

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)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

на разработку плагина моделирования цепного колеса  
для системы «AutoCAD»

Выполнил:

Студент группы 588-2

\_\_\_\_\_ / Рыжков Д.А.

«\_\_» \_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_ / Калентьев А. А.

«\_\_» \_\_\_\_\_ 2022 г.

Томск 2022

## Оглавление

<b>1 Введение.....</b>	<b>3</b>
<b>2 Описание САПР .....</b>	<b>4</b>
2.1 Описание AutoCAD 2022 .....	4
2.2 Object ARX SDK.....	4
2.3 AutoCAD .NET API.....	5
2.4 Обзор аналогов .....	9
2.4.1 AutoCAD Mechanical Toolset .....	9
2.4.3 SelfCAD .....	10
2.4.1 Autodesk App Store .....	11
<b>3 Описание предмета проектирования.....</b>	<b>12</b>
<b>4 Проект системы.....</b>	<b>14</b>
4.1 Описание технических и функциональных аспектов системы .....	14
4.2 Диаграмма классов.....	15
4.3 Макет пользовательского интерфейса .....	17
<b>5 Тестирование программы .....</b>	<b>20</b>
5.1 Функциональное тестирование.....	20
5.2 Модульное тестирование .....	22
5.3 Нагрузочное тестирование .....	23
5.3.1 Нагрузочное тестирование с параметрами по умолчанию.....	23
5.4 Выводы нагрузочного тестирования .....	26
<b>Вывод .....</b>	<b>27</b>
<b>Список литературы .....</b>	<b>28</b>
<b>Приложение А.....</b>	<b>30</b>

## **1 Введение**

Область применения систем автоматизированного проектирования (САПР) охватывает сегодня самые различные виды деятельности человека – от расстановки мебели в квартире до проектирования и изготовления интегральных микросхем и современной космической техники. Каждая категория задач технического черчения предъявляет к этим продуктам свои требования, однако наибольшее распространение они получили в машиностроении и архитектуре [1].

Средства автоматизированного проектирования имеют своей задачей повышение эффективности труда инженеров. При создании любой технической системы необходимо стремиться к экономии трех категорий трудозатрат: прошлого, или овеществлённого, труда; настоящего, или живого, труда; будущего труда, связанного с развитием системы. Основной целью создания систем компьютеризации инженерной деятельности является экономия живого труда проектировщиков, конструкторов, технологов, инженеров-менеджеров для повышения эффективности процесса проектирования и планирования, а также для улучшения качества результатов этой деятельности [2].

Целью же данной работы является разработка плагина, позволяющего строить модель ракеты для САПР AutoCAD.

## **2 Описание САПР**

### **2.1 Описание AutoCAD 2022**

AutoCAD — двух- и трёхмерная система автоматизированного проектирования и черчения, разработанная компанией Autodesk. AutoCAD и специализированные приложения на его основе нашли широкое применение в машиностроении, строительстве, архитектуре и других отраслях промышленности [10].

В области двумерного проектирования AutoCAD по-прежнему позволяет использовать элементарные графические примитивы для получения более сложных объектов. Кроме того, программа предоставляет весьма обширные возможности работы со слоями и аннотативными объектами (размерами, текстом, обозначениями). Использование механизма внешних ссылок (XRef) позволяет разбивать чертёж на составные файлы, за которые ответственны различные разработчики, а динамические блоки расширяют возможности автоматизации 2D-проектирования обычным пользователем без использования программирования.

AutoCAD включает в себя полный набор инструментов для комплексного трёхмерного моделирования (поддерживается твердотельное, поверхностное и полигональное моделирование). AutoCAD позволяет получить высококачественную визуализацию моделей с помощью системы рендеринга mental ray. Также в программе реализовано управление трёхмерной печатью (результат моделирования можно отправить на 3D-принтер) и поддержка облаков точек (позволяет работать с результатами 3D-сканирования) [3].

### **2.2 Object ARX SDK**

Среда программирования ObjectARX используется для адаптации и расширения функциональных возможностей AutoCAD и продуктов на его основе. Она обеспечивает непосредственный доступ к структурам базы

данных AutoCAD, графической системе и определениям встроенных команд [4].

### 2.3 AutoCAD .NET API

В состав ObjectARX SDK входит также управляемый API, который часто называют AutoCAD .NET API. Для адаптации и расширения функциональных возможностей AutoCAD и продуктов на его основе может применяться любой язык программирования, поддерживающий .NET, к примеру, C# или VB. Обеспечивается непосредственный доступ к структурам базы данных AutoCAD, определениям встроенных команд и другим внутренним программным элементам [4].

Принцип создания и работы плагина для AutoCAD:

1. Подключение всех необходимых библиотек, находящихся в ObjectARX;
2. Написание кода программы для реализации плагина;
3. Компилируется файл с расширением .dll;
4. Полученный файл загружается в AutoCAD, после чего необходимо вызвать команду созданного плагина.

Основные пространства имен, используемые при создании плагина:

- **Autodesk.AutoCAD.ApplicationServices** – позволяет получить доступ к приложению AutoCAD;
- **Autodesk.AutoCAD.EditorInput** – позволяет получить доступ к редактору AutoCAD;
- **Autodesk.AutoCAD.DatabaseServices** – дает доступ к базе данных и сущностям AutoCAD;
- **Autodesk.AutoCAD.Runtime** – отвечает за регистрацию команд.

В таблице 2.1 представлены свойства и методы интерфейсов, которые использовались при разработке плагина.

Таблица 2.1 – Свойства и методы интерфейсов и классов

Название	Возвращаемый тип	Описание
<b>Application</b>		
DocumentManager	DocumentCollection	Получает доступ к объекту DocumentManager.
ShowModelessDialog (Form)	bool?	Используется для отображения формы WinForms
<b>Transaction</b>		
Commit()	void	фиксирует изменения, внесенные во все объекты DBOject, открытые во время Транзакции.
Abort()	void	Прерывает транзакцию.
GetObject(ObjectId, DatabaseServices.OpenMod)	DBObject	Получение объекта по его идентификатору.
<b>Document</b>		
Database	Database	Обертывает функцию AcApDocument.database() ObjectARX, которая возвращает объект базы данных (базу данных), используемый этим документом
LockDocument()	void	Блокирует документ
<b>Solid3d</b>		
createFrustum(double, double, double, double)	ErrorStatus	Этот метод используется для создания цилиндра или конуса с центром мирового происхождения вокруг его диаметра и расположен на половине высоты

Продолжение таблицы 2.1

Название	Возвращаемый тип	Описание
createWedge(double, double, double)	ErrorStatus	Этот метод используется для создания твердого тела клина с центром в начале координат WCS
extrude(double, double)	ErrorStatus	Создает твердое тело путем выдавливания области, расстояния по высоте с углом конусности конуса
BooleanOperation(BooleanOperationType, Solid3d)	void	Выполняет логическую операцию между этим твердым телом и твердым телом. Возможными типами операций являются: объединение, пересечение, вычитание
<b>DocumentCollection</b>		
MdiActiveDocument	Document	Получает доступ к текущему открытому документу
<b>ObjectId</b>		
IsNull()	bool	Определяет, имеет ли идентификатор объекта нулевое значение.
<b>TransactionManager</b>		
StartTransaction()	Transaction	Начинает новую транзакцию
<b>BlockTableRecord</b>		
Name()	string	Возвращает имя записи в таблицы символов.
AppendEntity(Entity)	ObjectId	Добавляет объект в базу данных и в BlockTableRecord.

Окончание таблицы 2.1

Название	Возвращаемый тип	Описание
<b>Entity</b>		
TransformBy(Matrix3d)	void	Эта функция предоставляет средство, с помощью которого приложения AutoCAD и ObjectARX могут запросить объект применить матрицу преобразования к себе.
SetDatabaseDefaults()	void	Эта функция задает сущностям: цвет, слой, тип линии, шкала типов линий, видимость, название стиля сюжета, вес линии значения по умолчанию для базы данных, в которой в настоящее время находится объект
<b>Database</b>		
BlockTableId()	ObjectId	Возвращает идентификатор объекта BlockTable базы данных.
TransactionManager()	TransactionManager	Обращается к TransactionManager для базы данных.



## 2.4 Обзор аналогов

### 2.4.1 AutoCAD Mechanical Toolset

Самый лаконичный способ создания звёздочки является официальным: создать цепное колесо с помощью встроенного инструмента «вставка звёздочки». Однако, данный способ доступен только на специальной версии AutoCAD (AutoCAD Mechanical Toolset), что создаёт заметные ограничения для использования этого варианта [5].

### 2.4.2 Gear Generator

Помимо официального инструмента AutoCAD существует онлайн редактор для автоматизированного построения аналогичного объекта — шестерёнок. На сайте «geargenerator.com» можно построить несколько видов шестерней (внутренняя, наружная) и задать параметры: количество зубьев, шаг зуба, угол давления, диаметр. И скачать созданную схему в нужном формате (DXF, SVG) [11].

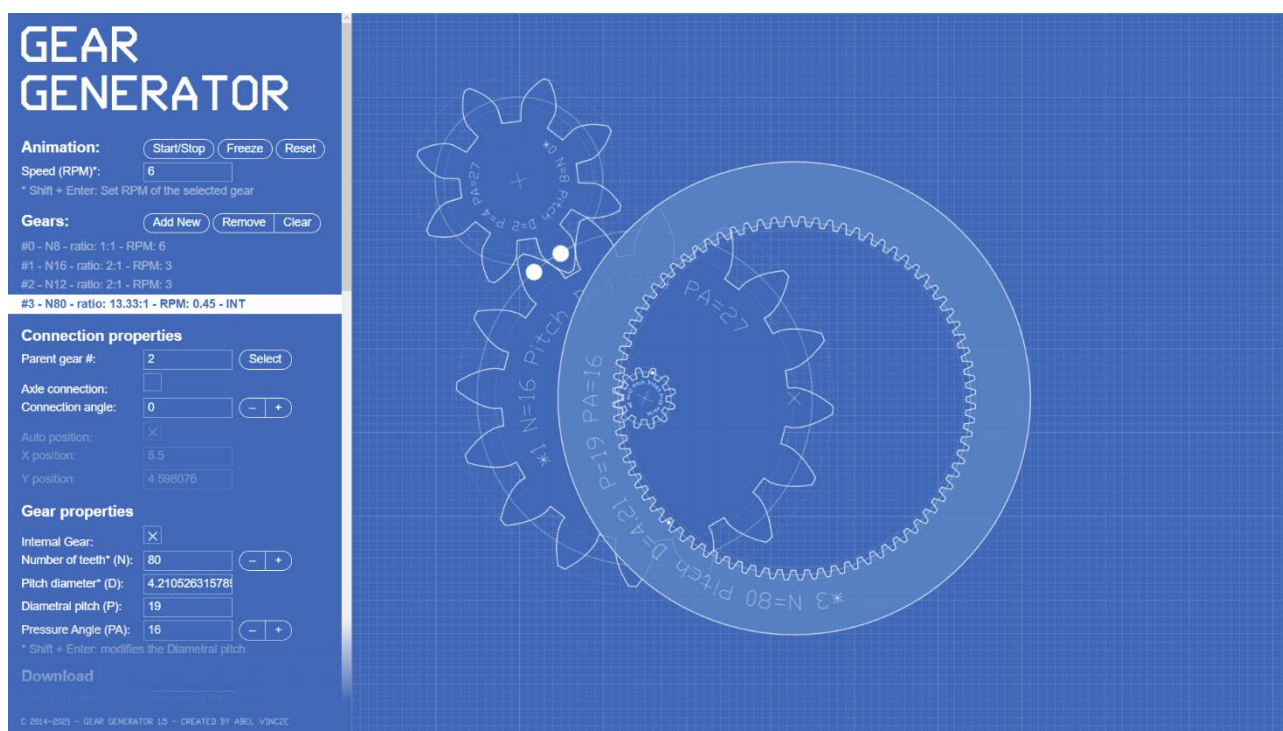


Рисунок 2.1 — Интерфейс сайта «geargenerator.com»

### 2.4.3 SelfCAD

SelfCAD — это онлайн-программа для автоматизированного проектирования 3D-моделей и их 3D-печати, выпущенная в 2016 году. Она основана на браузере и облаке. Её отличительная черта — это отсутствие потребности в скачивании программы и возможность работы напрямую в облаке с сохранением возможности работы в автономном режиме (для чего уже потребуется установка программного обеспечения SelfCAD). [12]

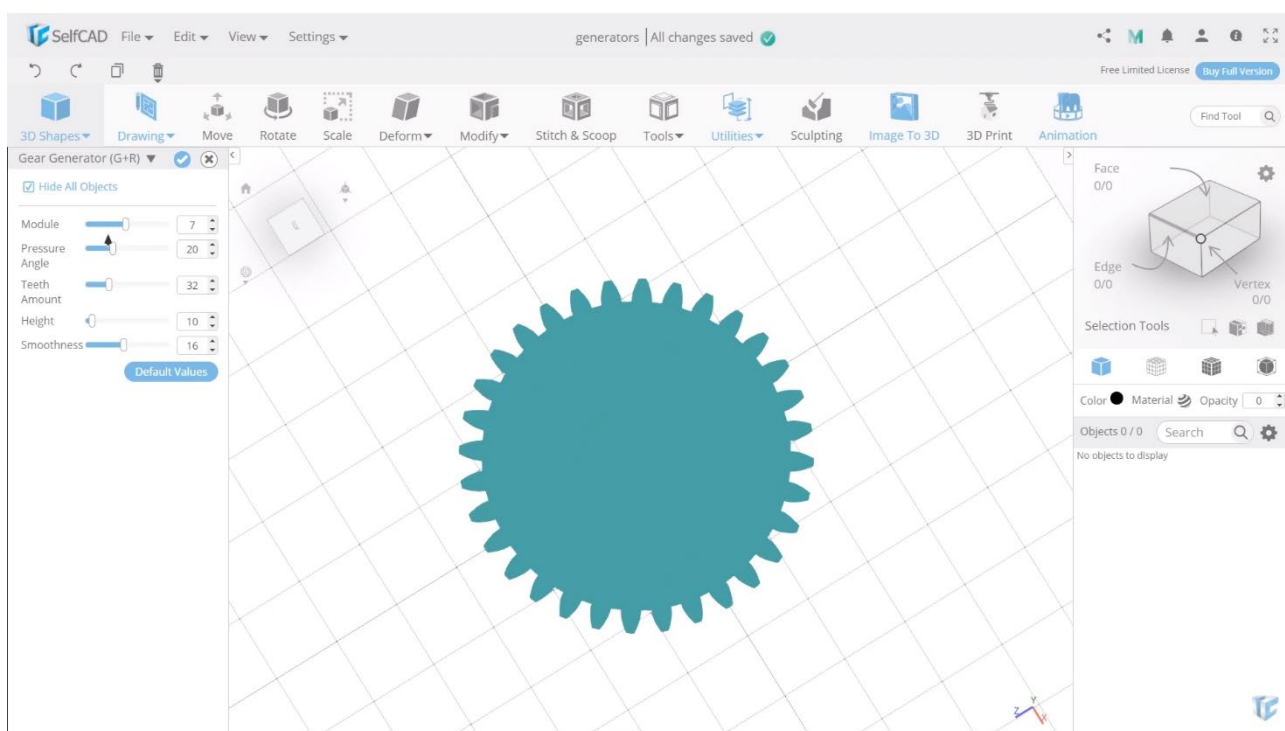


Рисунок 2.2 — Интерфейс программы SelfCAD

### 2.4.1 Autodesk App Store

Также, существует магазин плагинов для всей продукции компании Autodesk. Однако, в нём плагины по созданию звёздочки в основном создаются для программы Fusion 360. В то время как для нужной среды AutoCAD нужных плагинов намного меньше (всего два плагина) и они распространяются на платной основе. [13]

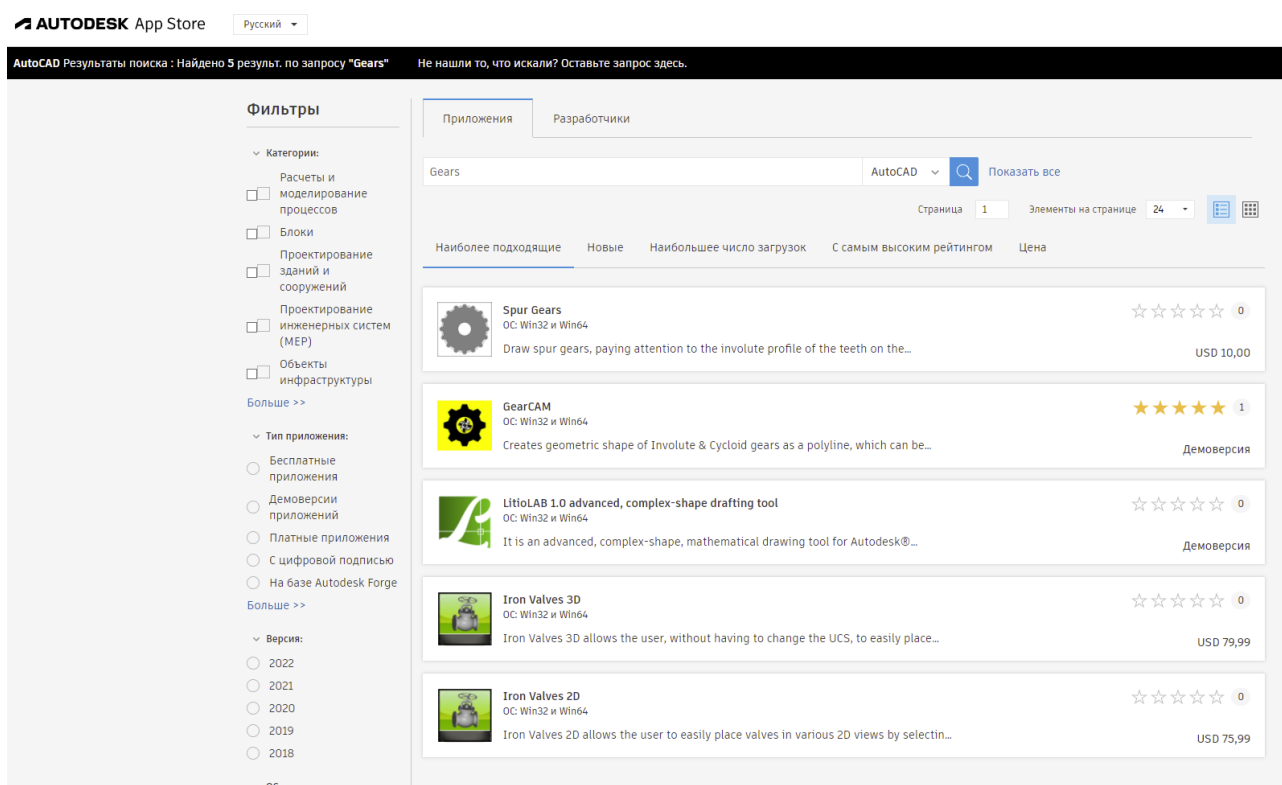


Рисунок 2.3 — Пара платных плагинов AutoCAD для создания звёздочки

### 3 Описание предмета проектирования

Звёздочка (цепное колесо) — это профилированное колесо с зубьями, которые входят в зацепление с цепью, гусеницей или с другими материалами с выемками или зазубринами. Звёздочки отличаются от зубчатых колёс тем, что никогда не входят в зацепление друг с другом непосредственно, и отличаются от шкивов тем, что у звёздочек есть зубья, в то время как шкивы имеют гладкие ободы.

Звёздочки применяются в велосипедах, мотоциклах, автомобилях, гусеничных транспортных средствах, и в других машинах, в которых применение зубчатых передач является неподходящим. Они выполняют функцию передачи вращательного движения между двумя валами посредством цепной передачи или функцию сообщения линейного движения звеньям гусениц. [14]

К изменяемым параметрам модели относятся:

- а) диаметр наружной окружности ( $d$ , 50 — 500 мм);

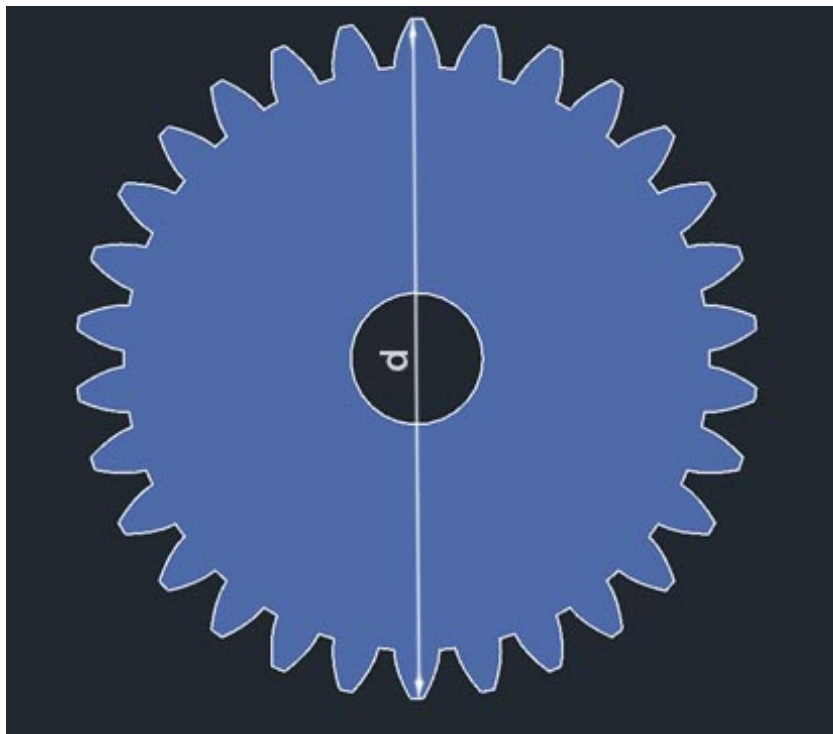


Рисунок 3.1 – Диаметр наружной окружности цепного колеса

- б) диаметр внутренней окружности ( $d_2$ , 25 — 250 мм);

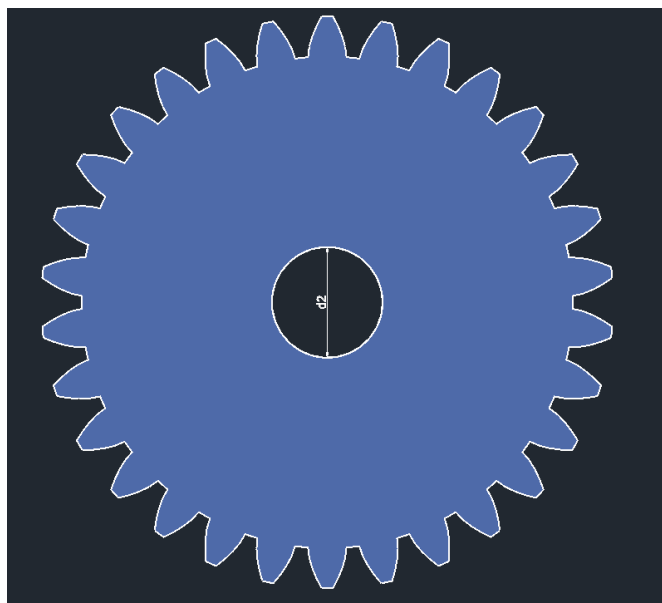


Рисунок 3.2 – Диаметр внутренней окружности звёздочки

- с) число зубьев ( $n$ , 5 — 30);
- d) высота зуба ( $h_r \leq 20\% * d$ );

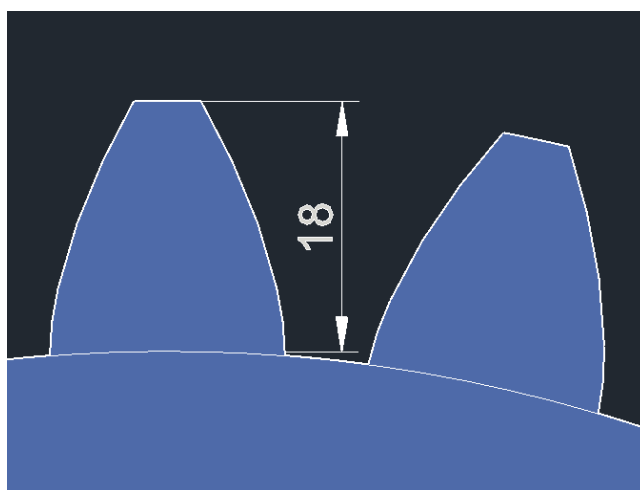


Рисунок 2.3 – Высота зуба

- е) толщина пластины ( $h$ , 5 — 50 мм).

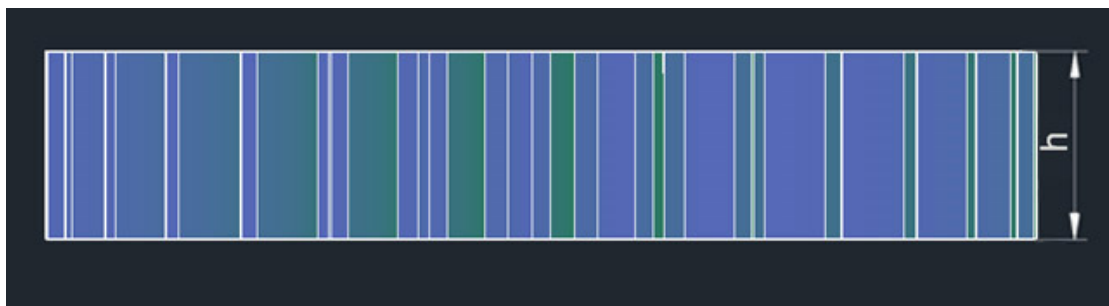


Рисунок 2.4 – Толщина пластины

## **4 Проект системы**

### **4.1 Описание технических и функциональных аспектов системы**

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 [6].

При использовании UML были построена диаграмма классов.

## 4.2 Диаграмма классов

Диаграмму классов используют для отображения структуры проекта. Она отражает отношения между главными сущностями и описывает их внутреннюю структуру.

Спроектированная диаграмма классов для проекта по созданию звёздочки показана на рисунке 4.1:

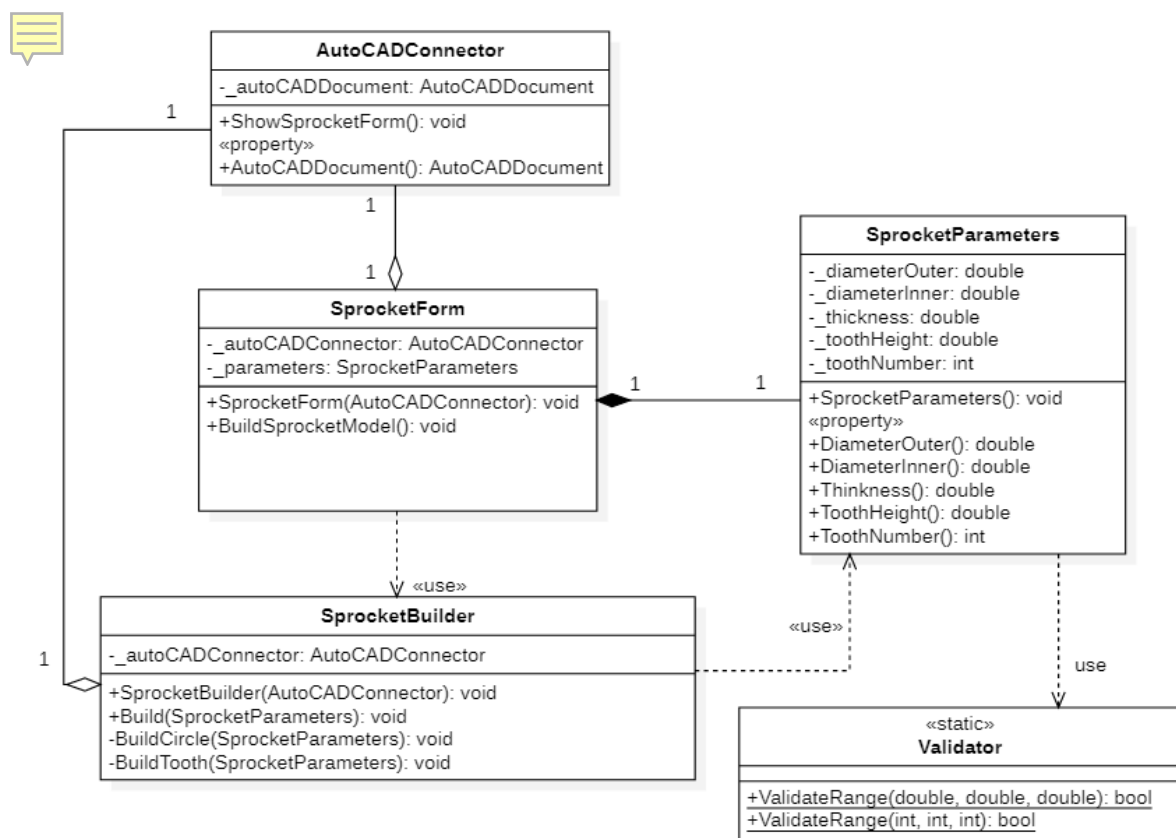


Рисунок 4.1 — Первоначальная UML-диаграмма классов

Для создания плагина AutoCAD будут реализованы следующие классы:

- **AutoCADConnector** — класс, откуда будет запускаться плагин при помощи вызова команды «BuildSprocket» в терминале AutoCAD;
- **SprocketForm** — класс, отвечающий за пользовательский интерфейс плагина;
- **SprocketParameters** — класс, хранящий в себе все параметры модели зубчатого колеса;

- Validator — класс, хранящий в себе методы проверки данных (проверка диапазона значений);
- SprocketBuilder — класс, хранящий в себе методы для построения модели зубчатого колеса.

В ходе работы над пролетом структура классов и отношения между ними были изменены, дописаны новые методы, поля и свойства.

Было принято решение избавиться от статического класса «Validator», так как он содержал в себе лишь один метод проверки значений (проверка по диапазону допустимых значений). Метод, содержащийся в статическом классе был перенесён в класс «SprocketBuilder», где и осуществлялись все проверки параметров.

Итоговая диаграмма классов продемонстрирована на рисунке 4.2.

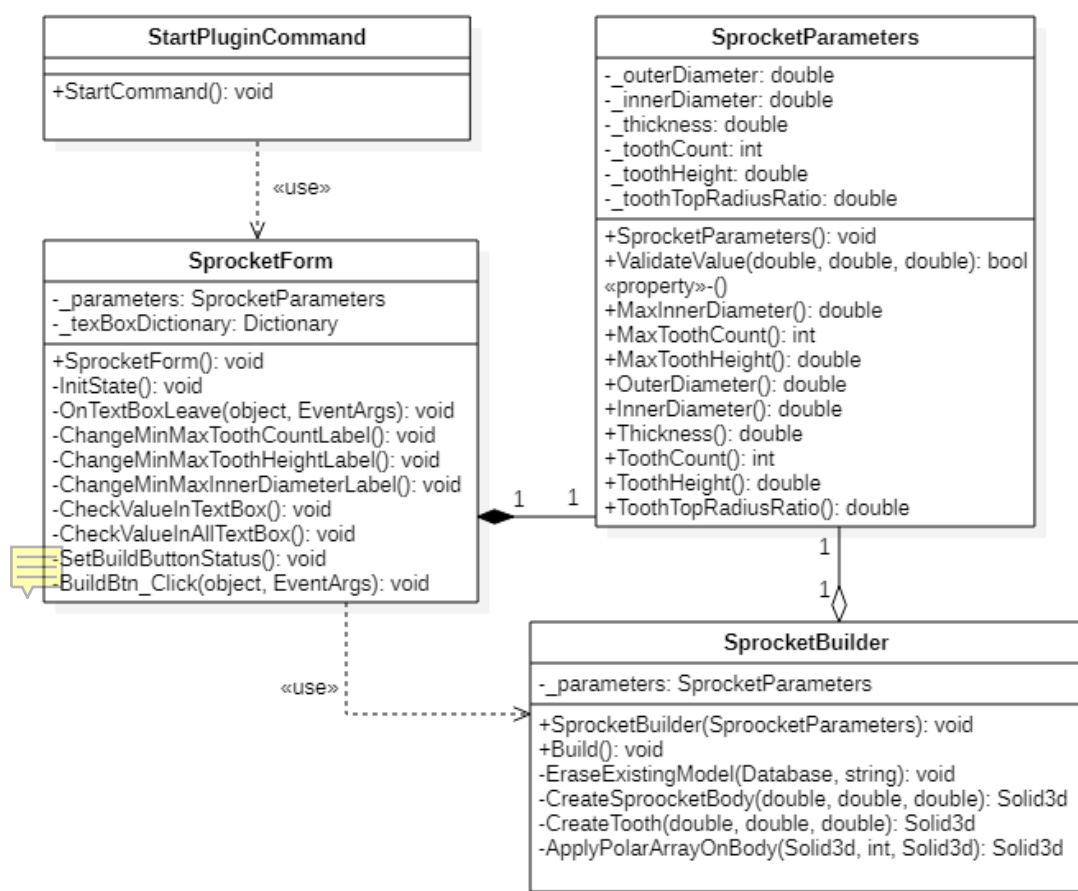


Рисунок 4.2 — Итоговая UML-диаграмма классов



### 4.3 Макет пользовательского интерфейса

Для создания звёздочки хватило бы и передачи аргументов (параметров) в терминале AutoCAD. Однако, такой способ взаимодействия с пользователем не очень дружелюбен.

Намного понятнее для пользователя воспользоваться специальным интерфейсом. С целью улучшения пользовательского опыта использования создаваемого плагина был разработан макет интерфейса.

Макет пользовательского интерфейса представлен на рисунке 4.3:

Построение звёздочки

Диаметр наружной окружности D:  (50 - 500 мм)

Диаметр внутренней окружности Di:  (25 - 250 мм)

Толщина пластины T:  (5 - 50 мм)

Высота зуба t:  (До 16 мм)

Число зубьев:  (5 - 30 шт)

Коэффициент верхнего радиуса зуба  (0.2 - 0.8)

Построить

Рисунок 4.3 — Макет пользовательского интерфейса

Также плагин будет уведомлять пользователя о некорректном вводе данных. Планируется выводить сообщения, содержащие следующую информацию:

- где была совершена ошибка (конкретное поле);
- совершённая ошибка (выход за пределы значений или ввод некорректных символов);
- что программа ожидала получить.

К примеру, когда пользователь вводит в поле «толщина пластины» значение «90» (превышающее указанный диапазон 5-50 мм), плагин выводит следующее сообщение: «Введено неверное значение толщины пластины».

В скобках рядом с полем ввода значение динамически меняется на требуемое, если оно зависит от других характеристик. Таким образом, подсказка пользователю становится более понятной.



Рисунок 4.4 — Пример блокировки и отображения сообщения о неверно введенных данных

В соответствии с макетом были реализованы интерфейс, представленный на рисунке 4.5, и отображение сообщения о неверно введенных данных (рисунок 4.6):

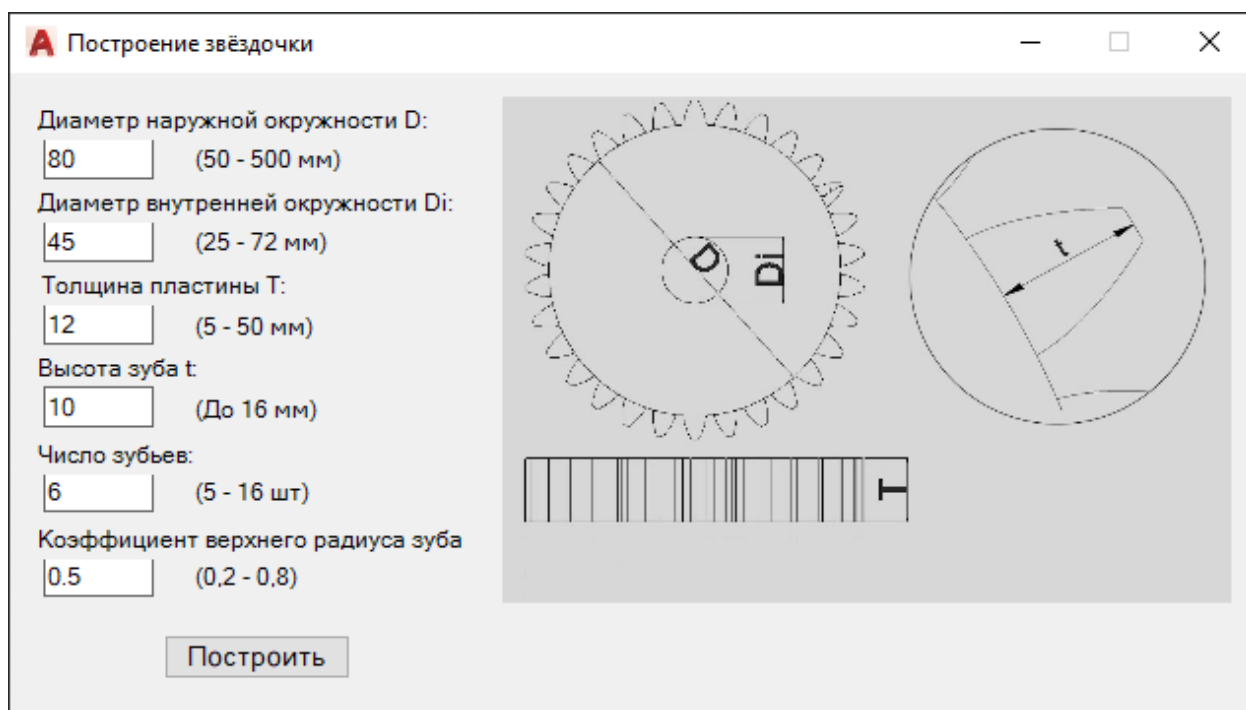


Рисунок 4.5 — Реализованный интерфейс

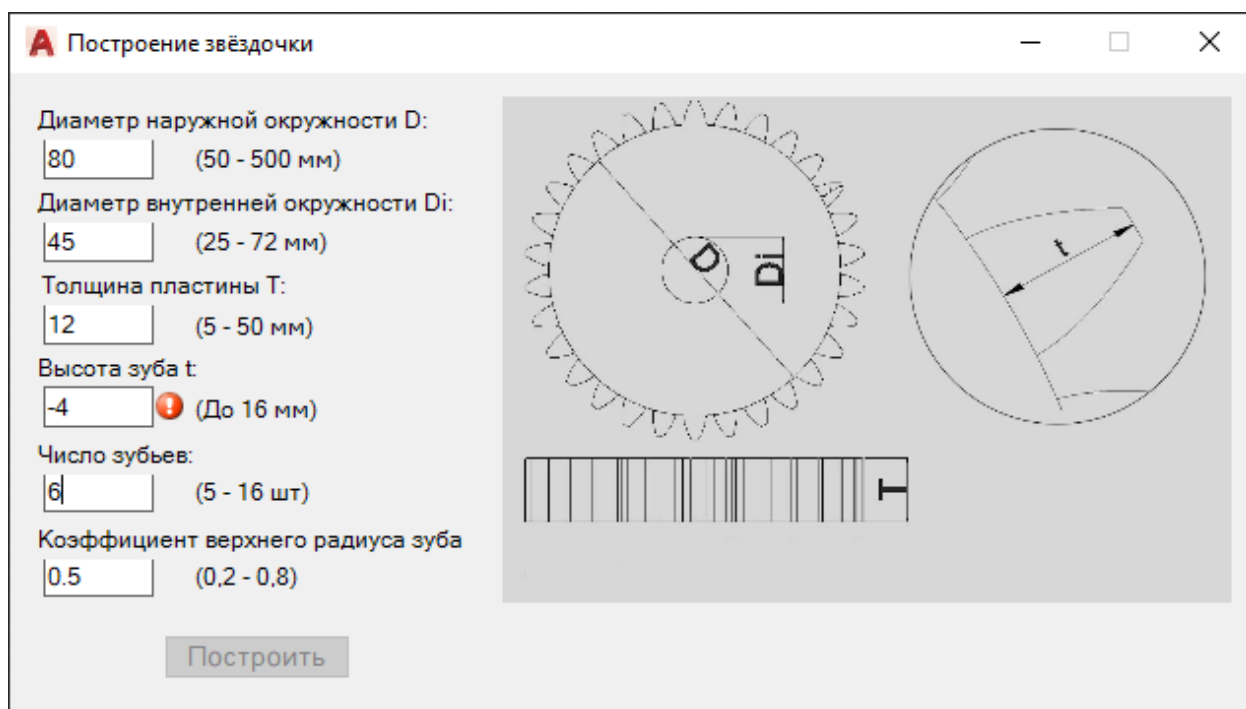


Рисунок 4.6 — Реализованное отображение сообщения о неверно введенных данных

## 5 Тестирование программы

### 5.1 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами [7]. Проведено тестирование максимальных, минимальных и параметров по умолчанию для построения модели.

На рисунке 5.1 показана модель зубчатого колеса, построенная по заданным параметрам по умолчанию (диаметр наружной окружности = 80мм, диаметр внутренней окружности = 45мм, толщина пластины = 12мм, высота зуба = 10мм, число зубьев = 6шт).

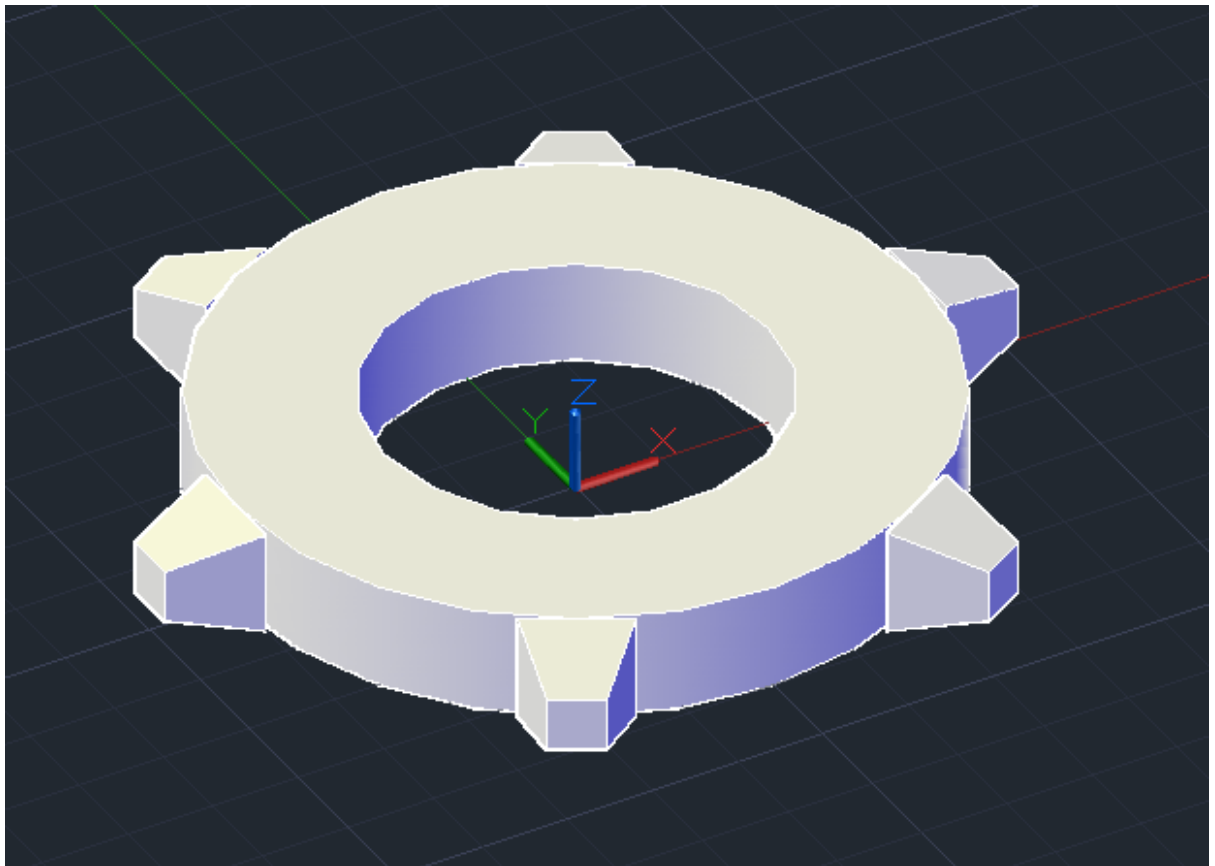


Рисунок 5.1 – Зубчатое колесо, построенное по заданными по умолчанию параметрам

На рисунке 5.2 продемонстрирована модель зубчатого колеса с минимальными параметрами (диаметр наружной окружности = 50мм, диаметр внутренней окружности = 25мм, толщина пластины = 5мм, высота зуба = 5мм, число зубьев = 5шт).

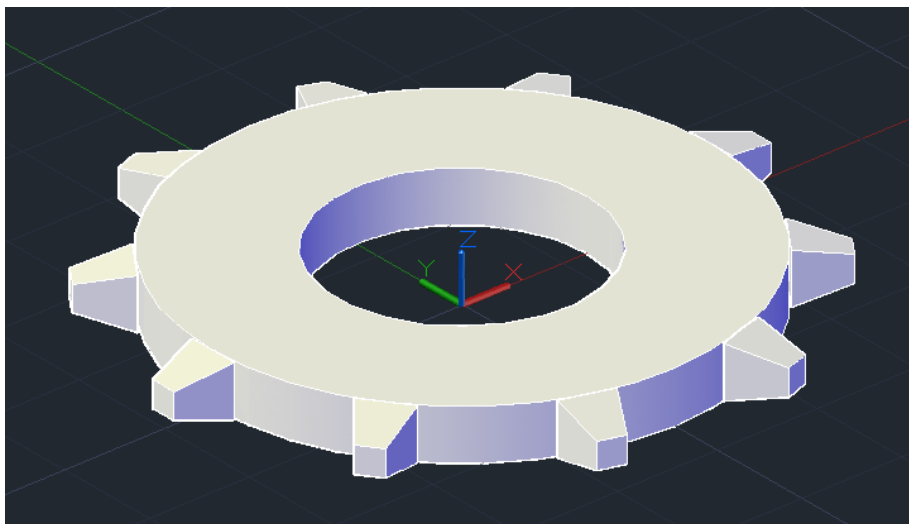


Рисунок 5.2 – Зубчатое колесо, построенное по заданным минимальным параметрам

На рисунке 5.3 продемонстрирована модель зубчатого колеса, построенная с максимальными параметрами (диаметр наружной окружности = 50мм, диаметр внутренней окружности = 25мм, толщина пластины = 5мм, высота зуба = 5мм, число зубьев = 5шт).

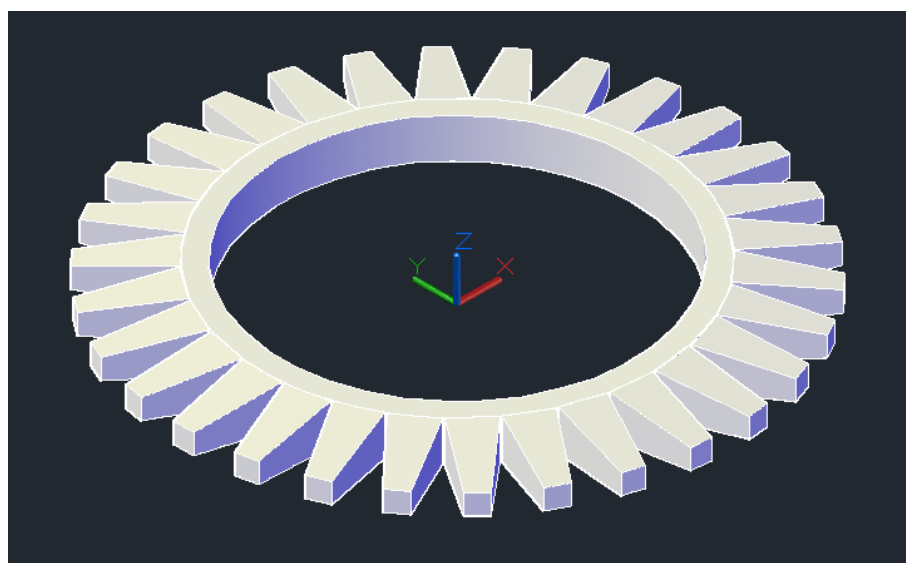
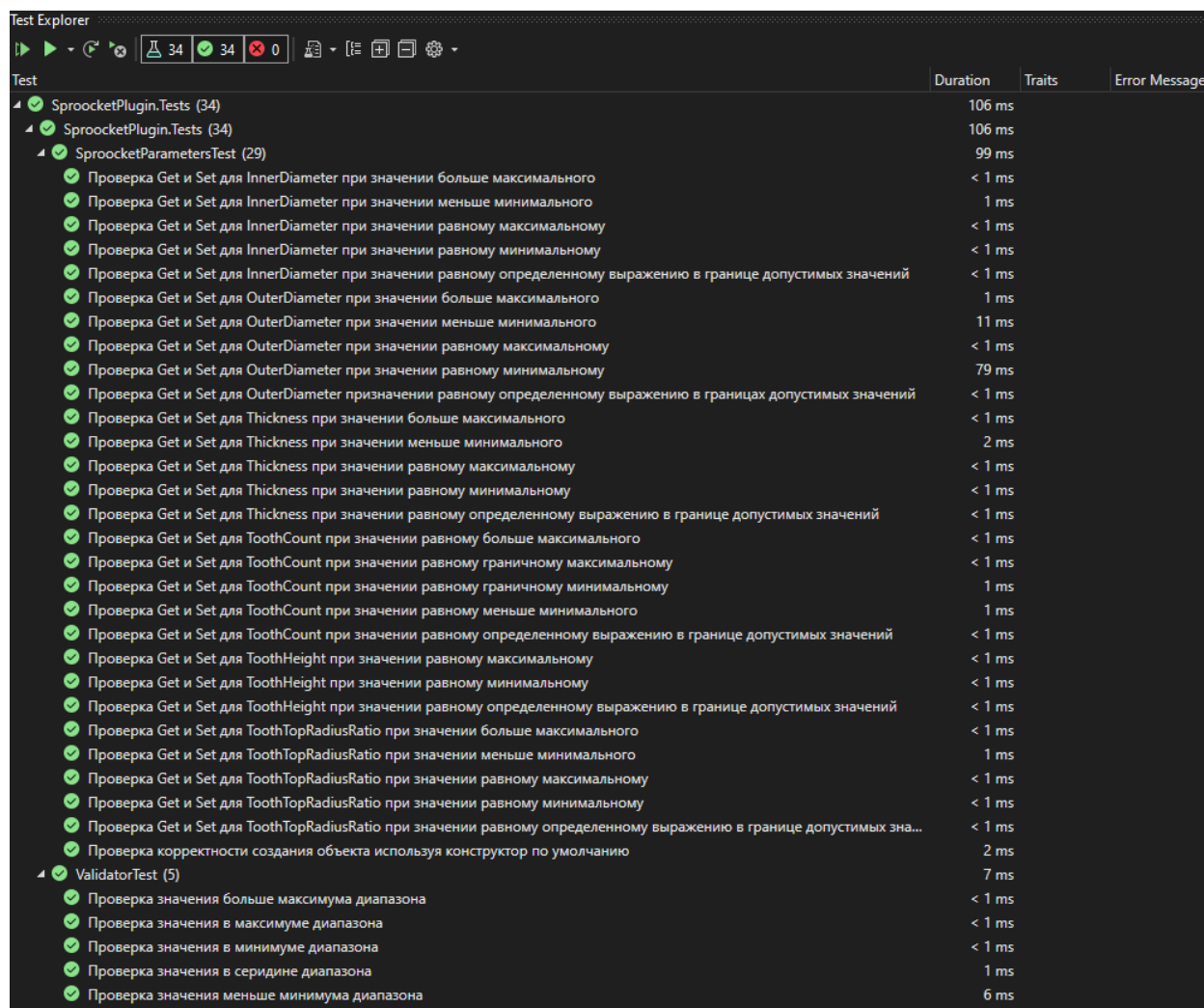


Рисунок 5.3 – Зубчатое колесо, построенное по заданным максимальными параметрам

## 5.2 Модульное тестирование

В целях проверки корректности работы всех модулей программы было проведено также модульное тестирование при помощи фреймворка NUnit версии 3.13 [8]. При модульном проверяются все открытые методы и свойства. Таким образом были написаны модульные тесты для SprocketParameters и Validator со сто процентным покрытием (рисунок 5.5). Результат запуска всех тестов показана на рисунке 5.4. Таблица с описанием тестовых случаев и параметрами тестов приведена в приложении А.



Test	Duration	Traits	Error Message
✓ SprocketPlugin.Tests (34)	106 ms		
✓ SprocketPlugin.Tests (34)	106 ms		
✓ SprocketParametersTest (29)	99 ms		
✓ Проверка Get и Set для InnerDiameter при значении больше максимального	< 1 ms		
✓ Проверка Get и Set для InnerDiameter при значении меньше минимального	1 ms		
✓ Проверка Get и Set для InnerDiameter при значении равному максимальному	< 1 ms		
✓ Проверка Get и Set для InnerDiameter при значении равному минимальному	< 1 ms		
✓ Проверка Get и Set для InnerDiameter при значении равному определенному выражению в границе допустимых значений	< 1 ms		
✓ Проверка Get и Set для OuterDiameter при значении больше максимального	1 ms		
✓ Проверка Get и Set для OuterDiameter при значении меньше минимального	11 ms		
✓ Проверка Get и Set для OuterDiameter при значении равному максимальному	< 1 ms		
✓ Проверка Get и Set для OuterDiameter при значении равному минимальному	79 ms		
✓ Проверка Get и Set для OuterDiameter при значении равному определенному выражению в границах допустимых значений	< 1 ms		
✓ Проверка Get и Set для Thickness при значении больше максимального	< 1 ms		
✓ Проверка Get и Set для Thickness при значении меньше минимального	2 ms		
✓ Проверка Get и Set для Thickness при значении равному максимальному	< 1 ms		
✓ Проверка Get и Set для Thickness при значении равному минимальному	< 1 ms		
✓ Проверка Get и Set для Thickness при значении равному определенному выражению в границе допустимых значений	< 1 ms		
✓ Проверка Get и Set для ToothCount при значении равному больше максимального	< 1 ms		
✓ Проверка Get и Set для ToothCount при значении равному граничному максимальному	< 1 ms		
✓ Проверка Get и Set для ToothCount при значении равному граничному минимальному	1 ms		
✓ Проверка Get и Set для ToothCount при значении равному меньше минимального	1 ms		
✓ Проверка Get и Set для ToothCount при значении равному определенному выражению в границе допустимых значений	< 1 ms		
✓ Проверка Get и Set для ToothHeight при значении равному максимальному	< 1 ms		
✓ Проверка Get и Set для ToothHeight при значении равному минимальному	< 1 ms		
✓ Проверка Get и Set для ToothHeight при значении равному определенному выражению в границе допустимых значений	< 1 ms		
✓ Проверка Get и Set для ToothTopRadiusRatio при значении больше максимального	< 1 ms		
✓ Проверка Get и Set для ToothTopRadiusRatio при значении меньше минимального	1 ms		
✓ Проверка Get и Set для ToothTopRadiusRatio при значении равному максимальному	< 1 ms		
✓ Проверка Get и Set для ToothTopRadiusRatio при значении равному минимальному	< 1 ms		
✓ Проверка Get и Set для ToothTopRadiusRatio при значении равному определенному выражению в границе допустимых зна...	< 1 ms		
✓ Проверка корректности создания объекта используя конструктор по умолчанию	2 ms		
✓ ValidatorTest (5)	7 ms		
✓ Проверка значения больше максимума диапазона	< 1 ms		
✓ Проверка значения в максимуме диапазона	< 1 ms		
✓ Проверка значения в минимуме диапазона	< 1 ms		
✓ Проверка значения в середине диапазона	1 ms		
✓ Проверка значения меньше минимума диапазона	6 ms		

Рисунок 5.4 – Результат запуска тестов

Symbol	Coverage (%) ▲	Uncovered/Total Stmts.
▲ Total	100%	0/66
▲ SprocketPlugin.BL	100%	0/66
▲ SprocketPlugin.BL	100%	0/66
▲ SprocketParameters	100%	0/66
▶ MaxInnerDiameter	100%	0/1
▶ MaxToothHeight	100%	0/1
▶ ValidateValue(double,double,doub	100%	0/3
▶ OuterDiameter	100%	0/7
▶ InnerDiameter	100%	0/7
▶ Thickness	100%	0/7
▶ ToothHeight	100%	0/7
▶ ToothTopRadiusRatio	100%	0/7
▶ MaxToothCount	100%	0/8
▶ ToothCount	100%	0/9
* SprocketParameters()	100%	0/9

Рисунок 5.5 – Анализ покрытия модульного тестирования.

### 5.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [9]. Тестирование производилось на ПК со следующей конфигурацией:

- Core i5-4430, 3.0 GHz;
- 8 Гб ОЗУ.

Было проведено нагрузочное тестирование с параметрами по умолчанию. Для наглядности результата в каждом тестировании проводилось построение 30000 моделей.

#### 5.3.1 Нагрузочное тестирование с параметрами по умолчанию

Для второго нагрузочного теста были выбраны минимальные параметры для построения модели, а именно:

- Диаметр наружной окружности = 80мм;
- Диаметр внутренней окружности = 45мм;
- Толщина пластины = 12мм;

- Высота зуба = 10мм;
- Число зубьев = 6шт.

Тестирование заняло 6 минут. Результаты тестирования продемонстрированы на графиках 5.6 и 5.7.



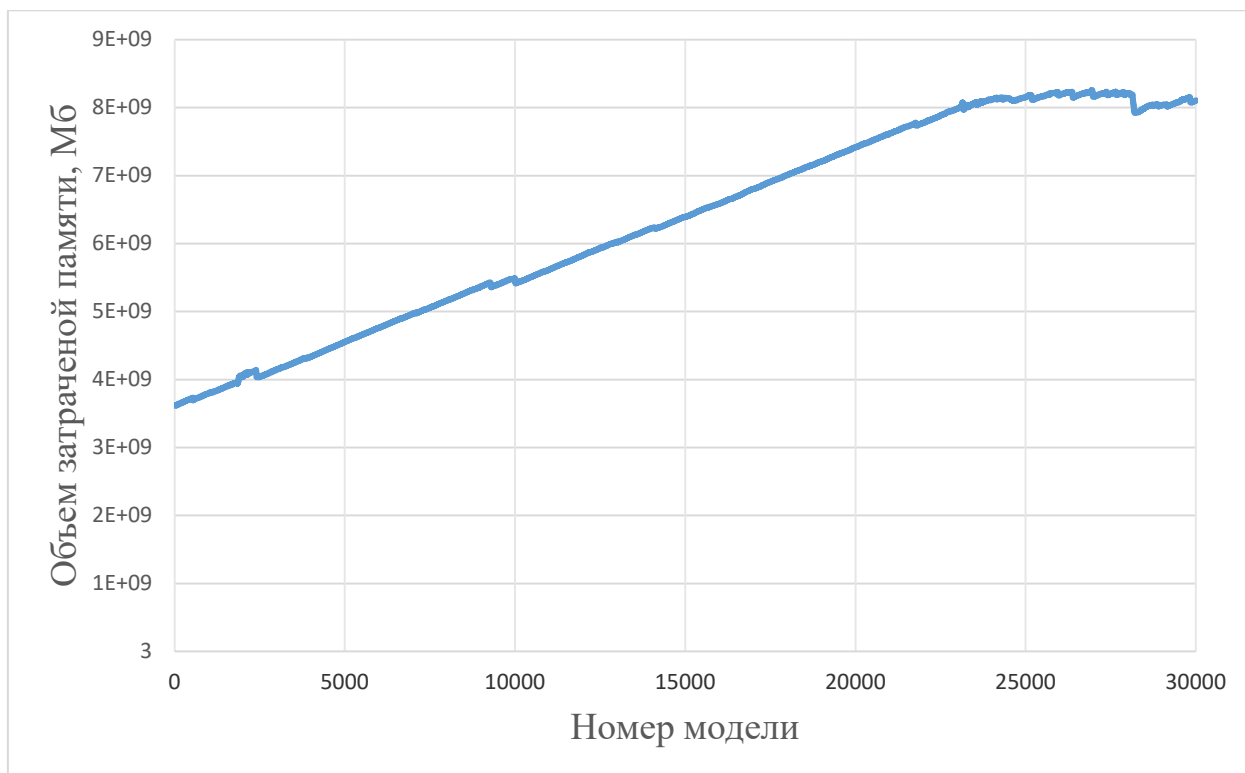


График 5.6 – График зависимости количества потребляемой оперативной памяти от количества моделей с минимальными параметрами

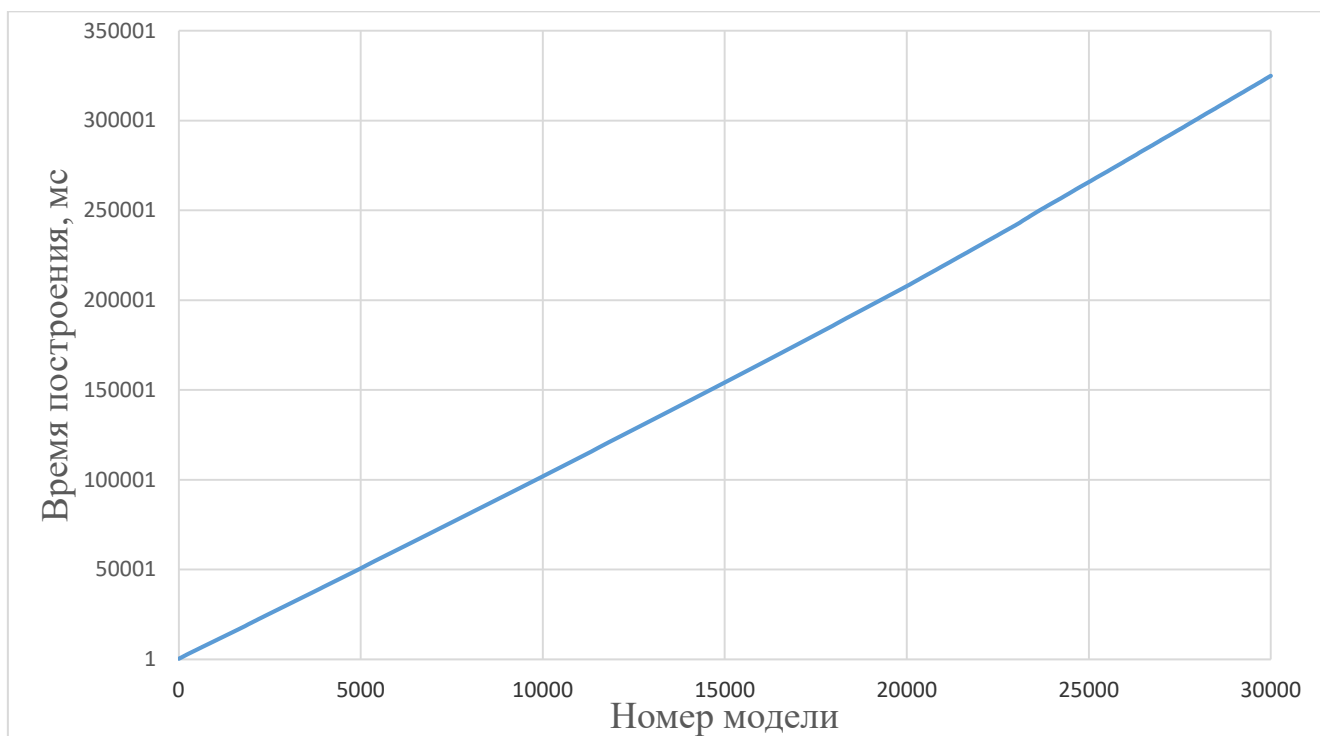


График 5.7 – График зависимости времени построения одной модели от общего количества моделей с минимальными параметрами

## **5.4 Выводы нагрузочного тестирования**

Исходя из всех полученных данных по нагрузочным тестам можно сделать следующие выводы:

Во-первых, использование оперативной памяти, затрачиваемое программой, линейно увеличивается до достижения предела объема оперативной памяти.

Во-вторых, AutoCAD не выключается при максимальной нагрузке на оперативную память, к примеру, как может происходить в САПР «Компас 3D». Можно предположить, что это происходит из-за того, что все построения происходят через записи в базу данных.

В-третьих, по графику 5.6 видно, как программа начинает переиспользовать доступные ресурсы оперативной памяти: после примерно 25 тысяч моделей доступная память заканчивается, и программа, предположительно, начинает выгружать старые модели и создавать на этих освобождённых ресурсах новые модели. На графике это выражается в непродолжительном уменьшении используемой памяти, которая вскоре вновь занимает программа: так получается почти прямая линия около 8Гб (максимального объёма ОЗУ ПК) и работоспособность программы сохраняется.

## **Вывод**

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API для взаимодействия с выбранной САПР, так же по итогам разработки плагина была спроектирована UML-диаграмма классов, разработан плагин для создания 3D модели зубчатого колеса в САПР AutoCAD 2022 и проведено функциональное и нагрузочное тестирование плагина.

## Список литературы

1. Актуальность применения САПР в машиностроении [Электронный ресурс]. URL: <https://sapr.ru/article/7837> (дата обращения: 17.02.2022).
2. Малюх В.Н. Введение в современные САПР: Курс лекций. – М.: ДМК Пресс, 2010. – 192 с.: ил.
3. AutoCAD 2D и 3D система Программы САПР от Autodesk [электронный ресурс]. – Режим доступа: <http://linter.ru/2017/08/09/autocad-2d-i-3d-sistema-programmy-sapr-ot-autodesk/> (дата обращения 17.02.2022);
4. Autodesk. Autodesk Developer Network open. Программные платформы. Разработка приложений для AutoCAD. [электронный ресурс]. – Режим доступа: <https://www.autodesk.ru/autodesk-developer-network/software-platform-russian/develop-autocad> (дата обращения 21.02.2022);
5. Вставка звездочки (AutoCAD Mechanical Toolset). [Электронный ресурс]. — Режим доступа: <https://knowledge.autodesk.com/ru/support/autocad-mechanical/learn-explore/caas/CloudHelp/cloudhelp/2019/RUS/AutoCAD-Mechanical/files/GUID-D8739549-39DC-48A2-97AF-976AE73CB132-htm.html> (дата обращения 31.10.2021)
6. UML. [электронный ресурс]. – Режим доступа: <http://www.uml.org/> (дата обращения 21.02.2022).
7. Функциональное тестирование [Электронный ресурс]. – URL: <https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 17.02.2022).
8. Юнит-тестирование для чайников [Электронный ресурс]. – URL: <https://habr.com/ru/post/169381/> (дата обращения: 17.02.2022).
9. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – URL: <https://habr.com/ru/company/jugru/blog/329174/> (дата обращения: 17.02.2022).

10. AutoCAD — Википедия [электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/AutoCAD> (дата обращения 20.03.2022)
11. Gear generator— онлайн редактор чертежей для создания шестерней. [Электронный ресурс]. — Режим доступа: <https://geargenerator.com> (дата обращения 31.10.2021).
12. 3D SHAPES: GEAR GENERATOR. [Электронный ресурс]. — Режим доступа: <https://www.selfcad.com/3d-modeling-features/3d-shapes-gear-generator> (дата обращения 31.10.2021).
13. Gears | Подключаемые модули, надстройки, расширения для AutoCAD — Autodesk App Store. [Электронный ресурс]. — Режим доступа: <https://apps.autodesk.com/ACD/ru/List/Search?isAppSearch=True&searchboxstore=ACD&facet=&collection=&sort=&query=Gears> (дата обращения 31.10.2021).
14. Звёздочка (техника) — Википедия. [Электронный ресурс]. — Режим доступа: [https://ru.wikipedia.org/wiki/Звёздочка\\_\(техника\)](https://ru.wikipedia.org/wiki/Звёздочка_(техника)) (дата обращения 31.10.2021).

## Приложение А

### Описание тестовых случаев с входными параметрами

Таблица А.1 – Описание тестовых случаев

Название метода	Описание тестового случая	
	Описание	Параметры
AnyParameter_GetSetValue_Success(double expectedValue, string parameterName)	Проверка Get и Set для OuterDiameter при значении равному граничному минимальному.	expectedValue = 50 parameterName = OuterDiameter
	Проверка Get и Set для OuterDiameter при значении равному определенному выражению в границе допустимых значений	expectedValue = 100 parameterName = OuterDiameter
	Проверка Get и Set для OuterDiameter при значении равному граничному максимальному	expectedValue = 500 parameterName = OuterDiameter
	Проверка Get и Set для InnerDiameter при значении равному граничному минимальному	expectedValue = 25 parameterName = InnerDiameter
	Проверка Get и Set для InnerDiameter при значении равному определенному выражению в границе допустимых значений	expectedValue = 100 parameterName = InnerDiameter
	Проверка Get и Set для InnerDiameter при значении равному граничному максимальному	expectedValue = 250 parameterName = InnerDiameter
	Проверка Get и Set для Thickness при значении равному граничному минимальному	expectedValue = 4 parameterName = Thickness
	Проверка Get и Set для Thickness при значении равному определенному выражению в границе допустимых значений	expectedValue = 5 parameterName = Thickness
	Проверка Get и Set для Thickness при значении равному граничному максимальному	expectedValue = 25 parameterName = Thickness
	Проверка Get и Set для ToothHeight при значении равному граничному минимальному	expectedValue = 1 parameterName = ToothHeight
	Проверка Get и Set для ToothHeight при значении равному определенному выражению в границе допустимых значений	expectedValue = 10 parameterName = ToothHeight
	Проверка Get и Set для ToothHeight при значении равному граничному максимальному	expectedValue = 16 parameterName = ToothHeight

Продолжение таблицы А.1

Название метода	Описание тестового случая	
	Описание	Параметры
AnyParameter_GetSetValue_Success(double expectedValue, string parameterName)	Проверка Get и Set для ToothTopRadiusRatio при значении равному граничному минимальному	expectedValue = 0.2 parameterName = ToothTopRadiusRatio
	Проверка Get и Set для ToothTopRadiusRatio при значении равному определенному выражению в границе допустимых значений	expectedValue = 0.5 parameterName = ToothTopRadiusRatio
	Проверка Get и Set для ToothTopRadiusRatio при значении равному граничному максимальному	expectedValue = 0.8 parameterName = ToothTopRadiusRatio
AnyParameter_GetSetValue_Success(int expectedValue, string parameterName)	Проверка Get и Set для ToothCount при значении равному граничному минимальному	expectedValue = 5 parameterName = ToothCount
	Проверка Get и Set для ToothCount при значении равному определенному выражению в границе допустимых значений	expectedValue = 10 parameterName = ToothCount
	Проверка Get и Set для ToothCount при значении равному граничному максимальному	expectedValue = 30 parameterName = ToothCount
AnyParameter_SetValue_Failed(double expectedValue, string parameterName)	Проверка Set для OuterDiameter при присвоении значения равному меньше минимального возможного	value = 5 parameterName = OuterDiameter
	Проверка Set для OuterDiameter при присвоении значения равному больше максимального возможного	value = 600 parameterName = OuterDiameter
	Проверка Set для InnerDiameter при присвоении значения равному меньше минимального возможного	value = 5 parameterName = InnerDiameter
	Проверка Set для InnerDiameter при присвоении значения равному больше максимального возможного	value = 500 parameterName = InnerDiameter

Продолжение таблицы А.1

Название метода	Описание тестового случая	
	Описание	Параметры
AnyParameter_SetValue_Failed(double expectedValue, string parameterName)	Проверка Set для Thickness при присвоении значения равному меньше минимального возможного	value = 2 parameterName = Thickness
	Проверка Set для Thickness при присвоении значения равному больше максимального возможного	value = 150 parameterName = Thickness
	Проверка Set для ToothHeight при присвоении значения равному меньше минимального возможного	value = 0 parameterName = ToothHeight
	Проверка Set для ToothHeight при присвоении значения равному больше максимального возможного	value = 100 parameterName = ToothHeight
	Проверка Set для ToothTopRadiusRatio при присвоении значения равному меньше минимального возможного	value = 0.1 parameterName = ToothTopRadiusRatio
	Проверка Set для ToothTopRadiusRatio при присвоении значения равному больше максимального возможного	value = 0.9 parameterName = ToothTopRadiusRatio
AnyParameter_SetValue_Failed(int expectedValue, string parameterName)	Проверка Set для ToothCount при присвоении значения равному меньше минимального возможного	value = 2 parameterName = ToothCount
	Проверка Set для ToothCount при присвоении значения равному больше максимального возможного	value = 150 parameterName = ToothCount
Constructor_CorrectCreation_Success()	Проверка корректности создания объекта используя конструктор по умолчанию	Отсутствуют



Окончание таблицы А.1

Название метода	Описание тестового случая	
	Описание	Параметры
Validate_CheckValue_IsValid(double min, double max, double value)	Проверка значения равного среднему значению диапазона	min = 5 max = 10 value = 7
	Проверка значения равного минимуму диапазона	min = 5 max = 10 value = 5
	Проверка значения равного максимуму диапазона	min = 5 max = 10 value = 10
Validate_CheckValue_NotValid(double min, double max, double value)	Проверка значения меньше минимума диапазона	min = 5 max = 10 value = 4.99
	Проверка значения больше максимума диапазона	min = 5 max = 10 value = 10.01