Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра компьютерных систем в управлении и проектировании (КСУП)

Пояснительная записка для приложения  
«ШЕВРОННАЯ ШЕСТЕРНЯ»

По дисциплине «Основы разработки САПР (ОРСАПР)»

Выполнил:

Студент гр. 589-2

Сухарев М.А.

Принял:

к.т.н., доцент каф. КСУП

Калентьев А.А.

**Оглавление**

[1 Введение 3](#_Toc128392308)

[2 Постановка и анализ задачи 4](#_Toc128392309)

[2.1 Описание объекта проектирования 4](#_Toc128392310)

[2.2 Назначение программы 5](#_Toc128392311)

[2.3 Выбор инструментов и средств разработки 6](#_Toc128392312)

[3 Обзор аналогов 7](#_Toc128392313)

[4 Описание реализации 8](#_Toc128392314)

[4.1 Диаграмма классов 8](#_Toc128392315)

[5 Описание программы для пользователя 11](#_Toc128392316)

[6 Тестирование программы 14](#_Toc128392317)

[6.1 Функциональное тестирование 14](#_Toc128392318)

[6.2 Модульное тестирование 16](#_Toc128392319)

[6.3 Нагрузочное тестирование 16](#_Toc128392320)

[7 Заключение. 19](#_Toc128392321)

[Список использованных источников 20](#_Toc128392322)

# Введение

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

Целью данной работы является разработка плагина для автоматизации построения твердотельной трехмерной модели в среде системы автоматизированного проектирования (САПР) Autodesk Inventor 2022 в виде классического desktop приложения, реализованного на платформе .Net 6.0 с использованием архитектурного паттерна MVVM [2] и компонента платформы WPF [3].

# Постановка и анализ задачи

В рамках проведенных лабораторных работ требовалось в соответствии с техническим заданием реализовать поставленную задачу в виде реализации плагина для автоматизации построения модели шевронной шестерни на основе входных параметров.

## Описание объекта проектирования

Зубчатое колесо (шестерня) – это основная деталь зубчатой передачи в виде диска с зубьями на цилиндрической или конической поверхности, входящими в зацепление с зубьями другого зубчатого колеса.

Изобретение шевронного профиля зуба часто приписывают Андре Ситроену, однако на самом деле он лишь выкупил патент на более совершенную схему, которую придумал польский механик-самоучка. Зубья таких колёс изготавливаются в виде буквы «V» (либо они получаются стыковкой двух косозубых колёс со встречным расположением зубьев). Шевронные колёса решают проблему осевой силы. Осевые силы обеих половин такого колеса взаимно компенсируются, поэтому отпадает необходимость в установке валов на упорные подшипники. При этом передача является самоустанавливающейся в осевом направлении, по причине чего в редукторах с шевронными колесами один из валов устанавливают на плавающих опорах (как правило — на подшипниках с короткими цилиндрическими роликами) [4].

Схема моделируемого объекта:

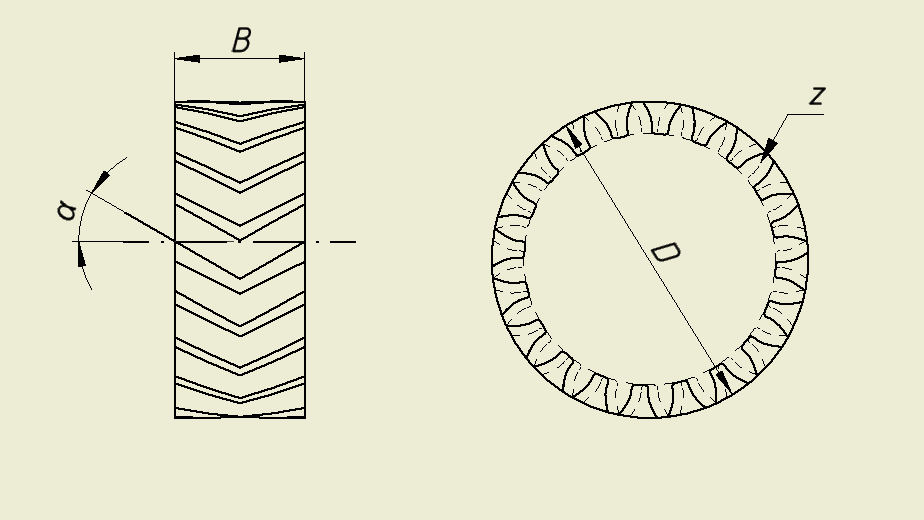


Рисунок 2.1. Схемотическое изображение моделируемого объекта

Измеряемые параметры для плагина:

* B – ширина шестерни (5мм – 5000мм)
* D – внешний диаметр (5мм – 5000мм)
* z – количество зубьев ()
* α – угол наклона зуба (±25° – ±45°)
* m – модуль шестерни. Рассчитывается из формулы , при установке значений D и z будет рассчитан автоматически. Также, при ручной установке значения m, плагин сам подберет значение недостающему параметру z или D.

## Назначение программы

Предназначение программы заключается в автоматизации моделирования шевронных шестерней.

Программа 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 правильно введенных значениях результатом работы программы будет созданная по ним трехмерная твердотельная модель шевронной шестерни.

## Выбор инструментов и средств разработки

В соответствии с техническим заданием, программа выполнена на платформе .NET 6.0 с использованием языка программирования C#. Также были задействованы сторонние библиотеки:

* Autodesk.Inventor.Interop – для подключения к САПР Inventor и взаимодействия с их API;
* CommunityToolkit.Mvvm – для реализации паттерна MVVM.

В качестве среды разработки использовалась IDE Visual Studio 2022 с подключенными плагинами JetBrains ReSharper и JetBrains DotCover.

В качестве обеспечения кодовой базы были использованы автоматизированные модульные и нагрузочные тесты. Модульные тесты реализованы с помощью библиотеки NUnit. Для реализации нагрузочного тестирования были задействованы стандартные средства платформы .NET. Функциональное тестирование программы на соответствие реализации с техническим заданием производится вручную. Также в проекте задействована система контроля версий Git.

# Обзор аналогов

Ближайшим аналогом для разрабатываемого расширения является встроенная утилита по автоматическому созданию шестерней в Autodesk Inventor.

При запуске генератора появляется окно (рис. 3.1), которое необходимо заполнить данными для создания зубчатого соединения. При правильном заполнении программа генерирует модели с учетом пользовательских параметров.

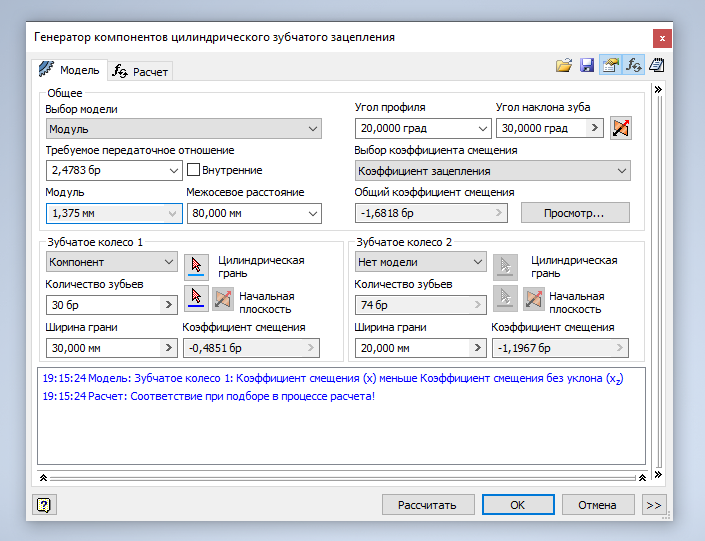


Рисунок 3.1. Интерфейс утилиты Inventor для построения целиндрических шестерней

# Описание реализации

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML[5].

UML диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

## Диаграмма классов

Разработанная на этапе проекта системы диаграмма имела следующий вид:

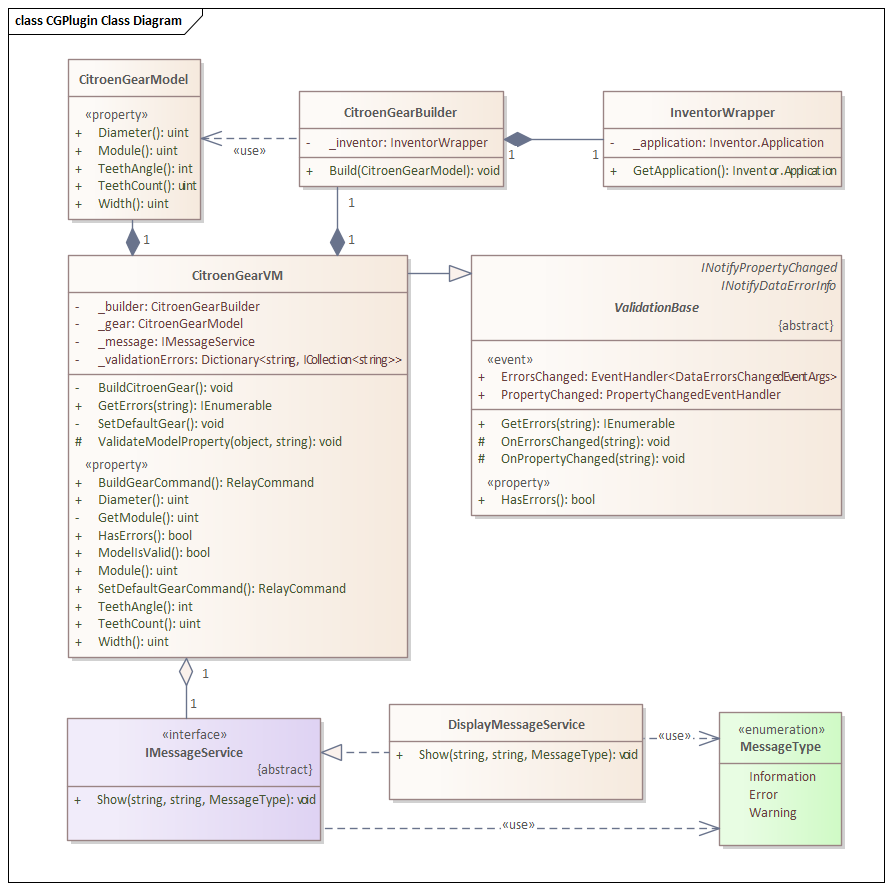


Рисунок 4.1.1. UML-диаграмма классов программы на этапе проекта системы

В процессе разработки были произведены отклонения от изначального варианта, и диаграмма приняла следующий вид:

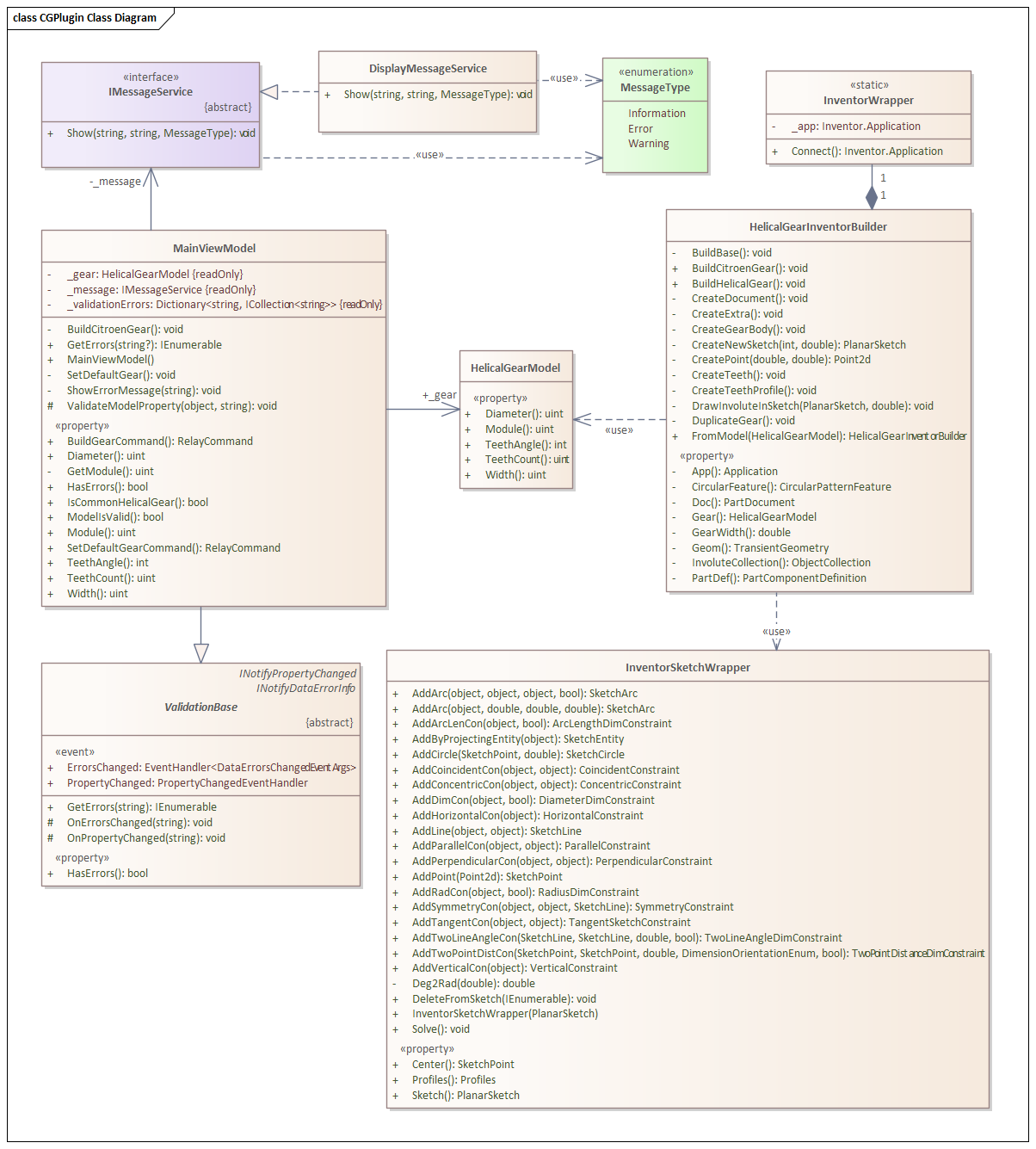


Рисунок 4.1.2. Конечная UML-диаграмма классов

Критических изменений изначальной архитектуры не произошло, но добавился новый класс InventorSketchWrapper, который реализует паттерн «декоратор» и используется для упрощения взаимодействия с API Inventor.

Класс CitroenGearBuilder сменил название на HelicalGearBuilder и стал полноценной реализацией паттерна «строитель», также в нем появился дополнительный функционал в виде построения косозубой шестерни.

Класс InventorWrapper стал больше реализацией паттерна «одиночка», чем изначально запланированного паттерна «декоратор», но свое поведение не изменил и все также предоставляет доступ к API Inventor.

Класс CitroenGearVM сменил название на MainViewModel, а также в нем появилась дополнительная логика для построения как шевронных, так и косозубых шестерней. Также в нем появись RelayCommand для связи с View

# Описание программы для пользователя

Макет пользовательского интерфейса представляет собой форму для ввода параметров. При запуске программы в полях для ввода параметров заданы значения в виде нуля, а кнопка Build не активна. Пользователь может менять данные параметры.

По изначальному проекту системы пользовательский интерфейс выглядел так:

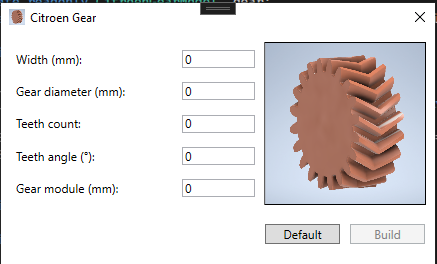


Рисунок 5.1. Макет пользовательского интерфейса из проекта системы

При добавлении дополнительного функционала в виде возможности построения косозубой шестерни, чье отличие от шевронной не так велико, появился дополнительный флажок при включении которого строится косозубая шестерня в место шевронной. Валидация параметров для шевронной и косозубой одинакова. Также при активации флажка изображение шестерни меняется на то, что будет строиться.

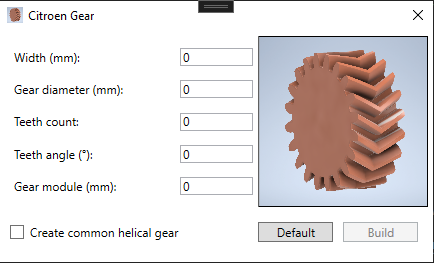


Рисунок 5.2 Конечный вид пользовательского интерфейса

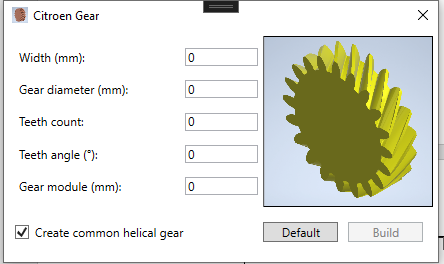


Рисунок 5.3. Демонстрация изменения изображения шестерни при активации флажка

При вводе некорректных значений также как и в проекте системы происходит изменение стиля поля ввода и появляется уведомление.

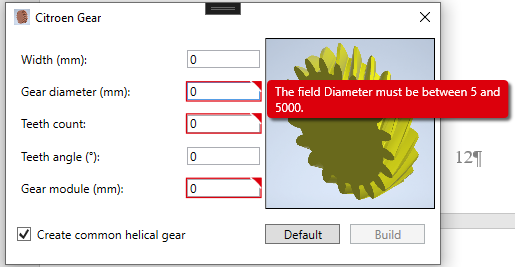


Рисунок 5.4. Пример обработки некоректных вводных данных

Поля Gear diameter, Teeth count, Gear module – являются взаимозависимыми. При изменении полей Gear diameter и Teeth count, значение поля Gear module тоже будет изменено.

Кнопка Default устанавливает заранее заготовленные значения. Если все значения корректны, то кнопка Build становится активной.

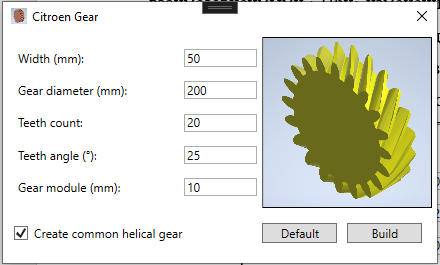


Рисунок 3.2.3. Пример установки заготовленных значений

# Тестирование программы

Тестирование программы необходимо для обеспечения надежности корректной работы при расширении функционала. Тесты помогают находить неполадки в уже реализованном коде, которые могут возникнуть при добавлении нового кода.

## Функциональное тестирование

При функциональном тестировании проверялось корректность работы программы, а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [6]

В качестве входных параметров взяты стандартные параметры, которые устанавливаются кнопкой «Default».

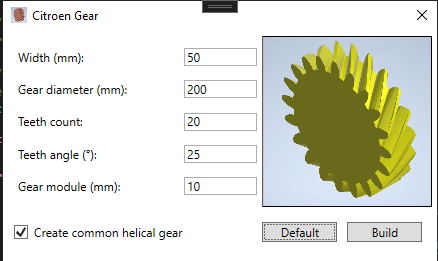


Рисунок 6.1.1. Входные параметры

После построения получается следующая модель:

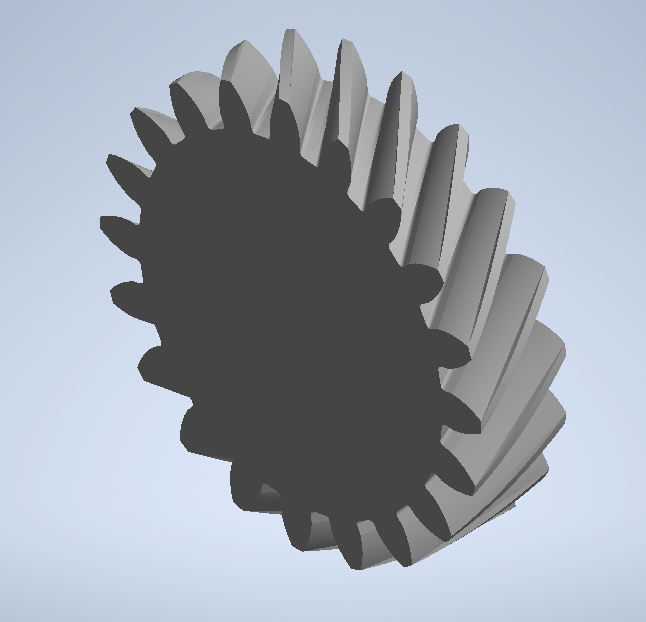


Рисунок 6.1.2. Результат работы программы

Для проверки правильности построения воспользуемся встроенными в Inventor средствами измерения модели. В результате получаем следующее:

* Толщина соответствует 50мм;
* Количество зубьев соответствует 20;
* Основной диаметр соответствует 200мм, внешний и внутренний соответствуют своим значениям, рассчитанным вручную по формулам;
* Модуль шестерни расходится с расчетным значением на примерно ±0.05мм;
* Угол соответствует 25°.

Повтор тестирования уже с моделью шевронной шестерни показал такие же результаты.

## Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit проведено модульное тестирование [7], проверялись открытые поля модели, а также реализованные DataAnnotations для валидации. Степень покрытия бизнес-логики – 100%, для этого был написан 51 тест.

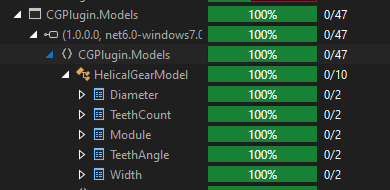


Рисунок 6.2.1. Покрытие модульными тестами бизнес-логики

## Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [8]. Тестирование производилось на ПК со следующей конфигурацией:

– CPU AMD Ryzen 5 3400g 3.9 GHz;

– RAM 16Gb;

– iGPU Vega 11 2Gb

Для нагрузочного тестирования был задан цикл на построение 1000 деталей. Для измерения времени был использован класс Stopwatch, а для показателей оперативной памяти использовался WMI класс Win32\_OperatingSystem. В качестве детали стоились шевронные шестерни так-как они требуют больше операций, чем косозубые и сильнее нагружают систему. Параметры шестерни, такие же, как и в функциональном тестировании – стандартные.

В ходе выполнения цикла, Inventor выдал ошибку и прекратил свою работу на 177 детали, в этот момент случилось переполнение оперативной памяти.

Все данные, полученные от теста были записаны в csv файл, а также продублированы в консоль. По этим данным можно сделать следующие выводы:

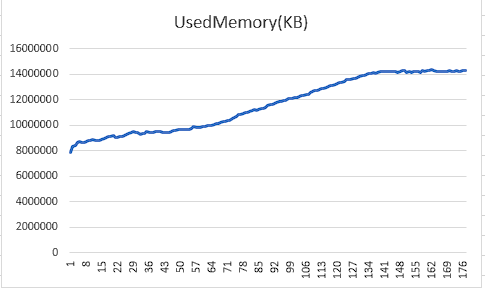


Рисунок 6.3.1. График использования оперативной памяти

Из примерно 14.7 Gb свободной памяти с самого начала было занято 8 Gb прочими службами и программами на устройстве. Нарастание используемой памяти самим Inventor заметно ощущается, в конце работы он занимал примерно 6.5 Gb. Рост был связан в основном из-за необходимости хранить несохраненные модели в оперативной памяти. Уже на 135 операции был достигнут предел занимаемой памяти.

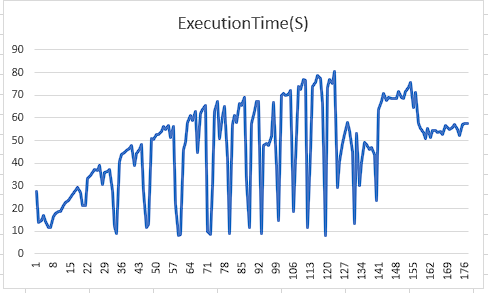


Рисунок 6.3.2. График веремени построения детали

На первой операции было немного завышенное время выполнения - это связано необходимостью загрузки самого Inventor, все следующие операции проходили в одном экземпляре программы.

График становится не стабилен после 30 операции, резкое уменьшение времени скорее всего связано с запуском сборщика мусора и освобождением оперативной памяти под процесс построения.

После 135 операции время начинает стабилизироваться, но места все равно уже перестает хватать.

Самое большое время построения – 80 с, а самое маленькое чуть меньше 10 с. Среднее время за 176 операций составило 47 с.

# Заключение.

В ходе выполнения лабораторных работ были изучены основные этапы проектирования программного продукта и его реализации, предметная область объекта проектирования Inventor API. Было составлено техническое задание, разработан проект системы, составлены UML диаграммы классов, разработан макет пользовательского интерфейса.

В результате работы был разработан и реализован плагин для САПР Autodesk Inventor, выполняющий построение 3D-моделей шевронной и косозубой шестерни по заданным параметрам.

Над реализованным плагином были проведены функциональные, модульные и нагрузочные тесты на платформе Windows 10.

# Список использованных источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – Режим доступа: <https://scienceforum.ru/2014/article/2014000201> (дата обращения: 18.02.2023).
2. MVVM: полное понимание (+WPF) Часть 1 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/338518> Дата обращения (21.10.2022)
3. Что такое Windows Presentation Foundation (WPF) [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-<ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022> (дата обращения: 10.12.2021).
4. Зубчатое колесо [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Зубчатое_колесо> Дата обращения (21.10.2022)
5. Фаулер М. UML. Основы, 3-е издание СПб: символ-Плюс, 2004. - стр. 192.
6. Функциональное тестирование [Электронный ресурс]. – Режим доступа: <https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 20.02.2023).
7. Юнит-тестирование для чайников [Электронный ресурс]. – Режим доступа: [https://habr.com/ru/post/169381/](https://habr.com/ru/post/169381/%20%20)  (дата обращения: 20.02.2023).
8. Нагрузочное тестирование: с чего начать и куда смотреть [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/jugru/blog/329174/> (дата обращения: 22.02.2023).