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

по дисциплине

«Основы разработки САПР»

Тема проекта: Разработка плагина «Звездочка цепной передачи» для САПР «Компас-3D» v 16

Выполнил:

Студент гр. 586-1

\_\_\_\_\_\_\_\_ А. С. Смакотина

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Проверил:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ А. А. Калентьев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Томск 2020

**Реферат**

Пояснительная записка 25 с., 16 рис., 2 таблицы, 9 источника.

КОМПАС-3D, ПЛАГИН, ЗВЕЗДОЧКА ЦЕПНОЙ ПЕРЕДАЧИ, САПР.

Целью данной работы является создание плагина для построения трехмерной модели в системе автоматизированного проектирования КОМПАС-3D 16, с помощью интегрированной среды разработки Visual Studio Сommunity 2019.

В результате разработан плагин, строящий трехмерную модель звездочки цепной передачи в КОМПАС-3D.

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2016.

**Содержание**

[Введение 4](#_Toc39338379)

[1 Постановка и анализ задачи 5](#_Toc39338380)

[1.1 Описание САПР 5](#_Toc39338381)

[1.2 Описание предмета проектирования 6](#_Toc39338382)

[1.3 Описание аналогов 8](#_Toc39338383)

[1.3.1 Дополнительный модуль КОМПАС-3D «Валы и механические передачи». 8](#_Toc39338384)

[1.3.2 MechaniCS 9](#_Toc39338385)

[1.4 Выбор инструментов и средств реализации 11](#_Toc39338386)

[2 Описание реализации 12](#_Toc39338387)

[2.1 Диаграмма прецедентов класса 12](#_Toc39338388)

[2.2 Диаграмма классов 14](#_Toc39338389)

[3 Описание программы для пользователя 16](#_Toc39338390)

[4 Тестирование программы 18](#_Toc39338391)

[4.1 Функциональное тестирование 18](#_Toc39338392)

[4.2 Модульное тестирование 20](#_Toc39338393)

[4.3 Нагрузочное тестирование 22](#_Toc39338394)

[Заключение 24](#_Toc39338395)

[Список используемых источников 25](#_Toc39338396)

# **Введение**

В настоящее время проектирование представляет собой автоматизированный процесс и в некотором роде программно-аппаратный. Проектировщику, который занимается разработкой сложного механизма, или устройства, требующего больших расчетов, математических вычислений при построении модели и высокой точности, подходят системы автоматизации проектных решений — САПР [1].

САПР позволяют сократить финансовые затраты на разработку макета (модели) проекта (объекта), время, которое тратит проектировщик на создание модели объекта и составление проектной документации.

В каждой крупной САПР есть свой средства для разработки, который предоставляются с целью дать возможность разработчикам расширить функционал данной системы под свои конкретные нужды. Расширение функционала в основном подразумевает разработку плагина или библиотеки на основе предоставленного API.

Целью данной работы является создание плагина для построения трехмерной модели звездочки цепной передачи в системе автоматизированного проектирования КОМПАС-3D 16, с помощью интегрированной среды разработки Visual Studio Сommunity 2019.

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

# **1 Постановка и анализ задачи**

В рамках учебной дисциплины ОРСАПР требовалось разработать плагин в соответствии с техническим заданием. На основе заданных параметров плагин, взаимодействуя с КОМПАС-3D, должен строить модель звездочки цепной передачи. Также плагин должен позволять изменять входные параметры звездочки цепной передачи. Изменяемые параметры: радиус внешней окружности, радиус цилиндра, радиус отверстия, толщину внешней части звездочки, толщину цилиндра, глубину шпоночной выемки, количество зубьев и глубину зубьев.

## **Описание САПР**

**«Компас 3D»** – является комплексной системой автоматизированного проектирования, направленная не только на машиностроение, но и на разработку чертежей, проектирование кабельных систем и создание документов для инженерных проектов [1].

Создает проекты для строительной и промышленной направленности любой степени сложности, позволяет создавать изделие от идеи до полного проекта с готовыми документами.

Система «Компас-3D» включает следующие компоненты: система трёхмерного твердотельного моделирования, универсальная система автоматизированного проектирования «Компас-График» и модуль формирования спецификаций. Ключевой особенностью «Компас-3D» является использование собственного математического ядра и пара2метрических технологий.

Общее назначение системы КОМПАС-3D — создание трехмерных ассоциативных моделей отдельных элементов и сборных конструкций из них. Конструкции могут содержать как оригинальные, так и стандартизованные конструктивные элементы, при использовании технологии интеллектуального проектирования [MinD](http://construction.ascon.ru/software/tasks/items/?prcid=104&prpid=829).

Применение свободного моделирования дает возможность создать индивидуальные проекты, отвечающие вкусам и потребностям заказчика и требующие концептуальной проработки и моделирования сложных инсталляций различных форм и композиций.

Для использования КОМПАС-3D в строительном проектировании существует несколько подходов:

1. Формирование 3D-моделей на основе 2D-моделей, выполненных с применением технологии [MinD](http://construction.ascon.ru/software/tasks/items/?prcid=104&prpid=829).
2. Редактирование сформированных 3D-моделей на основе 2D-моделей, выполненных с применением технологии [MinD](http://construction.ascon.ru/software/tasks/items/?prcid=104&prpid=829).
3. Свободное моделирование в трехмерном пространстве для создания нестандартных элементов, оборудования, проработки узлов.

Система обладает мощным функционалом для работы над проектами разнообразной направленности и сложности.

Особенности Компас-3D:

1. собственное ядро;
2. русскоязычный интерфейс;
3. интеграция с другими программам;
4. возможность проектирования трубопроводов, кабелей и кабельных систем;
5. встроенный модуль для создания электрических цепей.

## **Описание предмета проектирования**

Звёздочка – это профилированное [колесо](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D0%B5%D1%81%D0%BE) с зубьями, которые входят в зацепление с [цепью,](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D0%BB%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%86%D0%B5%D0%BF%D1%8C) [гусеницей](https://ru.wikipedia.org/w/index.php?title=%D0%93%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D1%8C&action=edit&redlink=1) или с другими материалами с выемками или зазубринами [2]. Звёздочки отличаются от [зубчатых колёс](https://ru.wikipedia.org/wiki/%D0%97%D1%83%D0%B1%D1%87%D0%B0%D1%82%D0%BE%D0%B5_%D0%BA%D0%BE%D0%BB%D0%B5%D1%81%D0%BE) тем, что никогда не входят в зацепление друг с другом непосредственно, и отличаются от [шкивов](https://ru.wikipedia.org/wiki/%D0%A8%D0%BA%D0%B8%D0%B2) тем, что у звёздочек есть зубья, в то время как шкивы имеют гладкие ободы.

Звёздочки применяются в [велосипедах,](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BB%D0%BE%D1%81%D0%B8%D0%BF%D0%B5%D0%B4) [мотоциклах,](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%82%D0%BE%D1%86%D0%B8%D0%BA%D0%BB) [автомобилях,](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C) [гусеничных](https://ru.wikipedia.org/wiki/%D0%93%D1%83%D1%81%D0%B5%D0%BD%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D1%8C) транспортных средствах, и в других машинах, в которых применение [зубчатых передач](https://ru.wikipedia.org/wiki/%D0%97%D1%83%D0%B1%D1%87%D0%B0%D1%82%D0%B0%D1%8F_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0) является неподходящим. Они выполняют функцию передачи вращательного движения между двумя валами посредством [цепной передачи](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BF%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0) или функцию сообщения линейного движения звеньям гусениц.

При использовании звёздочек в велосипедах можно изменять передаточное отношение цепной передачи путём изменения диаметра звёздочек (а значит, и количества зубьев). Такие звёздочки являются основой [велосипедного переключателя скоростей.](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BB%D0%BE%D1%81%D0%B8%D0%BF%D0%B5%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B5%D1%80%D0%B5%D0%BA%D0%BB%D1%8E%D1%87%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_%D1%81%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1%82%D0%B5%D0%B9)

Изображение звездочки цепной передачи с обозначенными параметрами Rокр (радиус внешней окружности), Rц (радиус цилиндра), r (радиус отверстия), S (толщина внешней части звездочки), Sц (толщина цилиндра), H (глубина шпоночной выемки), n (количество зубьев) приведено на рисунке 1.1.

Звездочки цепных передач по радиальному и осевому биению должны отвечать требованиям соответствующих стандартов - ГОСТ 591, ГОСТ 592 [3].

В таблице 1.1 представлены некоторые размеры профиля зубьев и венцов звездочек с параметрами z (число зубьев звездочки), dд (диаметр делительной окружности), De (диаметр окружности выступов), Di (диаметр окружности впадин), Dc (диаметр обода), Lx (наибольшая хорда (для контроля звездочек с нечетным числом зубьев)).

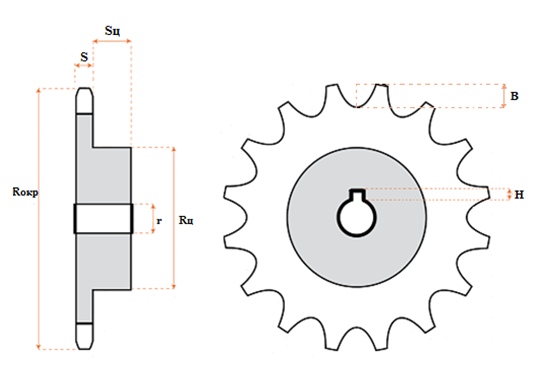


Рисунок 1.1 – Чертеж звездочки цепной передачи

Таблица 1.1 – Размеры профиля зубьев и венцов звездочек.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение цепи по ГОСТ 13568-97, ГОСТ 21834-87 | z | dд | De | Di | Dc | Lx для профиля зуба | |
| без смещения центров дуг впадин | со смещением центров дуг впадин |
| ПР-8-460 | 16 | 41,01 | 44,6 | 35,88 | 31 | - | - |
| 17 | 43,54 | 47,2 | 38,41 | 34 | 38,23 | 38,21 |
| 18 | 46,07 | 49,8 | 40,94 | 36 | - | - |
| 19 | 48,60 | 52,3 | 43,48 | 39 | 43,31 | 43,3 |
| 20 | 51,14 | 54,9 | 46,02 | 41 | - | - |
| 21 | 53,68 | 57,5 | 48,55 | 44 | 48,4 | 48,38 |
| 22 | 56,21 | 60,0 | 51,09 | 46 | - | - |
| 23 | 58,75 | 62,6 | 53,63 | 49 | 53,49 | 53,47 |
| 24 | 61,29 | 65,2 | 56,16 | 52 | - | - |
| 25 | 63,83 | 67,7 | 58,7 | 54 | 58,58 | 58,56 |

## **Описание аналогов**

### **1.3.1 Дополнительный модуль КОМПАС-3D «Валы и механические передачи».**

Приложение предназначено для автоматизации проектирования и построения трехмерных моделей валов, втулок, элементов механических передач и различных конструктивных элементов в среде КОМПАС-3D [4].

Средствами приложения Валы и механические передачи 3D могут быть спроектированы следующие элементы механических передач:

1. шестерни цилиндрические с внешним и внутренним зацеплением;
2. шестерни цилиндрические винтовых эвольвентных передач;
3. зубчатые рейки;
4. шестерни конические с прямым зубом;
5. шкивы клиноременных передач;
6. звездочки приводных роликовых цепей;
7. червяки и червячные колёса (цилиндрическая червячная передача);
8. зубчатые глухие муфты.

Пользователям доступны геометрические и проектные расчеты, расчеты передач на прочность и долговечность, а также оптимизационные расчеты.

На рисунке 1.2 представлено окно программы КОМПАС-3D с дополнительной библиотекой «Валы и механические передачи».

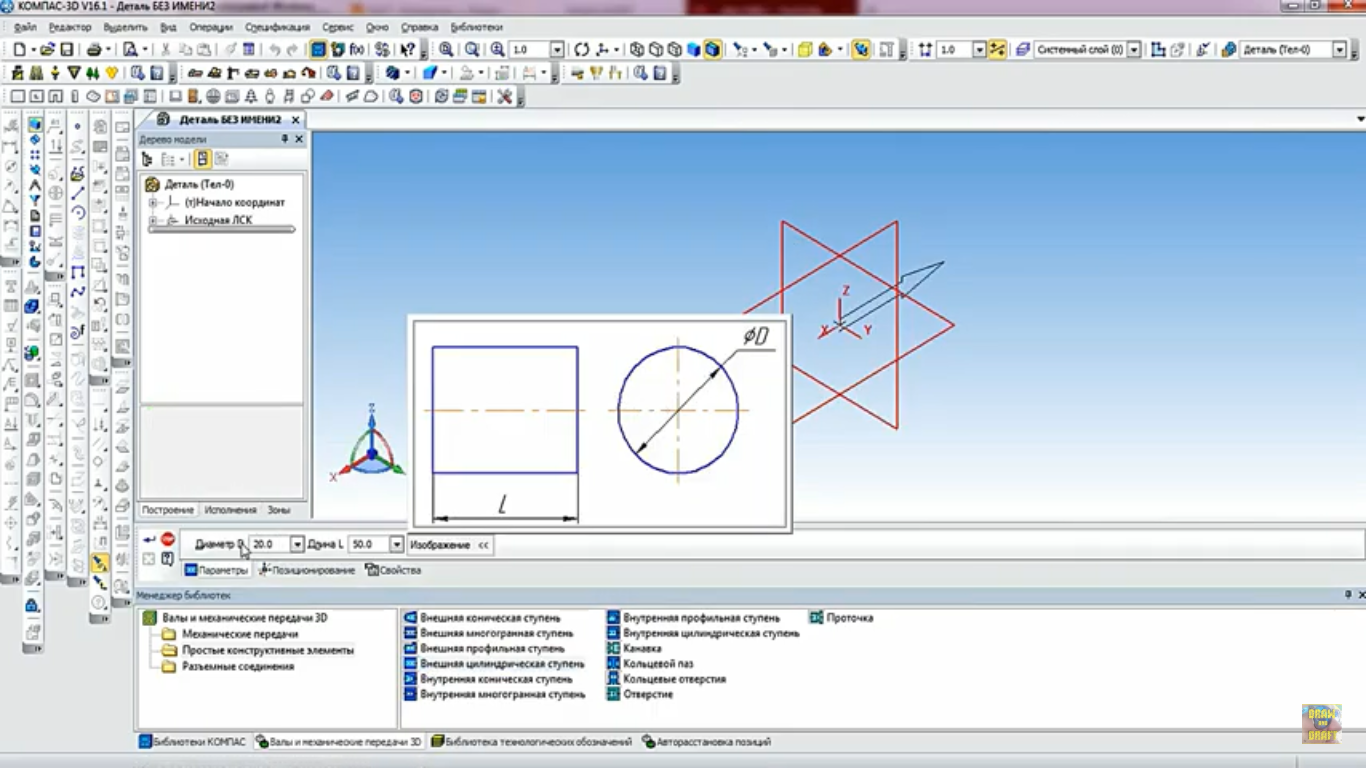


Рисунок 1.2 – Окно программы КОМПАС-3D

### **1.3.2 MechaniCS**

MechaniCS — приложение к AutoCAD или Autodesk Inventor, предназначенное для оформления чертежей в соответствии с ЕСКД, проектирования систем гидропневмоэлементов, зубчатых зацеплений, валов, инженерного анализа, расчета размерных цепей, создания пользовательских библиотек [5].

MechaniCS обеспечивает специалиста всем необходимым для проектирования машиностроительных объектов: более чем двумя тысячами стандартов (включая ГОСТ, ОСТ, DIN и ISO) и унифицированными компонентами, возможностью создавать собственные интеллектуальные объекты, выполнять инженерные расчеты с отображением результатов на модели, оформлять проекции чертежей по ЕСКД и многим другим.

Все детали общей конструкторско-технологической базы обладают интеллектом и являются объектно-зависимыми. При изменении параметров одной детали все связанные с ней объектно-зависимые детали изменятся автоматически, причем в соответствии с их параметрами в базе. Такая технология — мощный инструмент многовариантного проектирования, залог повышения качества выпускаемых проектов. Важно, что этот подход одинаково доступен пользователям AutoCAD и Autodesk Inventor.

MechaniCS дает конструктору возможность учитывать не только геометрические параметры стандартных элементов, но и их механические свойства. На объекты в сборочных чертежах (при использовании AutoCAD) можно накладывать геометрические и параметрические зависимости, использовать предустановленные зависимости при их размещении на чертеже.

Проектирование элементов передач:

1. Валы.
2. Шаблоны подшипниковых узлов. Шаблоны — это группа деталей с наложенными на них параметрическими и сборочными зависимостями. Геометрия деталей в шаблоне зависит от значений их параметров в базе, размеров связанных деталей, а также от результатов расчетов (например, для зубчатых колес).
3. Шаблоны схем редукторов. В единой базе данных можно хранить 2D-развертки схем редукторов, выполненных в AutoCAD, и открывать их в Autodesk Inventor уже как трехмерные объекты.

Библиотека состоит из более чем 2000 стандартных и унифицированных элементов.

На рисунке 1.3 представлено приложение MechaniCS.

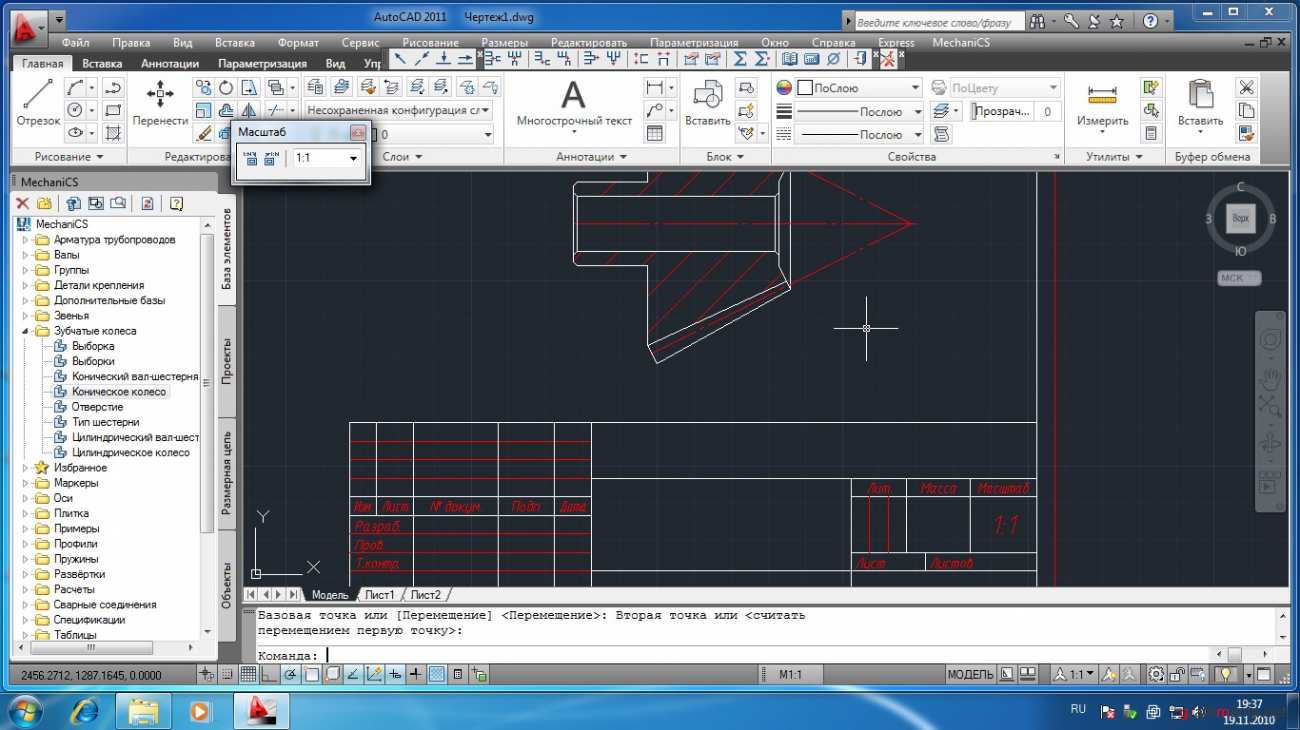


Рисунок 1.3 – Приложение MechaniCS

## **Выбор инструментов и средств реализации**

В связи с требованием технического задания плагин выполнен на языке C# в среде Visual Studio Community 2019 версия 16.3.0 с применением Microsoft.NET Framework 4.8.03761, для системы КОМПАС 3D V16. Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов среды Microsoft Visual Studio 2019 [6] с помощью библиотек NUnit версия 3.12.0, NUnit3TestAdapter версия 3.15.1.

Для реализации пользовательского интерфейса использовался WinForm.

Взаимодействие плагина с системой КОМПАС (с функциями моделирования, математическими функциями ядра системы и пр.) осуществляется посредством программных интерфейсов, называемых API. В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга [7].

В основном, для создания полноценных подключаемых модулей, достаточно методов и свойств интерфейсов API 5.

# **Описание реализации**

Для описания архитектуры, пользовательского сценария системы был выбран унифицированный язык моделирования (UML). Унифицированный язык моделирования (UML) – это семейство графических нотаций, в основе которого лежит единая метамодель [8]. Он помогает в описании и проектировании программных систем, в особенности систем, построенных с использованием объектно-ориентированных технологий. На основе UML построены: диаграммы вариантов использования, диаграммы классов.

В процессе разработки плагина диаграмма вариантов использования и диаграмма классов были изменены(дополнены). Изменения были связаны с тем, что не было навыков в разработке плагинов для САПР, нет большого опыта в разработке приложений и составления для них UML диаграмм. А также изменения связаны с дополнением функционала для плагина заказчиком.

## **Диаграмма прецедентов класса**

Прецеденты – это технология определения функциональных требований к системе [8]. Работа прецедентов заключается в описании типичных взаимодействий между пользователем системы и самой системой и предоставлении описания процесса ее функционирования. В терминах прецедентов пользователи называются актерами. Прецеденты считаются важной частью языка UML и предоставляют собой ценный инструмент для понимания функциональных требований к системе.

Изначальная версия диаграммы прецедентов класса представлена на рисунке 2.1.

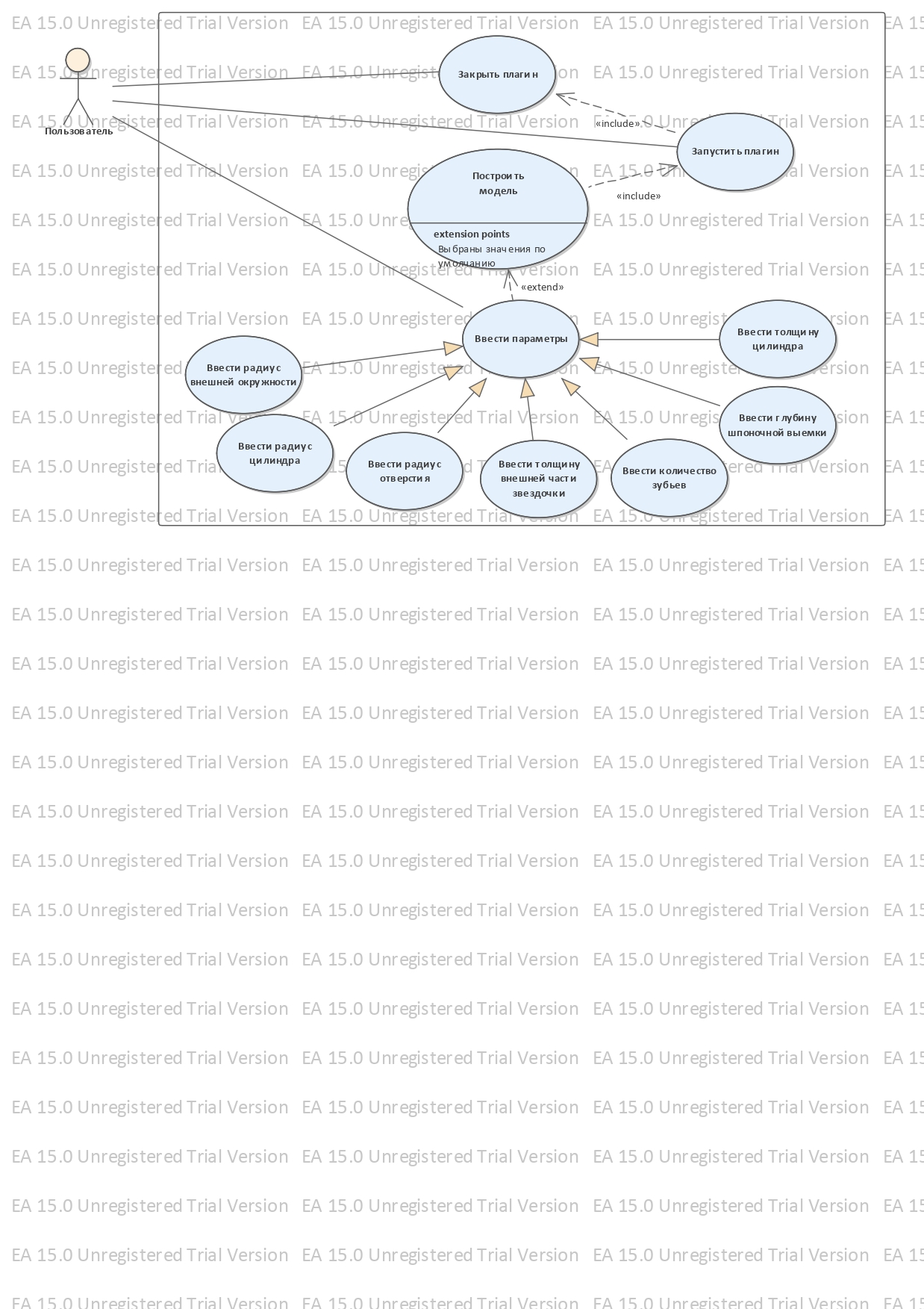


Рисунок 2.1 – Изначальная диаграмма прецедентов класса

На рисунке 2.2 представлена измененная диаграмма прецедентов класса.

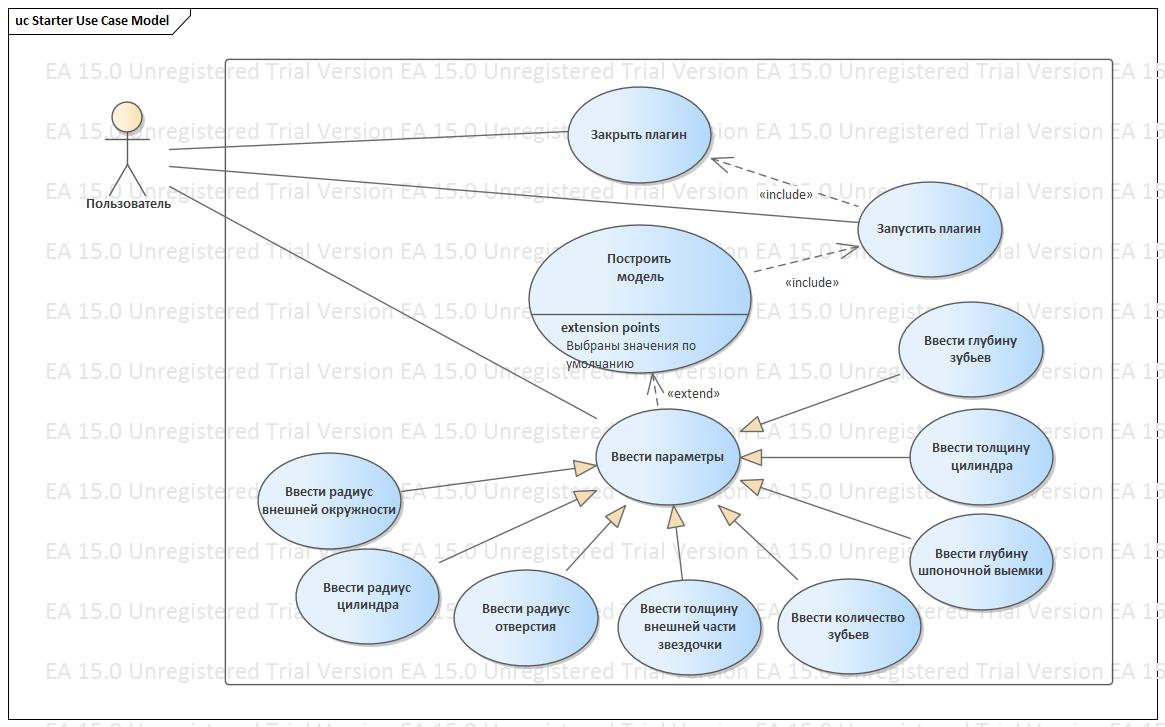


Рисунок 2.2 – Диаграмма прецедентов класса

Действующим лицом является пользователь. Он может выполнять следующие действия: открывать/закрывать плагин, вводить параметры модели, строить модель звездочки цепной передачи.

В результате изменений, связанных с включением в плагин дополнительной функциональности, а именно, ввод глубины зубьев, изначальная версия диаграммы прецедентов класса была изменена.

## **2.2 Диаграмма классов**

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними [8]. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами. В UML термин функциональность применяется в качестве основного термина, описывающего и свойства, и операции класса. Целью создания диаграммы классов является графическое представление статической структуры декларированных элементов системы.

На рисунке 2.3 представлена изначальная версия диаграммы классов.

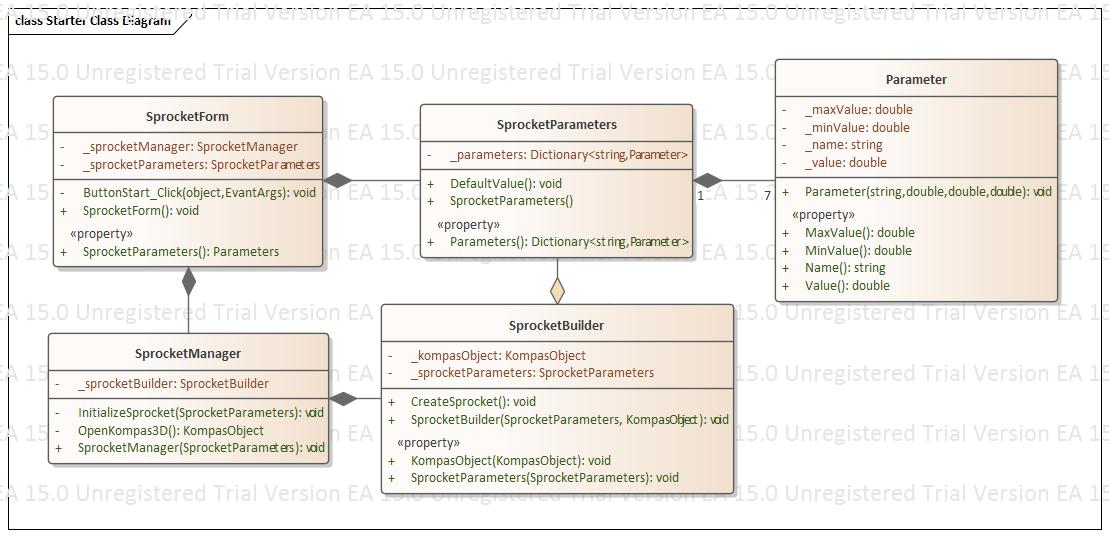


Рисунок 2.3 – Изначальная диаграмма классов

На рисунке 2.4 представлена измененная диаграмма классов плагина.

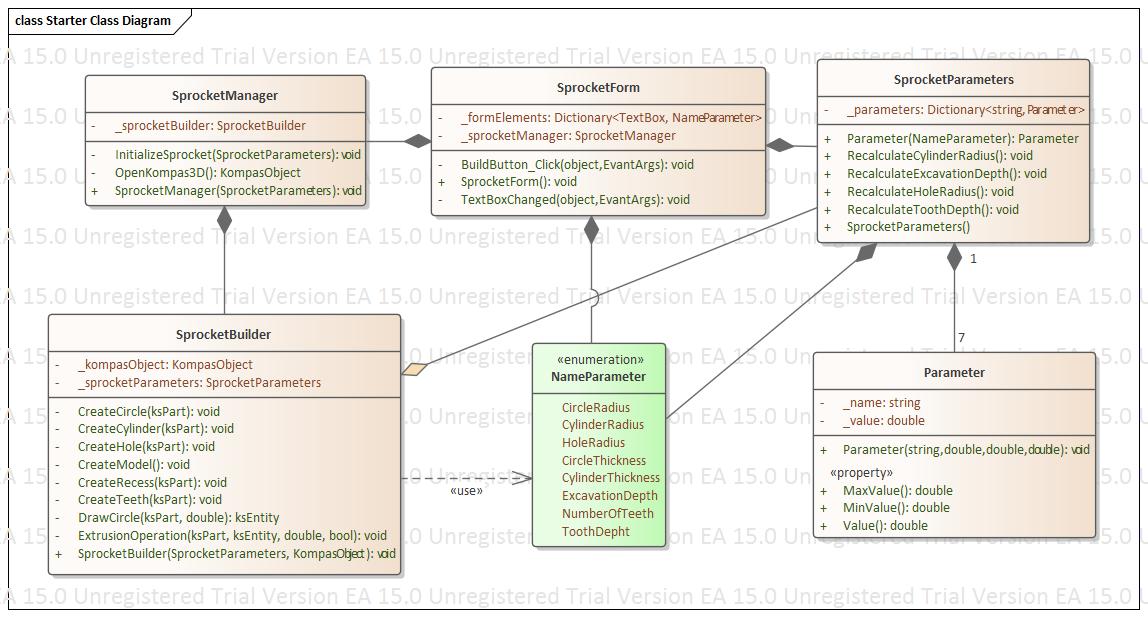


Рисунок 2.4 – Измененная диаграмма классов

Класс Parameter хранит информацию о параметре проектируемой модели звездочки.

Класс SprocketParameters хранит словарь со всеми параметрами модели звездочки.

Класс SprocketBuilder реализует методы построения модели.

Класс SprocketManager хранит экземпляр класса SprocketBuilder, реализует связь плагина с Компас3D.

Класс SprocketForm является формой пользовательского интерфейса.

В результате изменений функциональности плагина и не достаточности опыта в разработке приложений диаграмма классов подверглась изменениям.

Было добавлено перечисление ParametersName, чтобы уменьшить вероятность ошибки при введении названий параметров. А также добавлены некоторые методы в существующие классы.

# **Описание программы для пользователя**

Плагин состоит из одного диалогового окна, разделенного на две части. В одной вводятся параметры модели звездочки цепной передачи и находится кнопка «Построить», в другой представлен чертеж самой модели. Вид диалогового окна при запуске плагина представлен на рисунке 3.1.

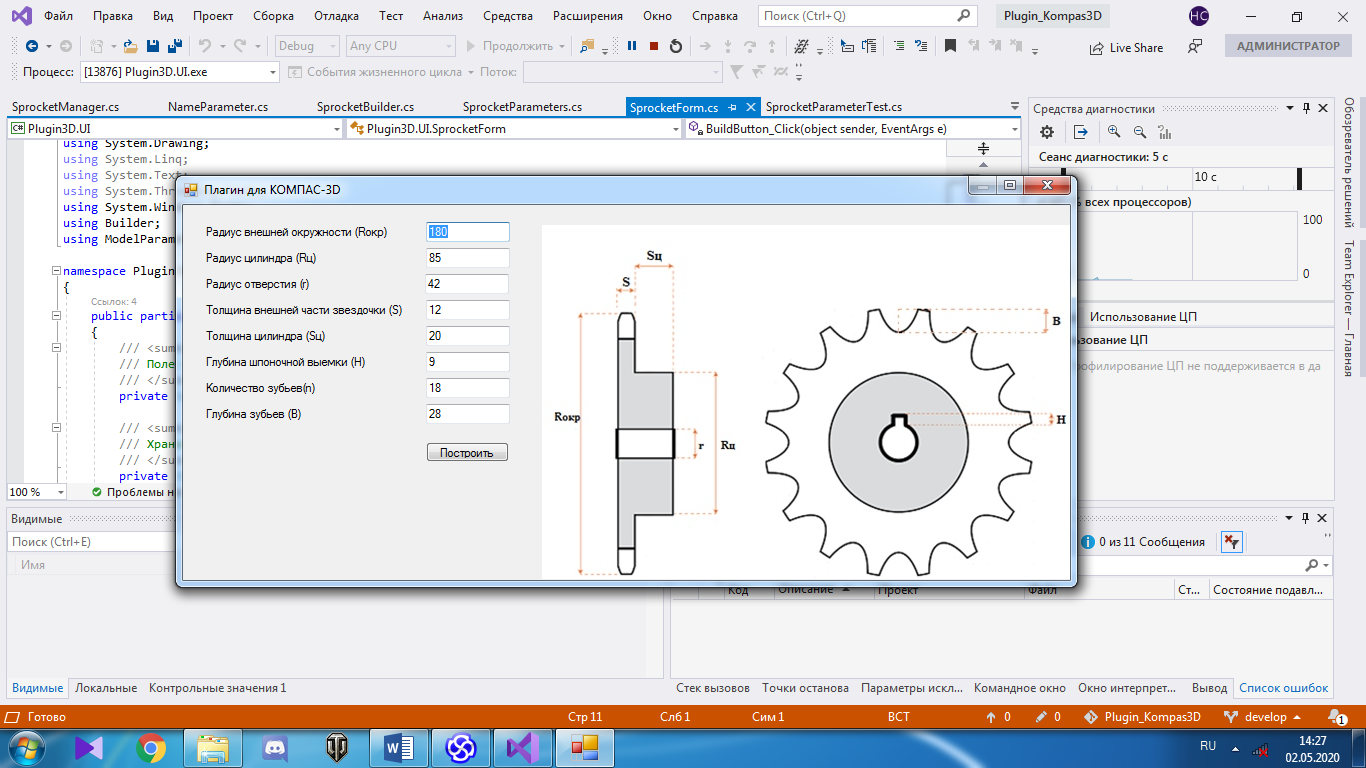


Рисунок 3.1 – Вид диалогового окна при запуске плагина

При введении некорректного значения параметра, поле будет окрашиваться в красный цвет и будет оставаться красным до тех пор, пока не будет введено корректное значение, после чего поле перекрасится в зеленый цвет. Данная функция работает на всех полях и при всех вводах параметров. Если значение будет не исправлено на корректное и нажата кнопка «Построить», некорректные значения будут принимать значения равные максимальным возможным значениям параметра при построении модели.

При изменении параметра «Радиус внешней окружности» происходит перерасчет интервала возможных значений у параметра «Радиус цилиндра» и параметра «Глубина зубьев».

При изменении параметра «Радиус цилиндра» происходит перерасчет интервала возможных значений у параметра «Радиус отверстия».

При изменении параметра «Радиус отверстия» происходит перерасчет интервала возможных значений у параметра «Глубина шпоночной выемки».

Параметры «Радиус внешней окружности», «Толщина внешней части звездочки», «Толщина цилиндра», «Количество зубьев» не зависят от других параметров.

Нажатие кнопки “Построить” возможно в любой момент времени заполнения параметров, начиная с момента запуска программы, потому что изначально заданы значения по умолчанию. Нажатие кнопки открывает САПР «КОМПАС – 3D» и создает новый пустой документ типа “Деталь”, в котором производиться построение модели, если САПР «КОМПАС – 3D» уже открыта, то произойдет создание только нового документа типа “Деталь”. На рисунке 3.2 изображено диалоговое окно документа САПР «КОМПАС – 3D» после нажатия кнопки «Построить» и построения модели.

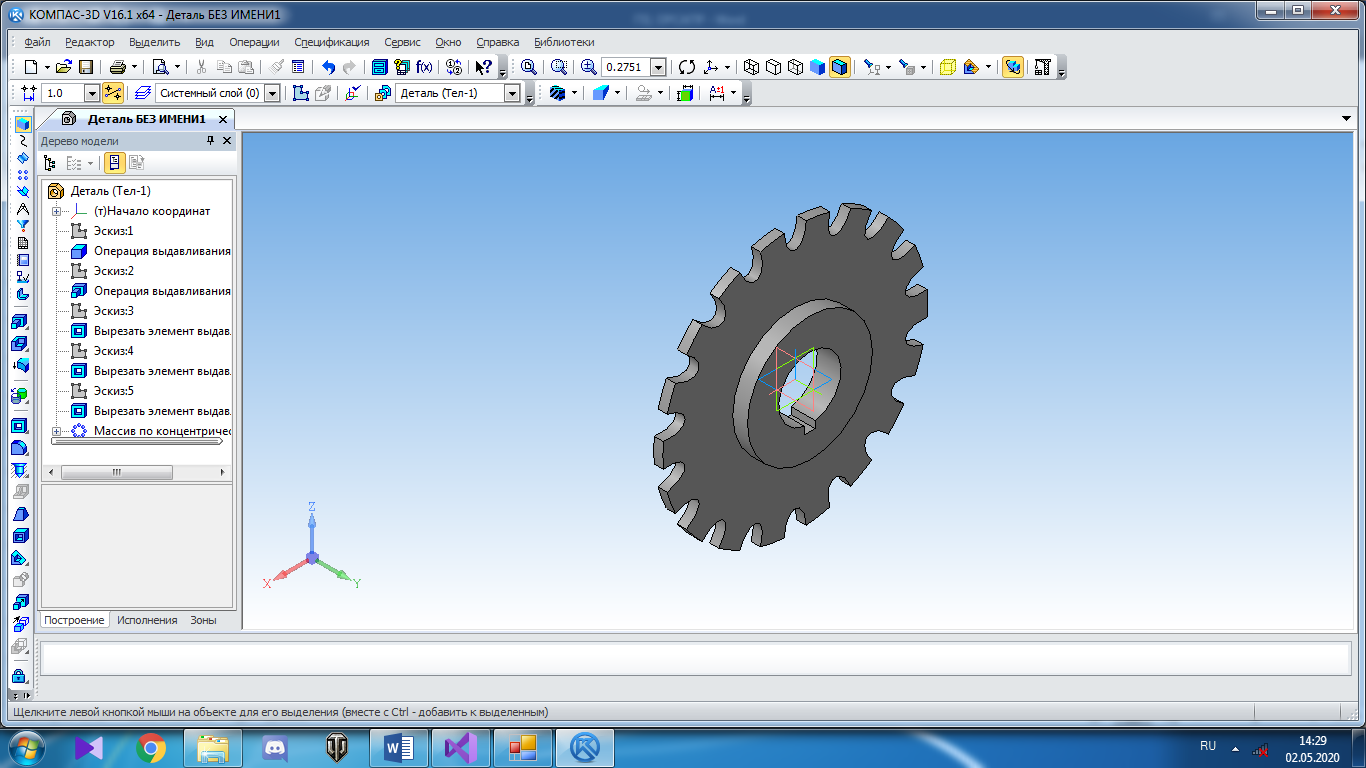


Рисунок 3.2 – Диалоговое окно документа САПР «КОМПАС – 3D»

# **Тестирование программы**

Тестирование — это проверка соответствия объекта желаемым критериям[9]. Несоответствие объекта желаемым критериям называется ошибкой. В данном проекте будет применятся три вида тестирования: функциональное тестирование, модульное тестирование, нагрузочное тестирование.

## **Функциональное тестирование**

Функциональное тестирование — тестирование функциональности объекта, т. е. правильно ли объект выполняет свои функции[9]. Фактически, выполняется проверка правильности выходных данных при соответствующих входных данных. В данном случае будет проверяться правильность построения детали при различных входных параметрах.

На рисунке 4.1 представлена модель звездочки цепной передачи с минимальными значениями.

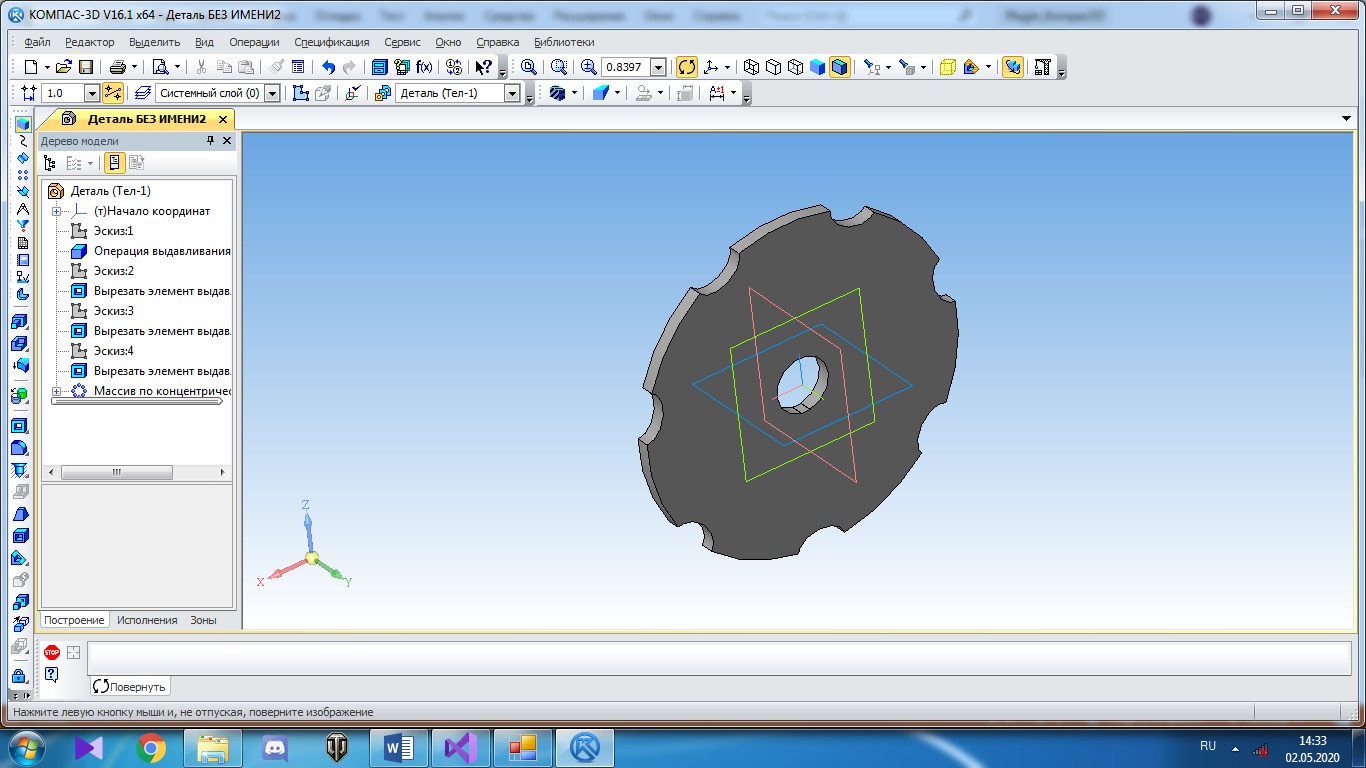


Рисунок 4.1 – Модель звездочки цепной передачи с минимальными значениями

На рисунке 4.2 представлена модель звездочки цепной передачи с максимальными значениями.

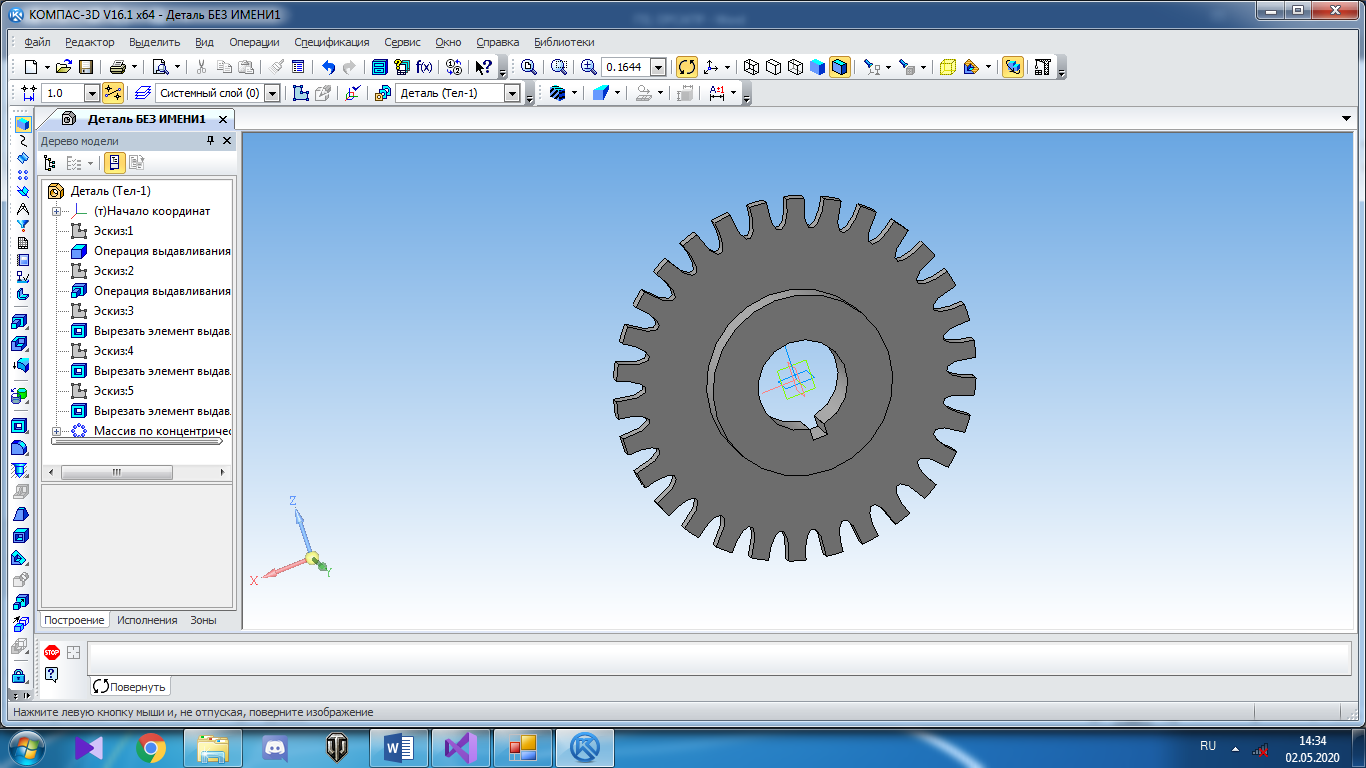


Рисунок 4.2 – Модель звездочки цепной передачи с максимальными значениями

На рисунке 4.3 представлена модель звездочки цепной передачи со значениями, введенными по умолчанию

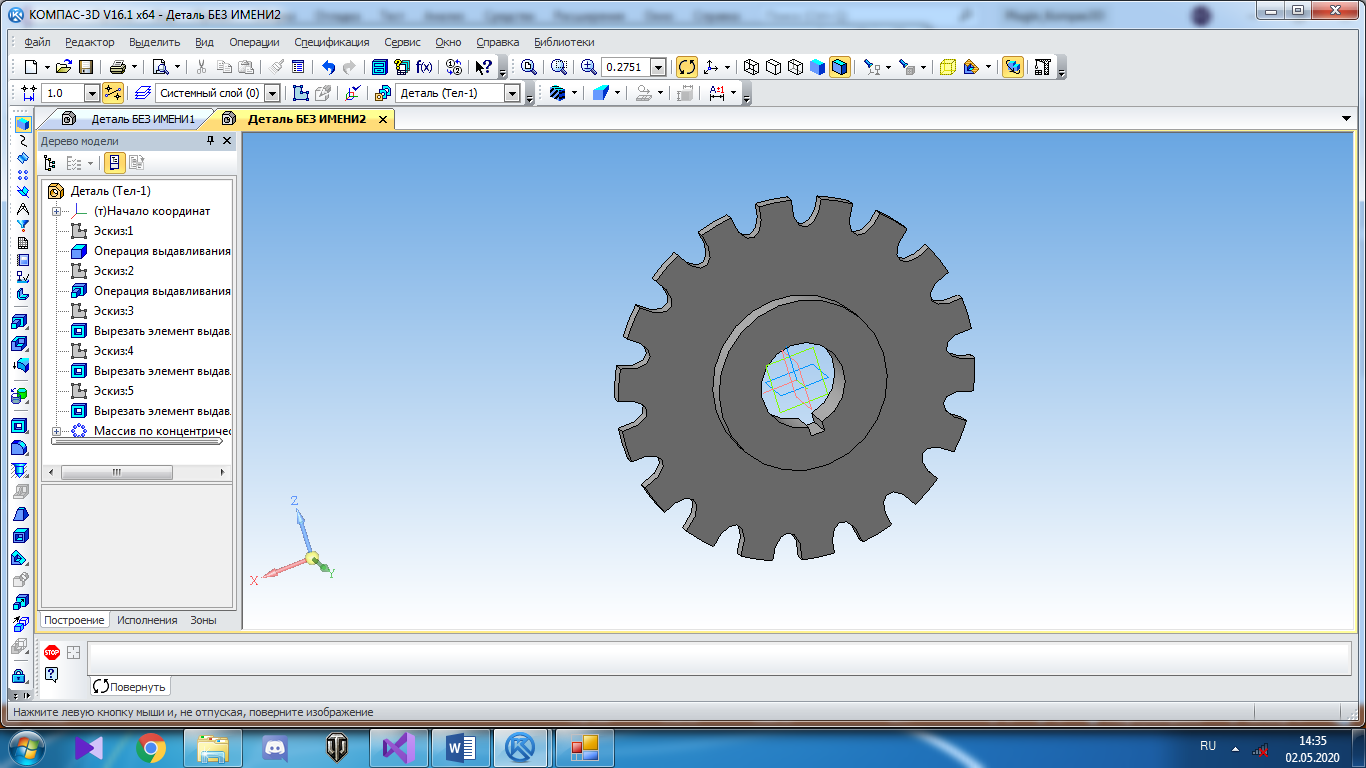


Рисунок 4.3 – Модель звездочки цепной передачи со значениями по умолчанию

На рисунке 4.4 представлено диалоговое окно при вводе некорректного значения параметра.

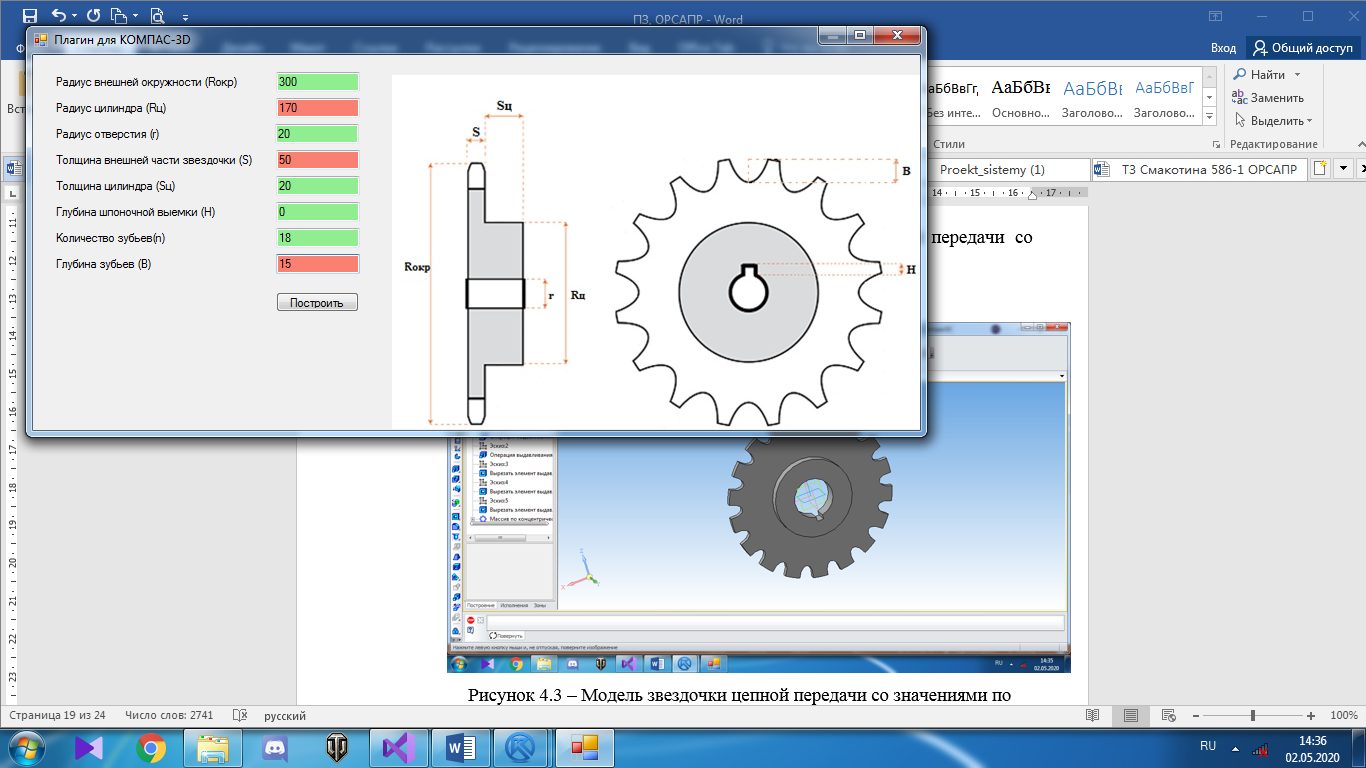


Рисунок 4.4 – Диалоговое окно при некорректном вводе параметра

## **Модульное тестирование**

Юнит-тестирование (модульное тестирование) — тестирование отдельного элемента изолированно от остальной системы [9]. На основе тестовых сценариев проводилось тестирование работы методов и свойств классов при помощи обозревателя тестов Visual Studio. Проверялись открытые поля и методы, для этого были созданы тестовые классы:

1. SprocketParameterTest. Класс, тестирующий свойства и методы класса SprocketParameters.
2. ParameterTest. Класс, тестирующий свойства и методы класса Parameter.

В таблице 4.1 представлены все тестовые сценарии.

Таблица 4.1 – Тестовые сценарии

|  |  |  |
| --- | --- | --- |
| Название тестового класса | Название тестового метода | Назначение |
| ParameterTest | TestMinValueGet\_CorrectValue() | Метод для проверки корректности возврата минимального значения |
| TestMinValueSet\_CorrectValue() | Метод для проверки корректности записи минимального значения |
| TestMaxValueGet\_CorrectValue() | Метод для проверки корректности возврата максимального значения |
| TestMaxValueSet\_CorrectValue() | Метод для проверки корректности записи максимального значения |

Окончание таблицы 4.1

|  |  |  |
| --- | --- | --- |
|  | TestValueGet\_CorrectValue() | Метод для проверки корректности возврата текущего значения |
| TestValueSet\_CorrectValue() | Метод для проверки корректности записи текущего значения |
| TestParameterConstructor() | Метод для проверки корректности работы конструктора |
| TestValueSet\_ArgumentExeption(string wrongValue, string message) | Метод для проверки вызова исключения при записи некорректных значения |
| SprocketParameterTest | TestParameter() | Метод для проверки корректности возвращаемых методом Parameter данных |
| TestRecalculateCylinderRadius() | Метод для проверки корректности расчета радиуса цилиндра |
| TestRecalculateHoleRadius() | Метод для проверки корректности расчета радиуса отверстия |
| TestRecalculateExcavationDepth() | Метод для проверки корректности расчета глубины шпоночной выемки |
| TestToString() | Метод для проверки корректности преобразования элемента перечисления в строку |
| TestSprocketParameters() | Метод для проверки корректности работы конструктора |
| TestRecalculateToothDepth() | Метод для проверки корректности расчета глубины зубьев |

На рисунке 4.5 изображено диалоговое окно запущенных тестов с успешным прохождением.

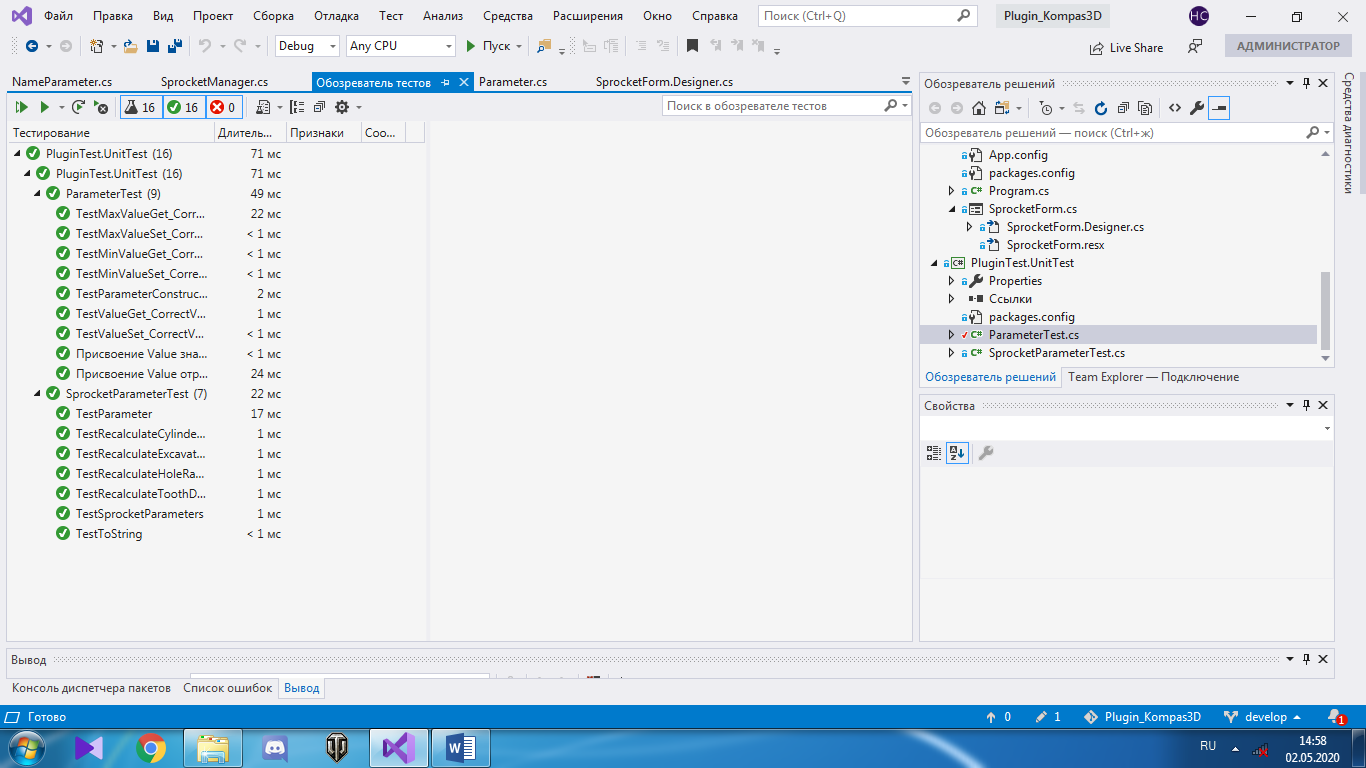


Рисунок 4.5 – Прохождение модульного тестирования

## **Нагрузочное тестирование**

Нагрузочное тестирование — 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)[9].

После построения 104 детали программа завершилась со сбоем, т.к. закончилась память. На момент сбоя количество потребляемой памяти было около 1457 МБ.

График потребляемой оперативной памяти относительно числа созданных деталей изображен на рисунке 4.6. Где ось ОУ – количество используемой оперативной памяти, в МБ, а OX – число построенных деталей.

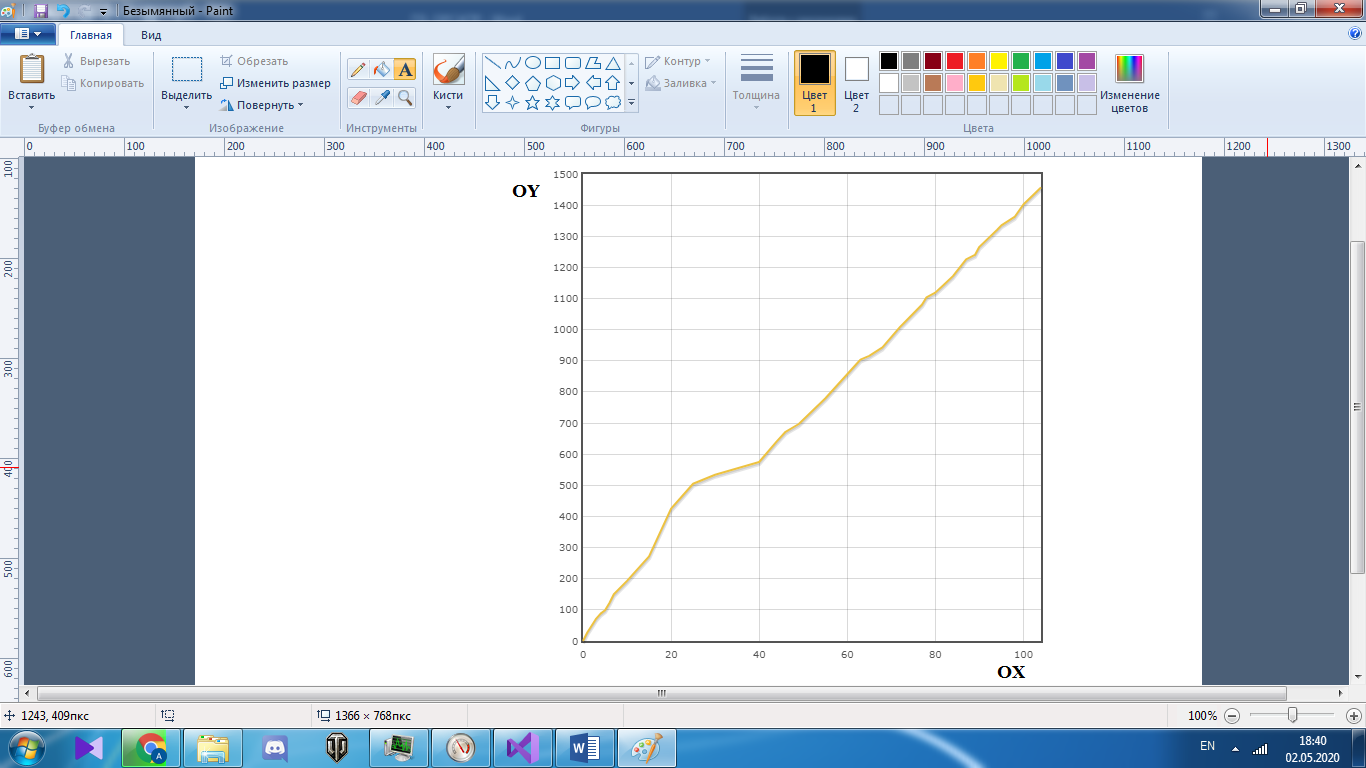


Рисунок 4.6 – График использования оперативной памяти относительно числа построенных деталей

График зависимости загрузки программой центрального процессора от количества построенных деталей изображен на рисунке 4.7. Где OY – загрузка центрального процессора в %, а OX – число построенных деталей.

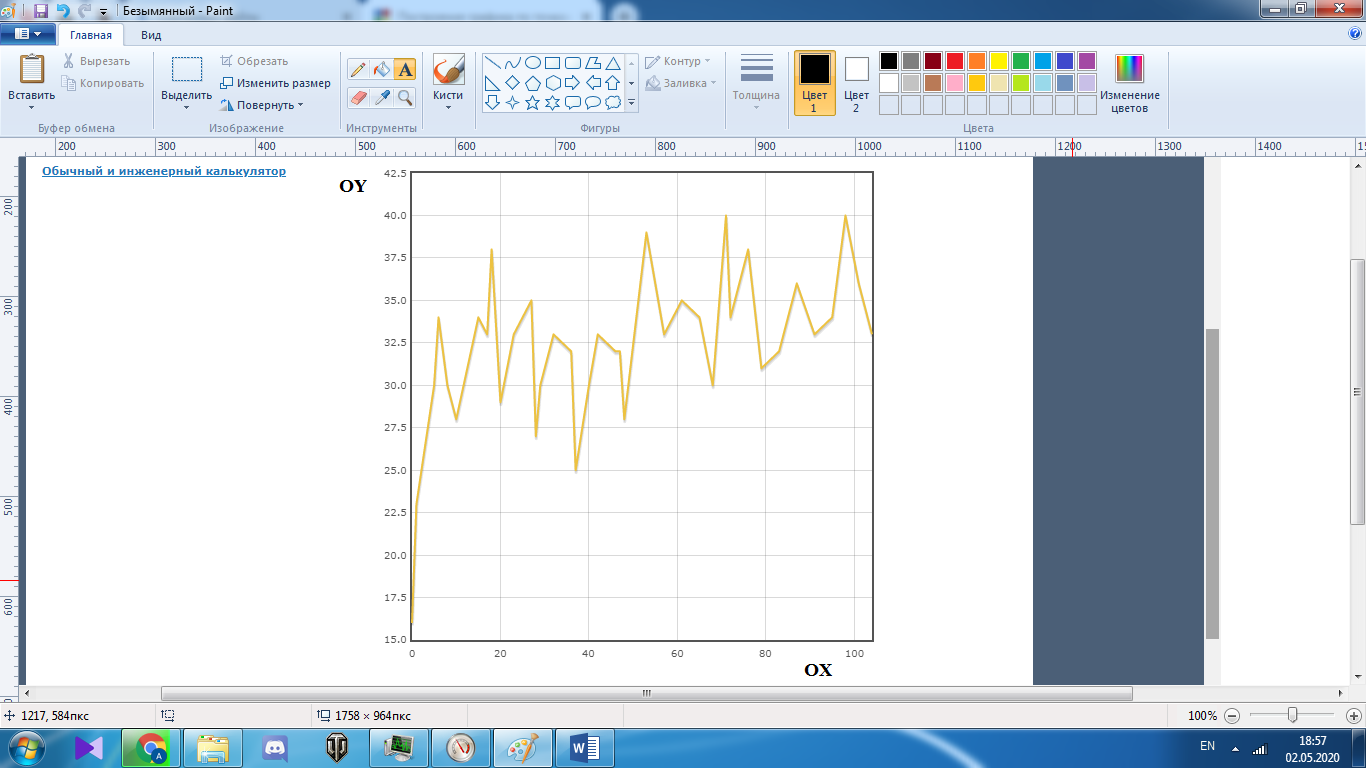


Рисунок 4.7 – График использования ресурсов центрального процессора относительно числа построенных деталей

Из всего можно сделать выводы, что при наличии свободных ресурсов – зависимость прямая. То есть, используемая оперативная память и нагрузка на ЦП, при увеличении количества деталей, увеличивается линейно.

# **Заключение**

В ходе выполнения лабораторных работ были изучены основные этапы проектирования программного продукта, изучена предметная область предмета проектирования, также было изучено API системы автоматизированного проектирования КОМПАС-3D.

В результате полученные знания были применены для реализации плагина для автоматизации построения модели «Звездочки цепной передачи» в рабочей плоскости программы КОМПАС-3D, проведены функциональные, модульные и нагрузочные тесты.

# **Список используемых источников**

1. Компас-3D [электронный ресурс]. – режим доступа: <https://kompas.ru/kompas-3d/about/> (дата обращения 24.02.2020)
2. Звездочка [электронный ресурс]. – режим доступа: <https://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%91%D0%B7%D0%B4%D0%BE%D1%87%D0%BA%D0%B0_(%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)> (дата обращения 24.02.2020)
3. ГОСТ 591 [электронный ресурс]. – режим доступа: <http://docs.cntd.ru/document/gost-591-69> (дата обращения 24.02.2020)
4. Валы и механические передачи 3D [электронный ресурс]. – режим доступа: <https://kompas.ru/kompas-3d/application/machinery/shafts-3d> (дата обращения 27.02.2020)
5. MechaniCS [электронный ресурс]. – режим доступа: <https://cad.ru/support/bz/archive/62/mechanics/> (дата обращения 27.02.2020)
6. Microsoft Visual Studio 2019 [электронный ресурс]. – режим доступа: <https://visualstudio.microsoft.com/ru/vs/> (дата обращения: 30.04.2020)
7. Базовые интерфейсы API системы КОМПАС [электронный ресурс]. – режим доступа: <https://it.wikireading.ru/23741> (дата обращения 24.02.2020)
8. Мартин Фаулер. UML. Основы. / Мартин Фаулер; пер. с англ. А. Петухова – 3-е издание. – Спб: Символ-Плюс, 2004 – 192с.
9. Виды тестирования [электронный ресурс]. – режим доступа: <https://qa-academy.by/qaacademy/news/klassifikaciya-vidov-testirovaniya/> (дата обращения: 02.05.2020)