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 «Клапан ДВС» ДЛЯ «КОМПАС 3D V20»**

Пояснительная записка по лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 588-3

\_\_\_\_\_\_\_\_\_Лыспак Н.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП:

\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2021 г.

Томск 2021

# Реферат

Лабораторная работа, 19 страниц, 15 рисунков, 7 источников

Ключевые слова: САПР, КОМПАС-3D, API, ПЛАГИН, РАЗРАБОТКА, КЛАПАН.

Целью данной работы является разработка плагина для создания трёхмерных моделей тарельчатого клапана, согласно заданным параметрам, для системы автоматизированного проектирования «КОМПАС-3D»

Отчёт по лабораторной работе выполнен в текстовом редакторе Microsoft Word 2016.

**Оглавление**

[Реферат 2](#_Toc92221127)

[1. Описание САПР 4](#_Toc92221128)

[1.1. Описание программы 4](#_Toc92221129)

[1.2. Описание API 4](#_Toc92221130)

[1.3. Обзор аналогов 9](#_Toc92221131)

[2. Описание предмета проектирования 10](#_Toc92221132)

[3. Описание плагина 11](#_Toc92221133)

[3.1 Диаграмма классов 11](#_Toc92221134)

[3.2 Макет пользовательского интерфейса 13](#_Toc92221135)

[4. Тестирование плагина 15](#_Toc92221136)

[4.1 Функциональное тестирование 15](#_Toc92221137)

[4.2 Модульное тестирование 18](#_Toc92221138)

[4.3 Нагрузочное тестирование 19](#_Toc92221139)

[Заключение 21](#_Toc92221140)

[Список литературы 22](#_Toc92221141)

# 1. **Описание САПР**

## **1.1. Описание программы**

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий и сотни тысяч профессиональных пользователей, благодаря сочетанию проектирования изделий любой сложности и простоты освоения. Особенностью продукта является использование российского геометрического ядра C3D и собственных программных технологий. КОМПАС-3D обеспечивает поддержку файлов наиболее распространённых CAD-систем (SolidWorks, Autodesk Inventor, Solid Edge, Creo, NX, Catia), что позволяет организовывать совместную работу со смежными организациями и заказчиками, использующими другие программные продукты.[1]

## 1.2. Описание API

В КОМПАС-3D на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда, очевидно, что обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно с помощью экспортной функции CreateKompasObject(). Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа). [2]

Ниже в таблице 1.1 представлены свойства и методы интерфейса KompasObject, которые будут использованы при разработке плагина.

Таблица 1.1 – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трёхмерного документа (детали или сборки) |
| GetMathematic2D() | ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями в графическом документе |
| GetDynamicArray(long type) | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| GetParamStruct (short structType) | StructType2D | Метод для получения указателя на интерфейс структуры параметров объекта нужного типа |
| visible | bool | Cвойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksDocument3D, которые будут использованы при разработке плагина

Таблица 1.2 – Некоторые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool typeDoc) | bool | Метод для создания документа-модели (детали или сборки) |
| UpdateDocumentParam() | bool | Метод для активизации изменённых параметров продукта |
| GetPart(int type) | ksPart | Метод для получения указателя на интерфейс компонента в соответствии с заданным типом |

В таблице 1.3 представлены методы интерфейса ksPart, которые будут использованы при разработке плагина

Таблица 1.3 – Некоторые методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection (short objType) | ksEntityCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity (short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity (short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.4 представлены методы интерфейса ksEntity, которые будут использованы при разработке плагина

Таблица 1.4 – Некоторые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnknown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблицах 1.5 и 1.6 представлены методы и свойства интерфейса ksPlaneOffsetDefinition, которые будут использованы при разработке плагина

Таблица 1.5 – Свойства интерфейса ksPlaneOffsetDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| direction | bool | Направление смещения от базовой плоскости |
| offset | double | Смещение (расстояние) от базовой плоскости |

Таблица 1.6 – Методы интерфейса ksPlaneOffsetDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных данных | Тип выходных данных | Описание |
| SetPlane | plane - указатель на интерфейс плоскости ksEntity или IEntity. | TRUE – в случае успешного завершения | Изменить указатель на интерфейс базовой плоскости |
| GetPlane | - | ksEntity | Получить указатель на интерфейс базовой плоскости |

## 1.3. Обзор аналогов

**MechaniCS**

CSoft Development MechaniCS представляет собой 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 и создания пользовательских библиотек.

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.

Продукт прост и понятен в освоении. Приступить к его использованию можно буквально сразу после его установки. [3]

# 2. Описание предмета проектирования

Предметом проектирования является модель клапана ДВС. Данная модель имеет 9 основных параметров

1. Длина клапана = 50 мм ≤ А ≤ 150 мм
2. Диаметр ножки клапана = 5 мм ≤B≤15 мм
3. Ширина паза под сухарь = 1 мм ≤ С ≤ 0.1A мм
4. Глубина паза под сухарь = 0.5 мм ≤ D ≤ 0.25B мм
5. Расстояние до паза под сухарь = 5 мм ≤ E ≤ 0.25A мм
6. Диаметр тарелки клапана = 2B мм ≤ F ≤ 70 мм
7. Толщина тарелки клапана = 1 мм ≤ G ≤ 0.75F мм
8. Длина рабочей фаски = 2 мм ≤ I ≤ 10 мм
9. Радиус плавного перехода = 5 мм ≤ H ≤ 0.75F мм

На рисунке 2.1 представлен чертёж клапана ДВС.



Рисунок 2.1 – Чертёж клапана ДВС.

# 3. Описание плагина

## 3.1 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.[4] Целью создания диаграммы классов является графическое представление статической структуры элементов системы.

Изначальная диаграмма классов плагина представлена на рисунке 3.1.

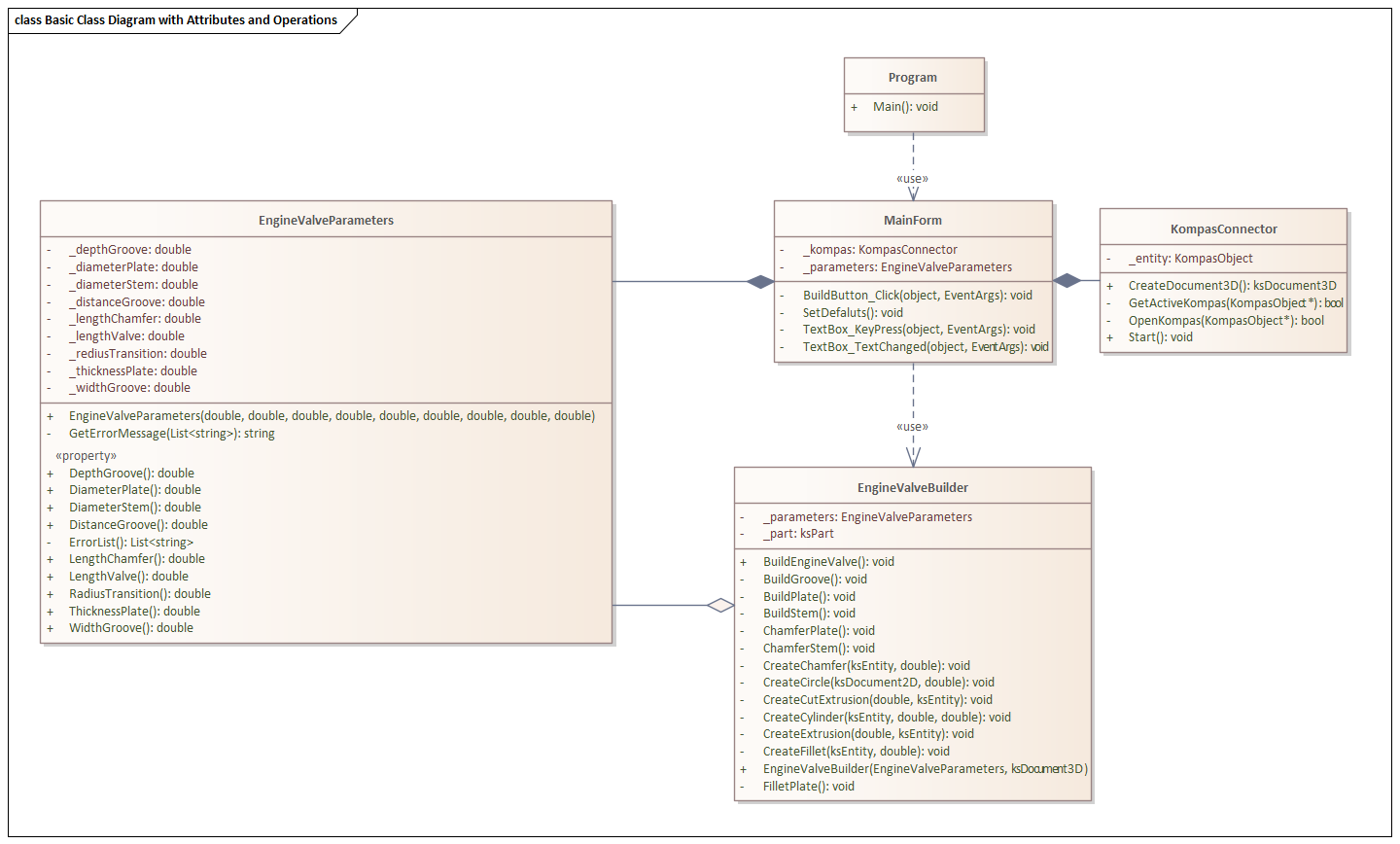


Рисунок 3.1 – Диаграмма классов плагина «Клапан ДВС»

* MainForm – является главным элементом управления для обработки действий в графическом интерфейсе;
* EngineValveParameters – содержит параметры клапана для построения;
* EngineValveBuilder – выполняет построение детали;
* KompasConnector – класс для связи с КОМПАС-3D.

Диаграмма классов после добавления дополнительной функциональности представлена на рисунке 3.2.

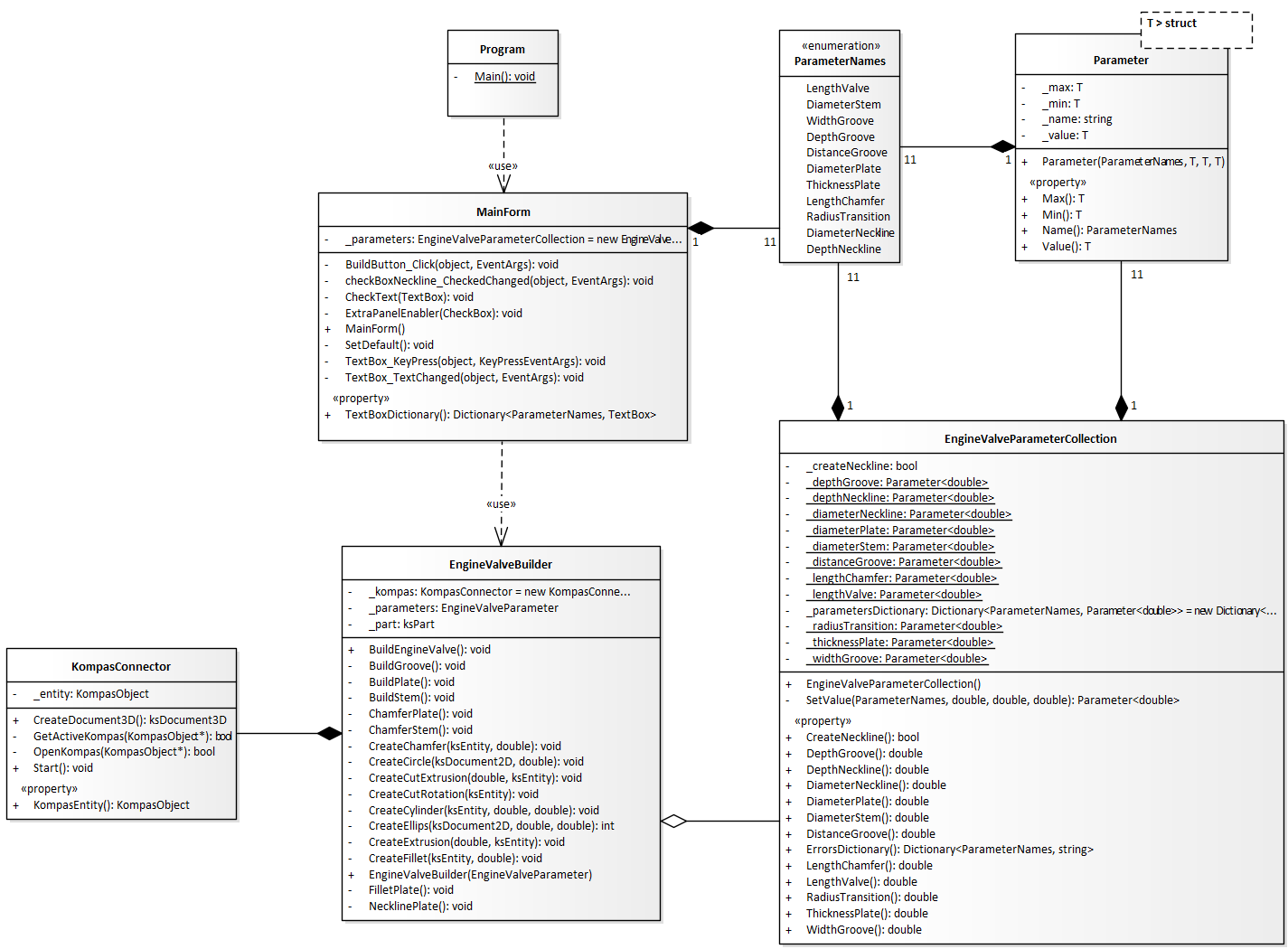


Рисунок 3.2 – Диаграмма классов с дополнительной функциональностью.

Опишем некоторые изменения, возникшие после добавления дополнительной функциональности и переработки кода в процессе разработки плагина.

Был создан шаблонный класс параметров Parameter для организации проверки значений.

Поля класса класса EngineValveParameter стали статическими и имеют тип данных Parameter<double>. Добавлен словари для индексации значений и для хранений ошибок ввода параметров. Добавлены параметры для реализации дополнительной функциональности

Объект класса KompasConnector хранится не в MainForm, а в EngineValveBuilder, для возможности доступа к некоторым методам API.

В класс EngineValveBuilder добавлены методы для реализации дополнительной функциональности.

## 3.2 Макет пользовательского интерфейса

Плагин представляет собой пользовательскую форму с полями для ввода 9 геометрических параметров: «Длина клапана», «Диаметр ножки клапана», «Ширина паза под сухарь», «Глубина паза под сухарь», «Расстояние до паза под сухарь», «Диаметр тарелки клапана», «Толщина тарелки клапана», «Длина рабочей фаски», «Радиус плавного перехода». Ниже располагается кнопка для запуска построения. Справа находится изображение, демонстрирующее размеры на модели. Под изображением расположен переключатель, активирующий дополнительную функциональность.

Макет пользовательского интерфейса представлен на рисунке 3.3.

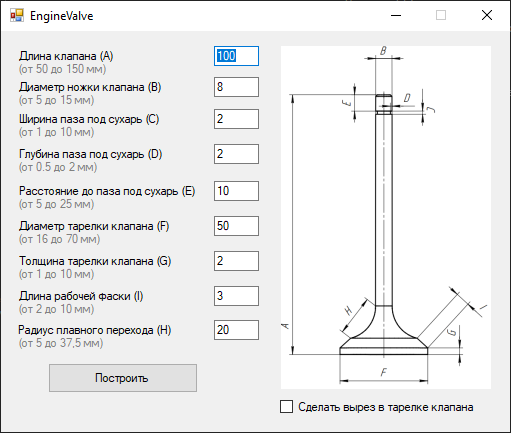


Рисунок 3.3 – Макет пользовательского интерфейса.

При попытке построить модель с неверными параметрами программа выдаст соответствующее сообщение. Пример сообщения представлен на рисунке 3.4.

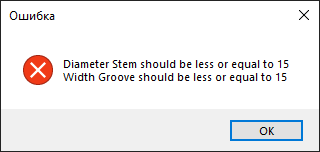


Рисунок 3.4 – Пример сообщения о некорректности данных.

При активации переключателя дополнительной функциональности окно программы меняет свой вид: появляются дополнительные поля для ввода необходимых параметров, кнопка «Построить» меняет своё местоположение. Макет пользовательского интерфейса после активации дополнительной функциональности представлен на рисунке 3.5.

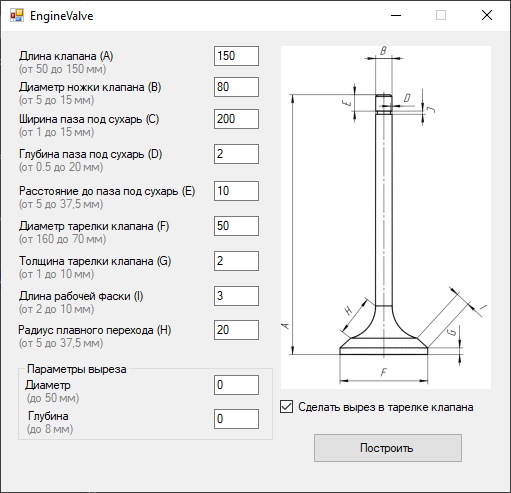


Рисунок 3.5 – Макет пользовательского интерфейса после активации дополнительной функциональности.

# 4. Тестирование плагина

## 4.1 Функциональное тестирование

При функциональном тестировании проверялась корректность работы плагина «Клапан ДВС», а именно, соответствие полученной трёхмерной модели с входными параметрами.

Проведено тестирование с максимальными, минимальными и данными по умолчанию [5].

На рисунке 4.1 представлена модель со стандартными параметрами.

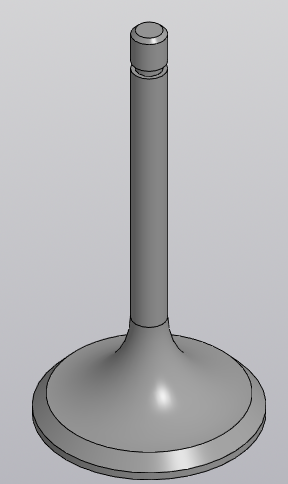


Рисунок 4.1 – Модель клапана с параметрами по умолчанию, без выреза в тарелке.

На рисунке 4.2 представлено сечение модели со стандартными параметрами с вырезом в тарелке клапана

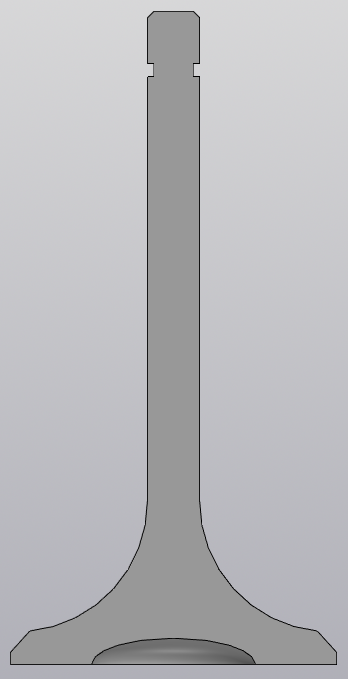


Рисунок 4.2 - Сечение модели со стандартными параметрами и вырезом в тарелке клапана.

На рисунке 4.3 представлена модель с максимальными параметрами.

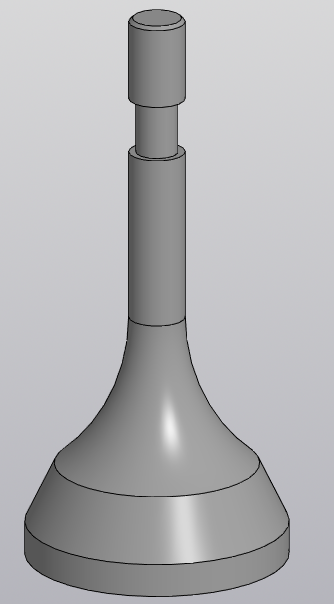


Рисунок 4.3 – Модель с максимальными параметрами.

На рисунке 4.4 представлено сечение модели с максимальными параметрами и вырезом в тарелке клапана.

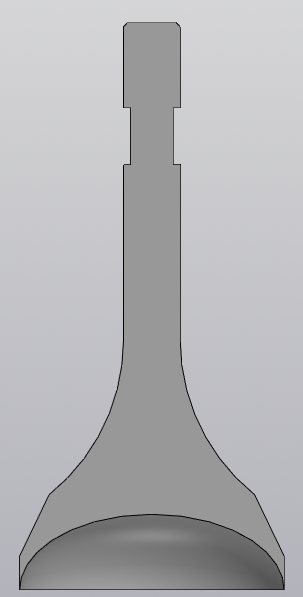


Рисунок 4.4 - Модель с максимальными параметрами и вырезом в тарелке.

На рисунке 4.5 представлена модель с минимальными параметрами.

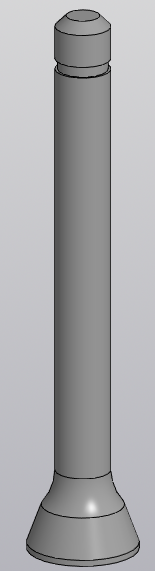


Рисунок 4.5 – Модель с минимальными параметрами.

На рисунке 4.6 представлено сечение модели с минимальными параметрами и вырезом в тарелке клапана.

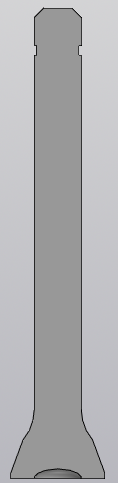


Рисунок 4.6 - Модель с минимальными параметрами и вырезом в тарелке.

## 4.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit [6] версии 3.13 проведено модульное тестирование, проверялись открытые поля, свойства и методы. Были протестированы классы EngineValveParameter и Parameter.

На рисунке 4.7 представлена информация о модульном тестировании программы.

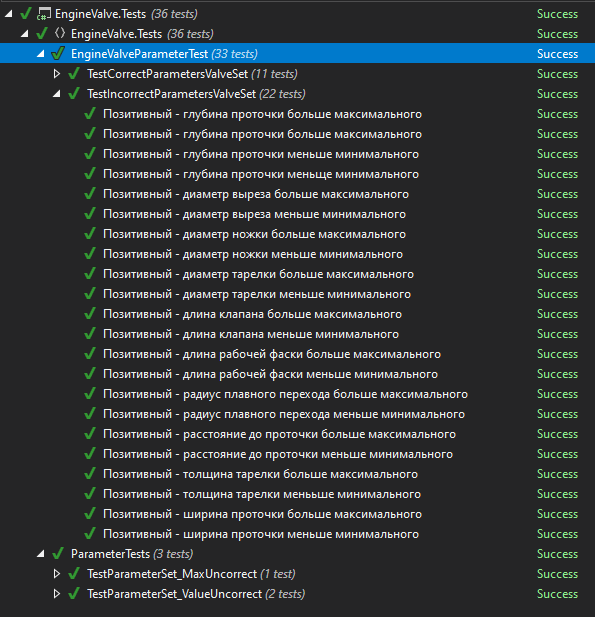


Рисунок 4.7 – Модульное тестирование плагина.

## 4.3 Нагрузочное тестирование

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

* ЦП AMD Ryzen 3 4300U with Radeon Graphics 2.70 ГГц;
* 8,00 ГБ (доступно: 6,87 ГБ) ОЗУ;
* Интегрированное графическое ядро AMD Radeon Vega 5 с объёмом выделенной памяти 1024 Мб.

Для проведения нагрузочного тестирования был добавлен секундомер, который засекал время от начала построения, с каждым успешным построением модели производилась запись результатов в текстовый файл «log.txt». Всего было построено шестьдесят восемь деталей, после чего программа «КОМПАС-3D» аварийно завершила свою работу.

На рисунке 4.8 представлен график зависимости времени построения детали от количества построенных деталей

Рисунок 4.8 – График зависимости времени построения от количества построенных деталей.

Долгое время построения первой детали можно объяснить затратами, необходимыми для открытия САПР. Время, необходимое для построения следующих деталей варьируется от 1 до 5 секунд.

На рисунке 4.9 представлен график зависимости загруженности оперативной памяти от количества построенных деталей.

Рисунок 4.9 – График зависимости загруженности памяти от количества построенных деталей.

Из графика, представленном на рисунке 4.9 можно сделать вывод, что использование оперативной памяти линейно увеличивается до окончания свободного места.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы архитектура и макет системы, создан плагин «Клапан ДВС», проведены модульные, функциональные и нагрузочные тесты.

# Список литературы

1. Система трёхмерного моделирования КОМПАС-3D [электронный ресурс]. – URL: https://ascon.ru/products/7/review/ (дата обращения 26.10.2021).
2. Базовые интерфейсы API системы КОМПАС [электронный ресурс]. – URL: https://it.wikireading.ru/23741 (дата обращения 26.10.2021).
3. MechaniCS [электронный ресурс]. – URL: https://axoft.ru/vendors/CSoft-Development/CSoft-MechaniCS/ (дата обращения 27.10.2021).
4. Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. – СПб.: Символ-Плюс, 2019. – 192 с. (дата обращения 09.11.2021).
5. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 10.12.2021).
6. NUnit [Электронный ресурс]. – URL: https://nunit.org/ (дата обращения: 10.12.2021).