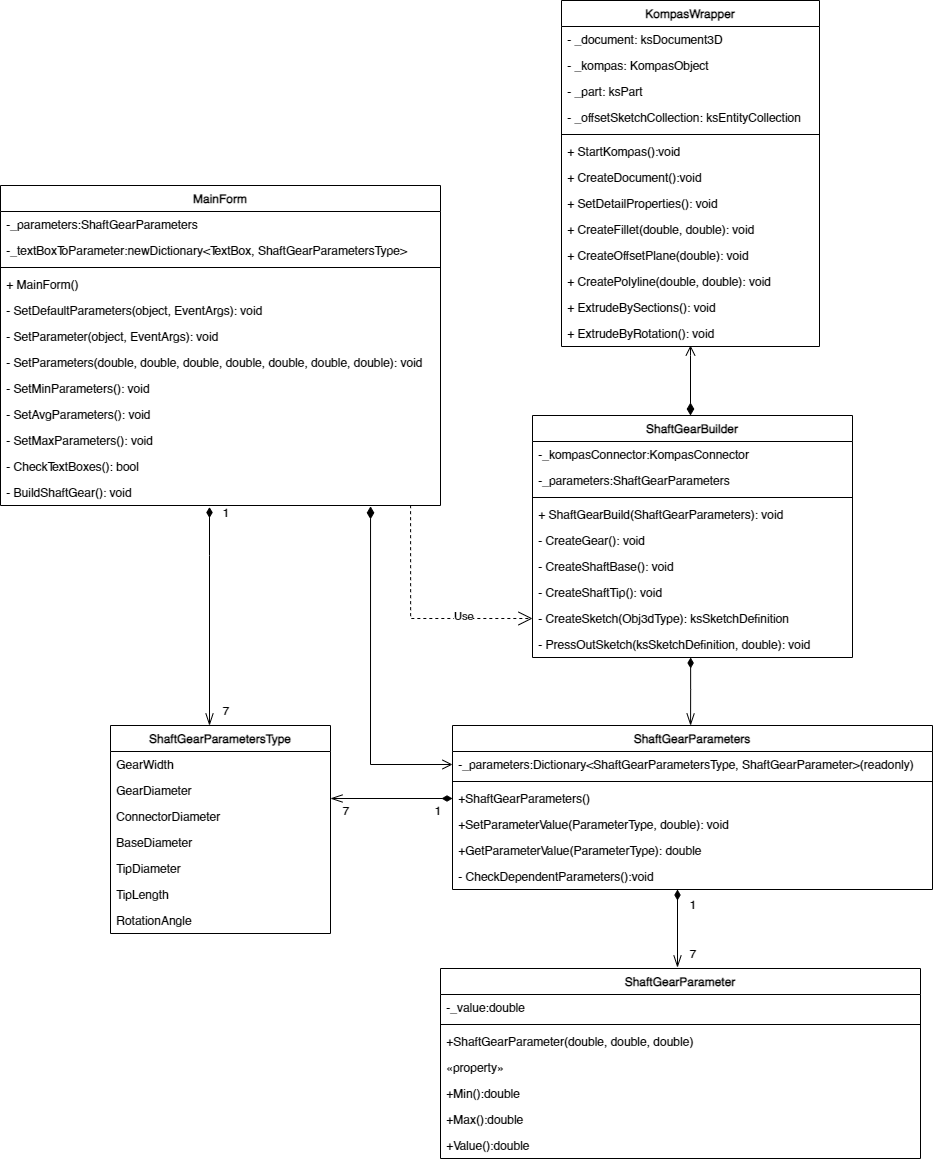


Рисунок 8.1 – Диаграмма классов UML

Далее в таблицах 8.1 – 8.4 представлено описание классов.

Таблица 8.1 – Описание полей, методов, сущностей класса «MainForm»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Хранит в себе набор методов для построения вал-шестерни |
| \_textBoxToParameter |  | Хранит в себе набор TextBox и соответствующий ему тип параметра из перечисления «ShaftGearParametersType» |
| SetParameter(object, EventArgs) | void | Устанавливает значение параметра |
| SetMinParameters() | void | Устанавливает минимальное значение всех параметров |
| SetMaxParameters() | void | Устанавливает максимальное значение всех параметров |
| SetAvgParameters() | void | Устанавливает среднее значение всех параметров |
| BuildShaftGear() | void | Строит коническую вал-шестерню по заданным параметрам |

Таблица 8.2 – Описание полей, методов, сущностей класса «ShaftGearParameters»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Хранит данные о каждом параметре модели из перечисления «ShaftGearParametersType» |
| ShaftGearParameters() |  | Конструктор для создания экземпляра класса |
| SetParameterValue(ParameterType, double) | void | Устанавливает значение определённого параметра |
| GetParameterValue(ParameterType) | double | Возвращает значение определённого параметра |

Таблица 8.3 – Описание полей, методов, сущностей класса «ShaftGearParameter»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_value |  | Хранит текущее значение |
| ShaftGearParameter(double, double, double) |  | Конструктор для создания экземпляра класса |
| Max() | double | Возвращает максимальное допустимое значение параметра |
| Min() | double | Возвращает минимальное допустимое значение параметра |
| Value() | double | Возвращает текущее значение параметра. Задает новое значение параметра |

Таблица 8.4 – Описание полей, методов, сущностей класса «ShaftGearBuilder»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| kompasConnector |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| \_parameters |  | Хранит данные о каждом параметре модели из перечисления «ShaftGearParameter» |
| ShaftGearBuild  (ShaftGearParameters) | void | Построение шестерни по заданным параметрам |
| CreateGear() | void | Построение основы шестерни |
| CreateShaftBase() | void | Построение основы вала |
| CreateShaftTip() | void | Построение вращательного цилиндра вала |
| PressOutSketch(ksSketchDefinition, double) | void | Выдавливает эскиз |

# 9 Описание программы для пользователя

Пользовательский интерфейс (UI) – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы [9].

Макет пользовательского интерфейса представляет собой форму для ввода параметров конической вал-шестерни (рисунок 9.1). Построение модели осуществляется путем нажатия на кнопку «Build Model».

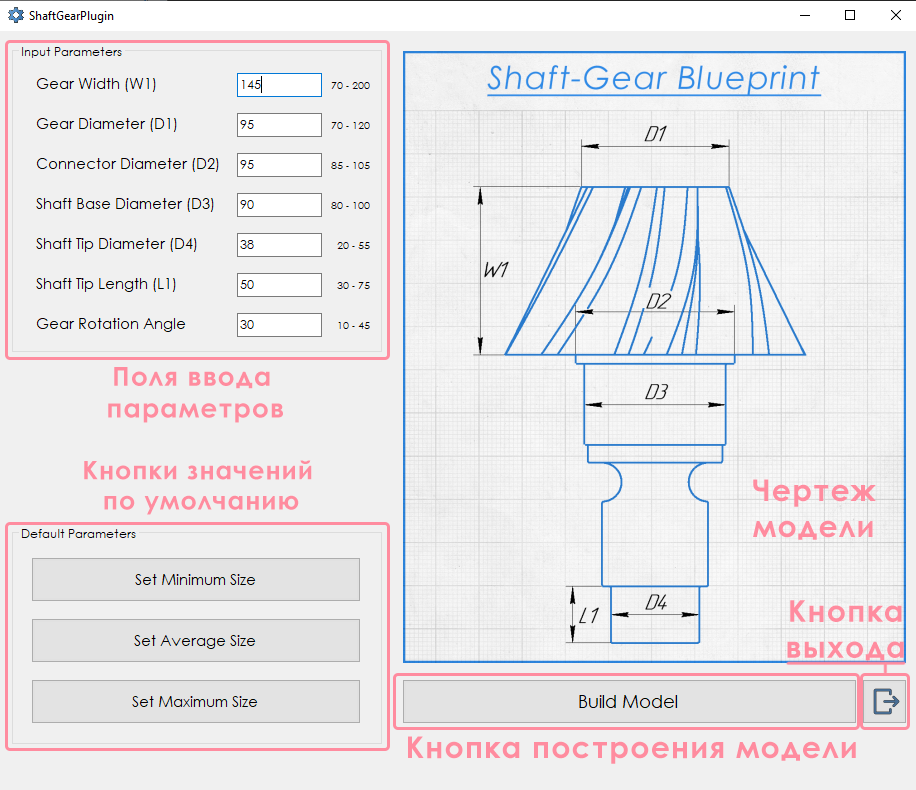


Рисунок 9.1 – Макет пользовательского интерфейса

С помощью данного окна пользователь может изменять параметры будущей 3D-модели конической вал-шестерни.

Рядом с полями ввода находятся название компонента, за который поле отвечает и его обозначение на чертеже. Также указаны допустимые значения корректных размеров для параметров модели.

На панели «Default Parameters» созданы 3 кнопки установки значений по умолчанию. При нажатии на кнопку «Set Minimum Size» будет создана 3D-модель с минимальными допустимыми размерами. При нажатии на кнопку «Set Average Size» будет создана 3D-модель со средними значениями размеров. При нажатии на кнопку «Set Maximum Size» будет создана 3D модель с максимальными корректными размерами.

Чертёж модели справа необходим для лучшего понимания расположения вводимых размеров – обозначения размеров соответствуют подписям полей ввода значений.

При вводе недопустимых значений рядом с полем ввода появляется предупреждение, при наведении на которое можно увидеть подробную информацию о причине его появления. На рисунке 9.2 продемонстрированы 2 предупреждения: первое из-за ввода слишком большого значения, выходящего за пределы допустимых; второе из-за нарушения зависимости параметров друг от друга.

Построение модели невозможно начать, пока все поля не будут корректно заполнены.

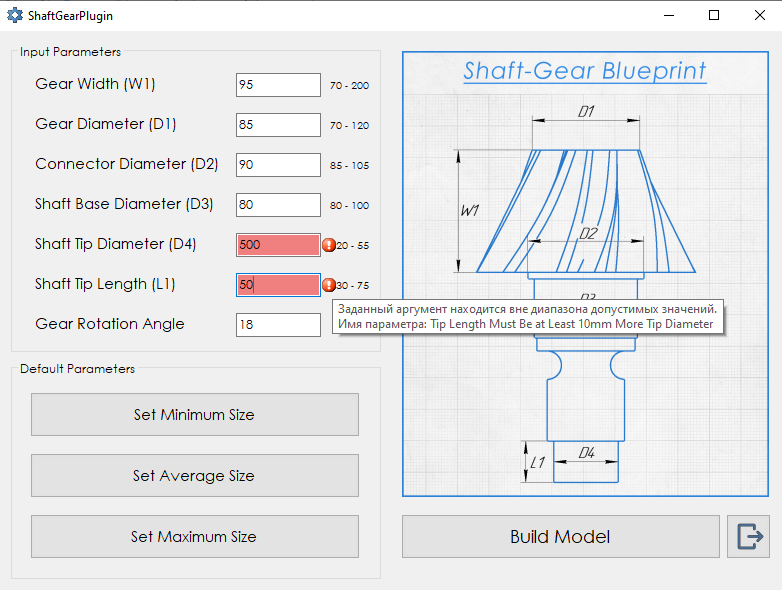


Рисунок 9.2 – Реализация предупреждений о недопустимых значениях

# 10 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Вал-шестерня», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведено тестирование максимальных и минимальных параметров модели.

На рисунке 10.1 представлена проверка размеров модели с минимальным введенными параметрами в САПР Kompas 3D. Параметры были установлены автоматически нажатием на кнопку “Set Minimum Size”.

Минимальные параметры вал-шестерни:

* Диаметр профиля шестерни – 70 мм;
* Ширина шестерни – 70 мм;
* Диаметр соединяющего цилиндра – 85 мм;
* Диаметр основного цилиндра – 80 мм;
* Диаметр вращательного цилиндра – 20 мм;
* Длина вращательного цилиндра – 30 мм;
* Угол вращения шестерни – 0 градусов.

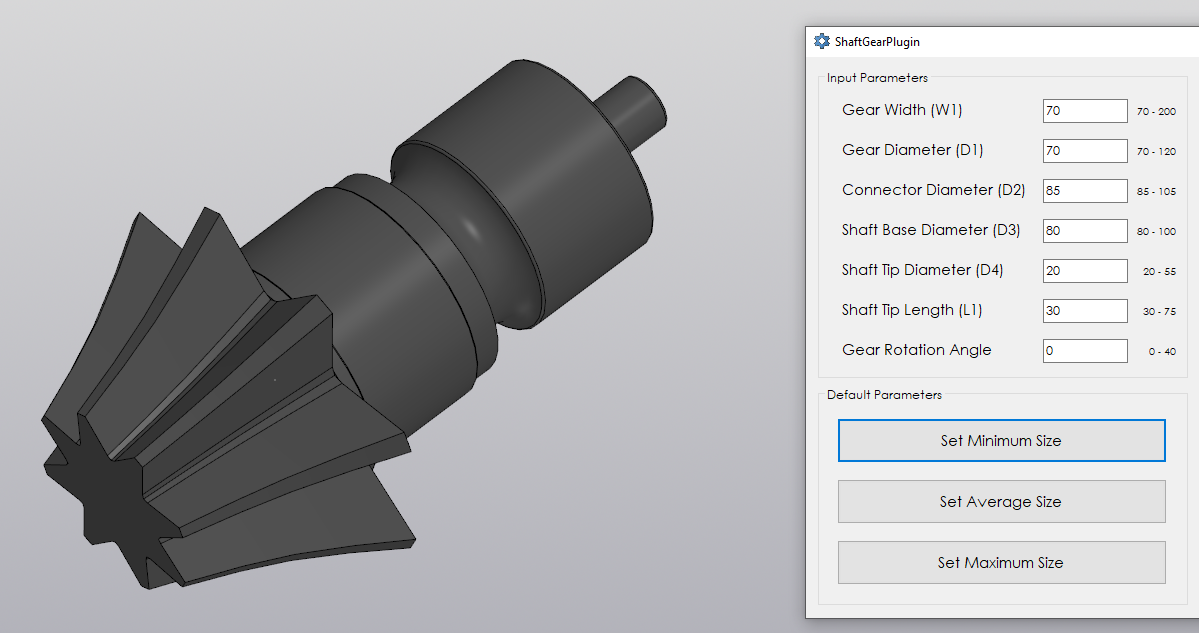


Рисунок 10.1 – Модель с минимальными введенными параметрами

Ниже на рисунке 10.2 представлена проверка размеров модели с максимальными введенными параметрами в САПР Kompas 3D. Параметры были установлены автоматически нажатием на кнопку “Set Maximum Size”.

Максимальные параметры вал-шестерни:

* Диаметр профиля шестерни – 200 мм;
* Ширина шестерни – 120 мм;
* Диаметр соединяющего цилиндра – 105 мм;
* Диаметр основного цилиндра – 100 мм;
* Диаметр вращательного цилиндра – 55 мм;
* Длина вращательного цилиндра – 75 мм;
* Угол вращения шестерни – 40 градусов.

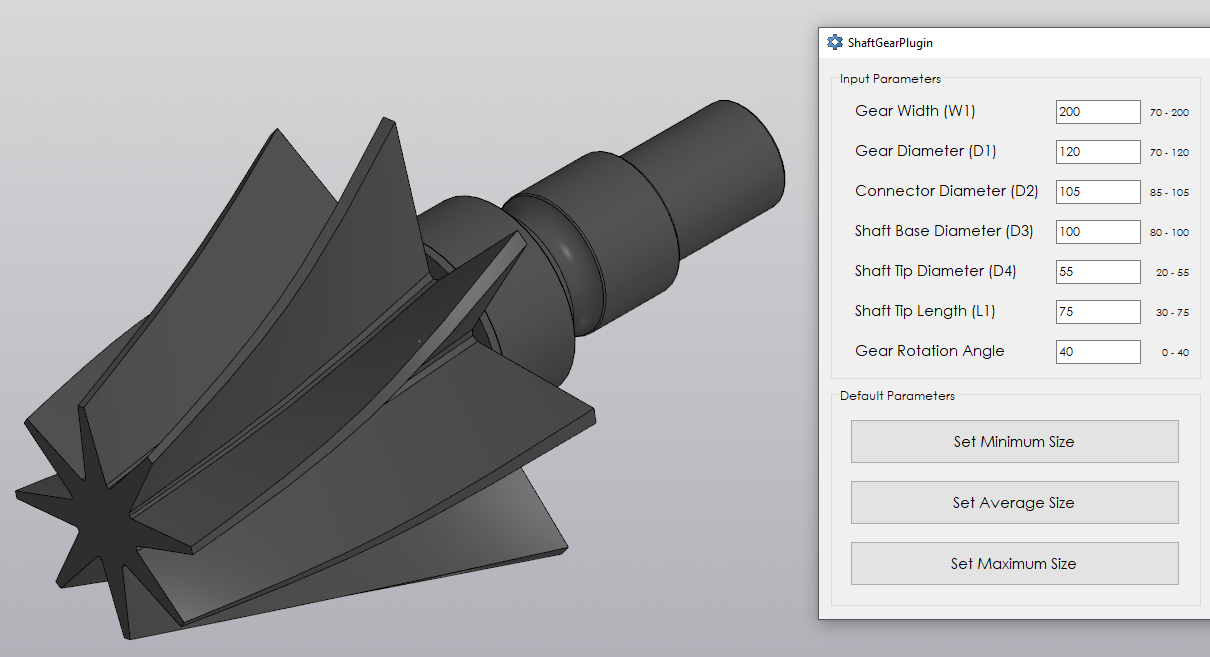


Рисунок 10.2 – Модель с максимальными введенными параметрами

# 11 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование [10], проверялись открытые поля и методы. На рисунке 11.1 представлено тестирование библиотеки классов бизнес логики проекта. Степень покрытия модульными юнит-тестами – 100%. Информация о покрытии отображена на рисунке 11.2.

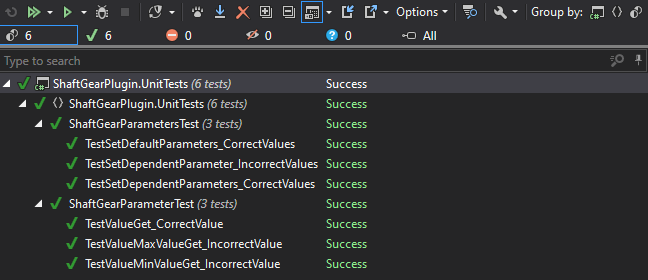


Рисунок 11.1 – Тестирование бизнес логики

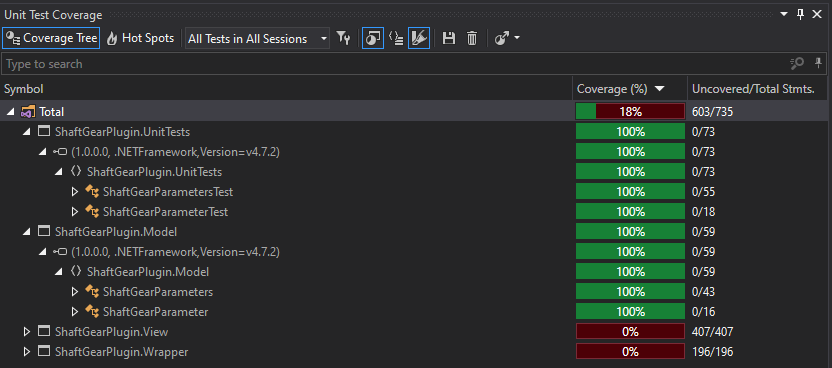


Рисунок 11.2 – Покрытие бизнес логики юнит-тестами

Перечень тестов для бизнес-логики Model с их описанием представлен в таблице 11.1.

# 12 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование.

Нагрузочное тестирование – это 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) [11].

Тестирование производилось на ПК со следующей конфигурацией:

* процессор AMD Ryzen 7 3700Х 3.90 GHz;
* оперативная память 12 ГБ;
* графический процессор объемом памяти 6 ГБ.

На рисунке 12.1 показан график зависимости количества используемой оперативной памяти в мегабайтах от количества построенных деталей. Всего за процесс тестирования было построено 58 деталей.

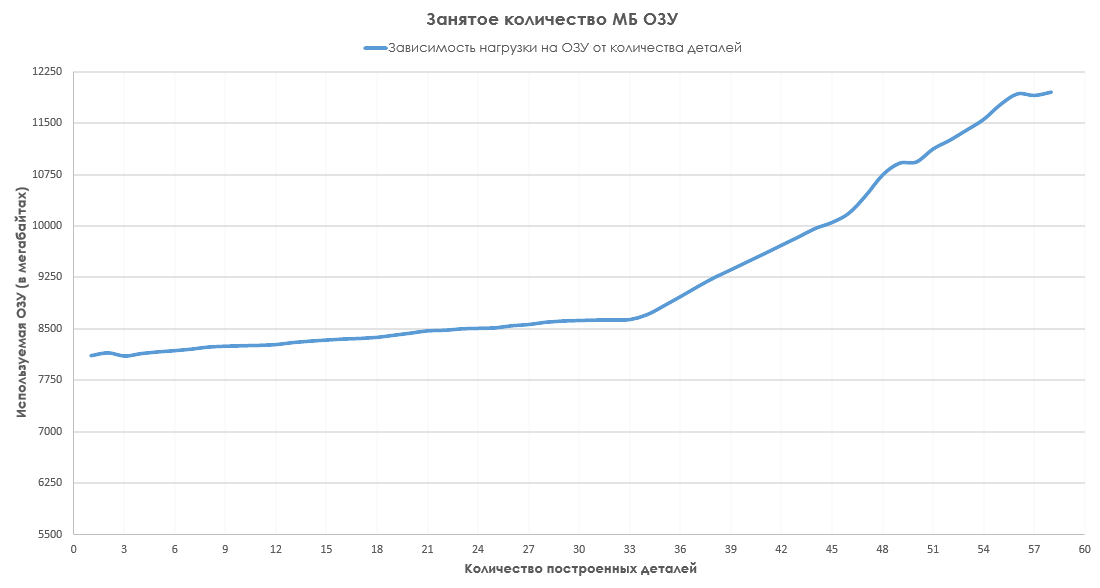


Рисунок 12.1 – Зависимость загрузки ОЗУ от количества деталей

Из графика видно, что зависимость является линейной. На графики имеются просадки, то есть обращение к оперативной памяти периодически останавливается, и система использует файл подкачки

На рисунке 12.2 показана зависимость пройденного времени тестирования и количества построенных деталей.

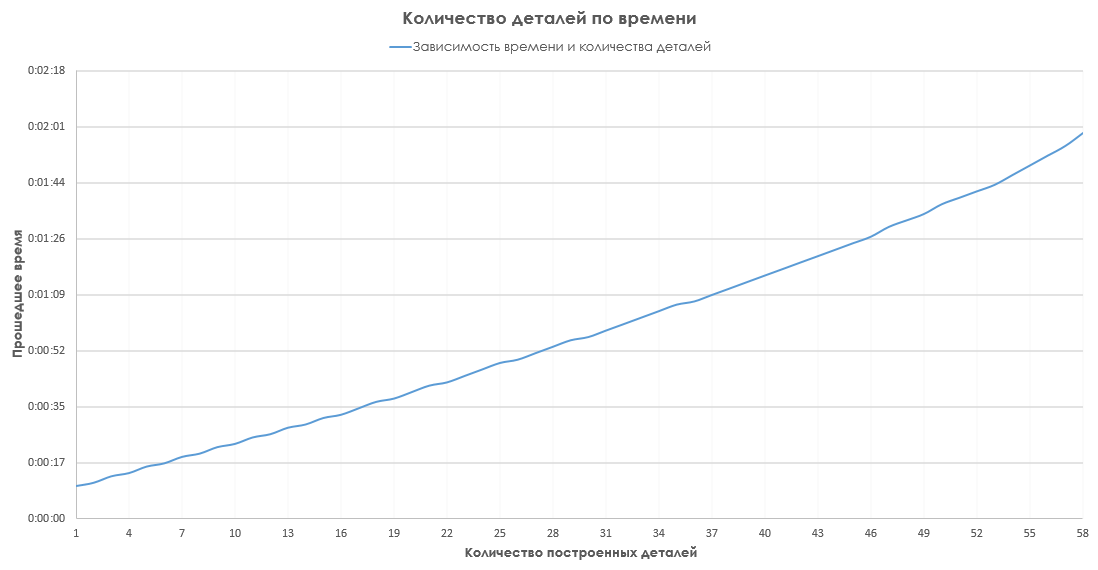


Рисунок 12.2 – Зависимость количества деталей и прошедшего времени

Из графика видно, что зависимость величины пройденного времени и количества построенных деталей является линейной.

На рисунке 12.3 показана зависимость потраченного на построение каждой детали времени. При большой загрузке оперативной памяти и центрального процессора скорость построения снижается.

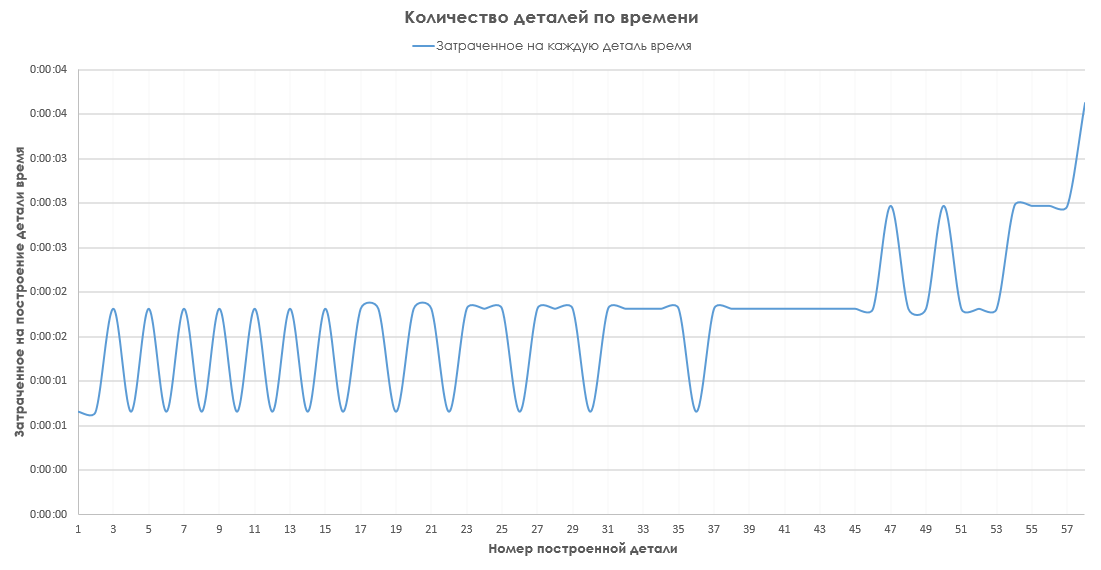


Рисунок 12.3 – Зависимость затраченного на каждую деталь времени

# 13 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 «Коническая вал-шестерня» в САПР Kompas 3D, и проведено функциональное и нагрузочное тестирование плагина.

# Список литературы

1. Компас (САПР) — Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Компас_(САПР)> (дата обращения 02.11.2022).

2. Коническая вал-шестерня — Википедия. [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/Bevel_gear> (дата обращения 18.11.2022).

3. Платформа .NET Framework. [Электронный ресурс]. — Режим доступа: <https://dotnet.microsoft.com/en-us/download/dotnet-framework> (дата обращения 10.01.2023).

4. API — Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 08.11.2022).

5. NUnit [Электронный ресурс]. – Режим доступа: <https://nunit.org/> (дата обращения: 27.12.2022).

6. Документация по Windows Forms для .NET 6. [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/dotnet/desktop/winforms> (дата обращения: 06.01.2023).

7. Генератор шестерни онлайн — Gear Generator [Электронный ресурс]. — Режим доступа: <https://geargenerator.com/beta> (дата обращения 18.11.2022).

8. UML. [Электронный ресурс]. – Режим доступа: <http://www.uml.org/> (дата обращения 14.11.2022).

9. Интерфейс пользователя [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Интерфейс\_пользователя (дата обращения: 17.12.2022)

10. Модульное тестирование [Электронный ресурс] – Режим доступа: https://logrocon.ru/news/unit\_testing (дата обращения: 30.11.2022)

11. Нагрузочное тестирование [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Нагрузочное\_тестирование (дата обращения: 29.12.2022)