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

«ОРСАПР»

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_ Федяев Н.В.

Проверил:

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_А.А.Калентьев

Томск 2023 г.

**Реферат**

Пояснительная записка 26 страниц, 11 рисунков, 10 таблиц, 10 источников

ТОПОРИЩЕ, ТЕСТИРОВАНИЕ, САПР, ДИАГРАММА КЛАССОВ, ПЛАГИК, КОМПАС 3D, C#.

Объектом является разработка плагина «Топорище» для САПР Компас-3D v.21.

Цель работы – разработать плагин для «Топорища» для САПР Компас-3D v.2.1.

В работе проведено изучение Компас-3D API и разработка плагина «Топорище».

Пояснительная записка содержит:

1. Титульный лист;
2. Реферат;
3. Техническое задание;
4. Введение;
5. Основная часть;
6. Заключение;
7. Список используемых источников.

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)

ЗАДАНИЕ

на индивидуальное задание

по дисциплине «Основы разработки САПР»

Выдано: студенту группы 589-3 Федяеву Николаю Викторовичу

1. Тема: разработка плагина “Топор” для САПР Компас-3D 21.0.2
2. Срок сдачи студентом готовой работы: «31» декабря 2022г.
3. Исходные данные к работе:

- изображения модели топора с размерами рисунок 1

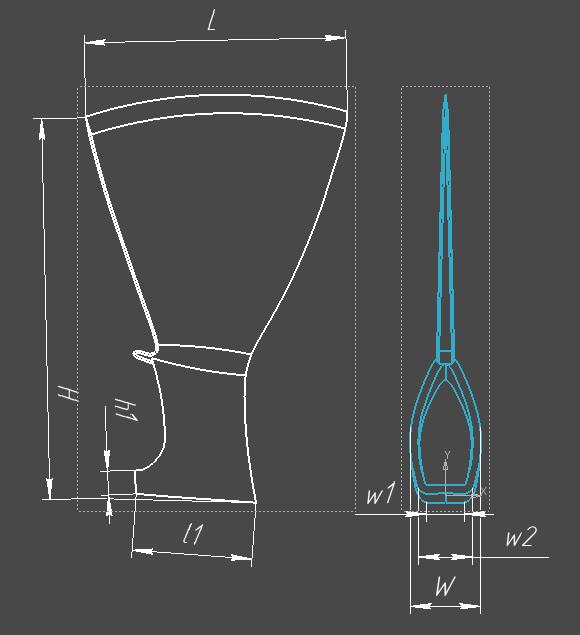


Рисунок 1 – Чертеж топорища в двух проекциях

* изменяемые параметры для плагина:

1. Длина топора **L** (минимум – 135мм, максимум – 165мм);
2. Высота топора **H** (минимум – 170мм, максимум – 215мм);
3. Ширина топора **W** (минимум – 35мм, максимум – 41мм);
4. Высота части топора **h1** (не меньше 1/12 **H**,не больше 1/15 **H**, минимум – 12мм, максимум – 15мм);
5. Длина части топора **l1** (Зависит от длины топора **L**: **L/**2.5 = **l1** минимум – 48мм, максимум – 68мм);
6. Ширина топора **w1** (не больше ширины топора **W** и не меньше ширины топора **w2**, **W > w1 > w2**,минимум – 22мм, максимум – 28мм);
7. Ширина топора **w2** (не больше ширины топора **w1**, **w1 > w2**, минимум – 19мм, максимум – 25мм).

Будет разработана программа с пользовательским интерфейсом с возможностью изменения значений, представленных выше, и последующим построении объекта в САПР Компас-3D. В плагине будут проходить проверки значений, вводимых пользователем. При введении некорректных значений будет изменятся цвет элемента управления. При нажатии на кнопку «Построить» будет проходить проверка правильности ввода данных. Если данные некорректные, то высветится окно с ошибкой построения и не будут применяться введенные параметры.

1. Требование к программному обеспечению:

- Microsoft Windows 10 (64-разрядная версия);

- язык программирования C# с использованием платформы .NET;

- среда разработки Visual Studio 2022;

- плагин для программы Компас-3D 21.0.2;

- библиотека для тестирования NUnit 3.13.3;

- система контроля версии Git.

5. Требование к аппаратному обеспечению:

- ЦП Intel Core i5-11400 2.6ГГц;

- 16 ГБ ОЗУ;

- место на диске — 40 ГБ;

- графический процессор объемом памяти 6 ГБ.

6. Пояснительная записка к индивидуальному заданию должна включать в себя следующие разделы:

* титульный лист;
* реферат;
* техническое задание;
* введение;
* основная часть;
* заключение;
* список используемых источников;
* чертёж детали.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания: «23» сентября 2022 г.

|  |  |
| --- | --- |
| Задание принял к исполнению:  Студент гр. 589-3  Федяев Н.В. \_\_\_\_\_\_\_\_\_\_\_\_ | Руководитель:  к.т.н., доцент каф. КСУП  Калентьев А.А.\_\_\_\_\_\_\_\_\_\_\_\_ |

**Оглавление**

1 Введение…………………………………………………………………………7

2 Постановка и анализ задачи………………………………………………..…..8

2.1 Описание предмета проектирования…………………….…………….…….9

2.2 Выбор инструментов и средств реализации……………………………….11

2.3 Назначение плагина……………………………………………………….…11

3 Обзор аналогов……………………………………………………….…….…..12

4 Описание реализации………………………………………………….…..…..13

4.1 Диаграмма классов…………………………………………………………..13

5 Описание программы для пользователя……………………………………...17

6 Тестирование программы……………………………………………………..19

6.1 Модульное тестирование……………………………………………………20

6.2 Нагрузочное тестирование…………………………………………………..22

Заключение……………………………………………………………………….25

Список использованных источников…………………………………………...26

**1 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].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Топор» для системы автоматизированного проектирования Компас 3D с помощью интегрированной среды разработки Visual Studio 2019 Сommunity [2].

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.

**2 Постановка и анализ задачи**

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой САПР Kompas 3D, строит модель топора [3]. Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры кружки, такие как:

* Длина топора **L**
* Высота топора **H**
* Ширина топора **W**
* Высота части топора **h1**
* Длина части топора **l1**
* Ширина первой части топора **w1**
* Ширина второй части топора **w2**

**2.1 Описание предмета проектирования**

Топорище [5] – часть топора, благодаря которому можно рубить дрова. Существуют различные виды топорища. Изображение моделируемого объекта представлено на рисунке 2.1.

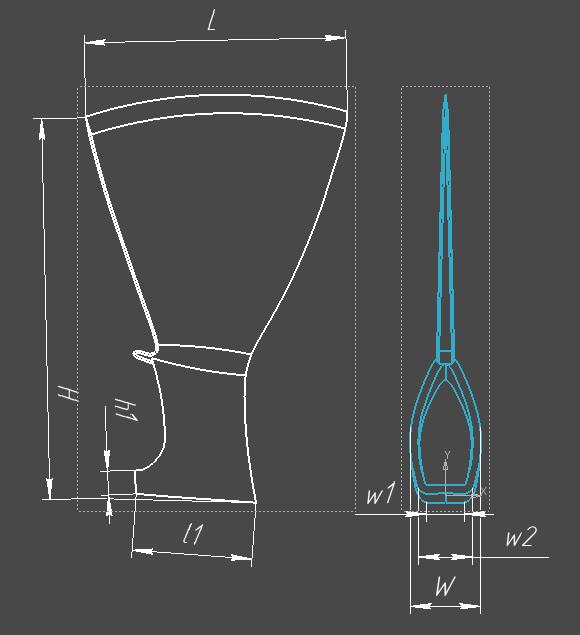


Рисунок 2.1 Чертеж топора

Параметры топора:

* L – длина острой части топора (минимум – 135мм, максимум – 165мм);
* l1 – длина части топора l1 (Зависит от длины топора L: L/2.5 = l1 минимум – 48мм, максимум – 68мм);
* H – высота топора (минимум – 170мм, максимум – 215мм);
* h1 – высота части топора (не меньше 1/12 H, не больше 1/15 H, минимум – 12мм, максимум – 15мм);
* W – ширина топора (минимум – 35мм, максимум – 41мм);
* w1 – ширина кусочка топора (не больше ширины топора W и не меньше ширины топора w2, W > w1 > w2, минимум – 22мм, максимум – 28мм);
* w2 – ширина второго кусочка топора (не больше ширины топора w1, w1 **>** w2, минимум – 19мм, максимум – 25мм).

**2.2 Выбор инструментов и средств реализации**

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2022 с использованием .NET Framework 4.7.2 [2], библиотеки для Kompas 3D [3]. Для реализации пользовательского интерфейса использовалась система для построения настольных приложений WindowsForms.

**2.3 Назначение плагина**

Назначение разрабатываемого плагина обусловлено быстрым моделированием топоров различных размеров. Благодаря данному расширению, любой желающий может наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

**3 Обзор аналогов**

PTC Creo — это масштабируемый, функционально совместимый пакет программного обеспечения для конструирования изделий. Он позволяет группам конструкторов создавать, анализировать, просматривать и максимально использовать проекты изделий при дальнейшем конструировании, используя 2 - и 3-мерное моделирование CAD, параметрическое и прямое моделирование. [4] Классический интерфейс PTC Creo представлен на рисунке 1.1.

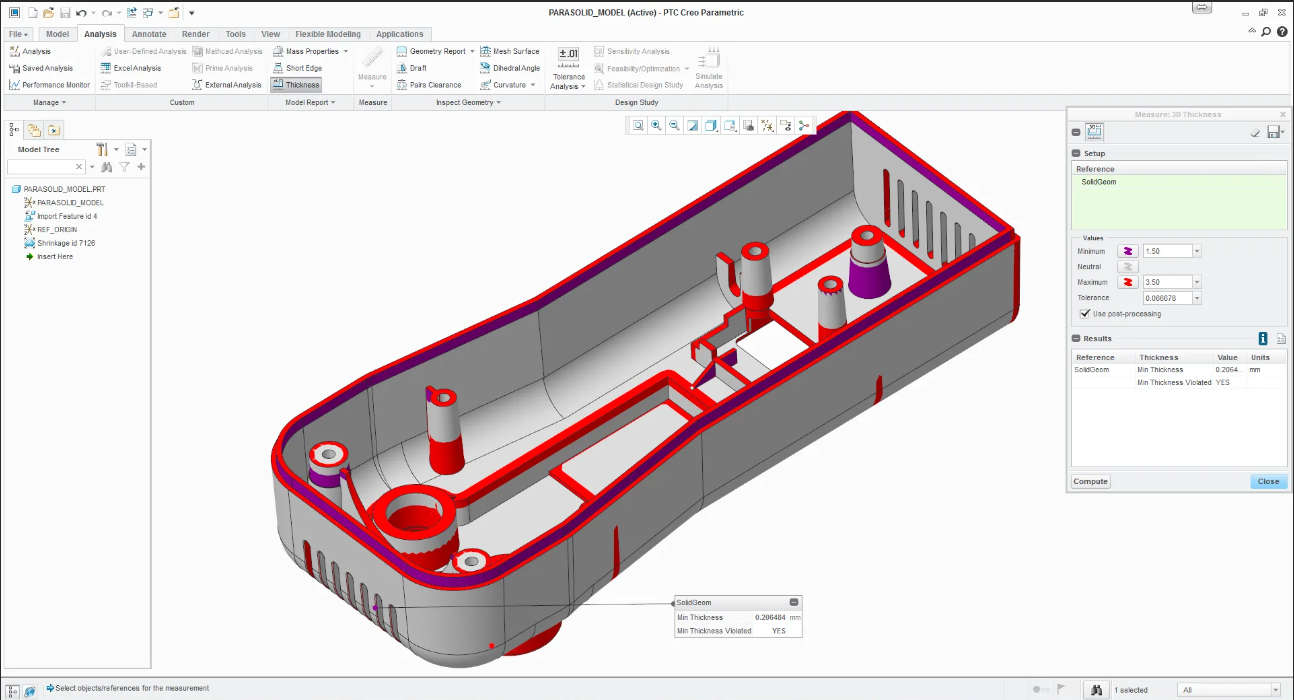


Рисунок 3.1 – Классический интерфейс PTC Creo

**4 Описание реализации**

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 простроена диаграмма классов.

**4.1 Диаграмма классов**

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. Диаграмма классов UML представлена на рисунке 4.1

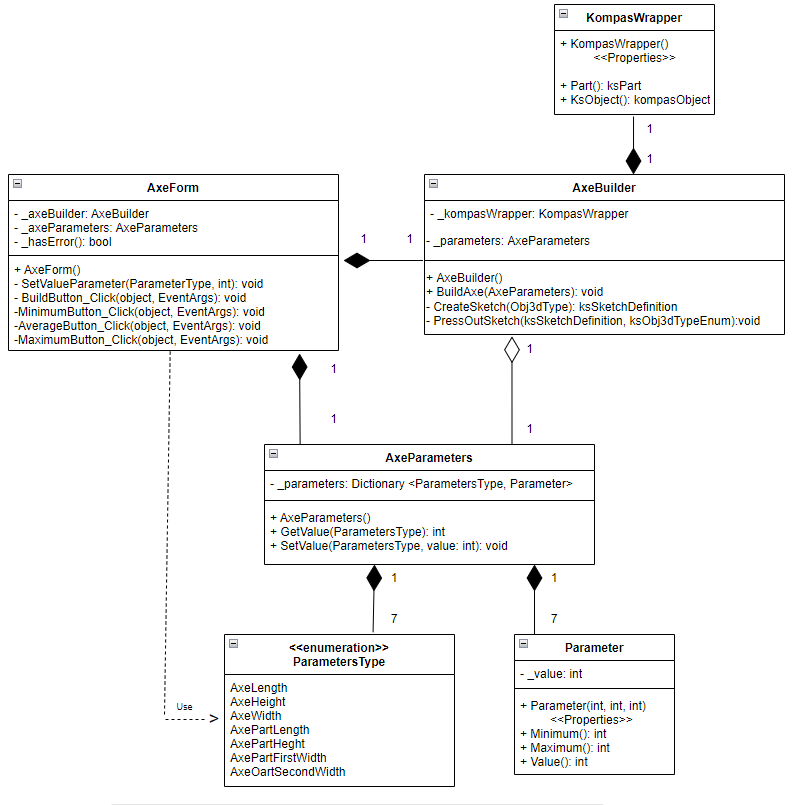


Рисунок 4.1 – Диаграмма классов

Далее в таблицах 4.1 – 4.4 представлено описание классов.

Таблица 4.1 – Описание полей, методов, сущностей класса «AxeForm»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_axeBuilder: AxeBuilder |  | Хранит в себе набор методов для построения топорища |
| - \_axeParameters:  AxeParameters |  | Устанавливает значение параметра |
| -\_hasError() | bool | Выдает ошибку |
| -SetValueParameter  (ParametersType, int) | Void | Устанавливает значение параметра |
| -BuildButton\_Click() | Void | Построение топорища по заданным параметрам |
| -MinimumButton\_Click() | Void | Устанавливает минимальное значение всех параметров |
| -AverageButton\_Click() | Void | Устанавливает среднее значение всех параметров |
| -MaximumButton\_Click() | Void | Устанавливает максимальное значение всех параметров |

Таблица 4.2 - Описание полей, методов, сущностей класса “AxeParameters”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_parameters: Dictionary <ParametersType, Parameter> |  | Хранит данные о каждом параметре модели из перечисления “AxeParameter” |
| + AxeParameters() |  | Конструктор для создания экземпляра класса |
| + GetValue(ParametersType) | Int | Устанавливает значение определённого параметра |
| + SetValue(ParametersType, value: int) | void | Возвращает значение определённого параметра |

Таблица 4.3 - Описание полей, методов, сущностей класса “Parameter”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_value: int |  | Поле, хранящее текущее значение |
| + Parameter(int, int, int) |  | Конструктор для создания экземпляра класса |
| + Minimum(): | Int | Возвращает минимально допустимое значения параметра |
| + Maximum(): | Int | Возвращает максимально допустимое значения параметра |
| + Value(): | Int | Возвращает и задаёт значение параметра |

Таблица 4.4 - Описание полей, методов, сущностей класса “AxeBuilder”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_kompasWrapper: KompasWrapper |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| - \_parameters: AxeParameters |  | Хранит данные о каждом параметре модели из перечисления “AxeParameter” |
| + BuilderAxe(AxeParameters) | Void | Построение топора по заданным параметрам |
| - CreateSketch(obj3dType) | ksSketchDefinition | Возвращает интерфейс параметров эскиза |
| -PressOutSketch (ksSketchDefinition, int) | Void | Выдавливает эскиз |

**5 Описание программы для пользователя**

Макет пользовательского интерфейса представляет собой форму для ввода параметров топора. Построение модели осуществляется путем нажатия на кнопку «Build».

На рисунке 5.1 представлен макет пользовательского интерфейса.

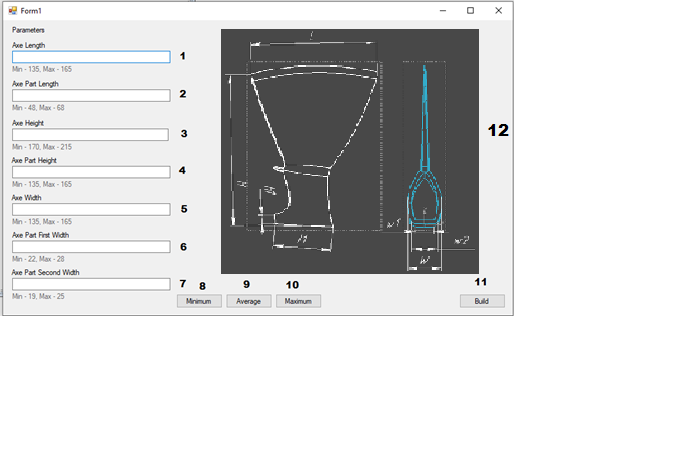


Рисунок 5.1 – Макет пользовательского интерфейса

Описание пользовательского интерфейса:

1. Поле ввода параметра длины топора;
2. Поле ввода параметра части длины топора;
3. Поле ввода параметра высота топора;
4. Поле ввода параметра части высота топора;
5. Поле ввода параметра ширины топора;
6. Поле ввода параметра первой части ширины топора;
7. Поле ввода параметра второй части ширины топора;
8. Кнопка для установления минимального значения топора;
9. Кнопка для установления среднего значения топора;
10. Кнопка для установления максимального значения топора;
11. Кнопка для построения 3D модели топора;
12. Область для вывода чертежа.

После ввода некорректных значений, окно для ввода данных станет красным. Пример представлен на рисунке 5.2.

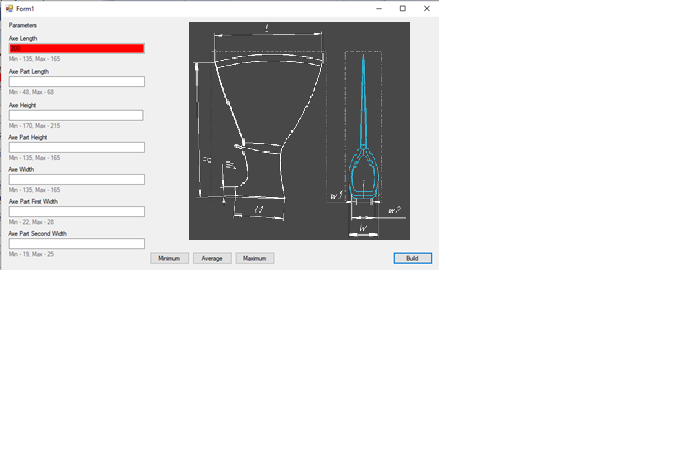


Рисунок 5.2 – Макет пользовательского интерфейса с некорректными данными

После ввода необходимых параметров, построить деталь в САПР Компас 3D можно с помощью кнопки «Build». Топор, построенный по заданным параметрам в САПР Kompas 3D, представлена на рисунке 5.4.

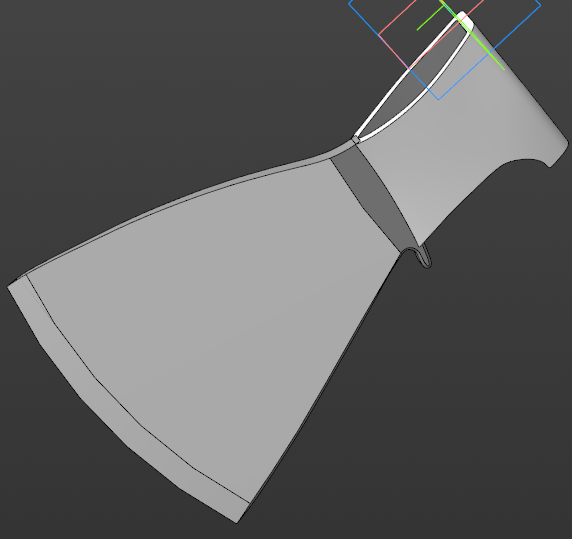


Рисунок 5.3 – Топор, построенный по заданным параметрам в САПР Компас 3D

**6 Тестирование программы**

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

**6.2 Модульное тестирование**

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование [7,8], проверялись открытые поля и методы. На рисунке 6.3 представлено тестирование библиотеки классов: Model.

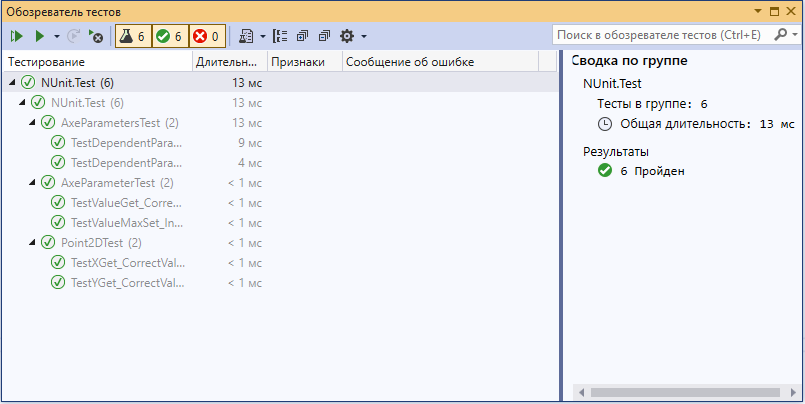


Рисунок 6.3 – Тестирование классов

Перечень тестов с их описанием представлен в таблице 6.1.

Таблица 6.1 – тестовые случаи метода TestDependentParameterSet\_CorrectValues

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| Diameter | 21 | Проверка корректного присвоения значения свойства |
| Length | 45 | Проверка корректного присвоения значения свойства |
| RadiusInnerRings | 30 | Проверка корректного присвоения значения свойства |

Таблица 6.2 – тестовые случаи метода TestDependentParameterSet\_IncorrectValues.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| Diameter | 60 | Проверка некорректного присвоения значения свойства |
| Length | 30 | Проверка некорректного присвоения значения свойства |
| RadiusInnerRings | 20 | Проверка некорректного присвоения значения свойства |

Таблица 6.3 – тестовые случаи метода TestValueGet\_CorrectValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| Thickness | 20 | Проверка корректного получения значения свойства |

Таблица 6.4 – тестовые случаи метода TestMaxSet\_IncorrectValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| Thickness | 40 | Проверка некорректного присвоения значения свойства |

Таблица 6.5 – тестовые случаи метода TestXGet\_CorrectValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| X | 10 | Проверка корректного присвоения значения свойства |

Таблица 6.6 – тестовые случаи метода TestYGet\_CorrectValue.

|  |  |  |
| --- | --- | --- |
| Тестируемое свойство | Параметры | Описание |
| Y | 10 | Проверка корректного присвоения значения свойства |

**6.2 Нагрузочное тестирование**

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [10]. Тестирование производилось на ПК со следующей конфигурацией:

- ЦП Intel Core i5-11400 2.6ГГц;

- 16 ГБ ОЗУ;

- место на диске — 40 ГБ;

- графический процессор объемом памяти 6 ГБ.

На рисунке 6.2 для проведения нагрузочного, с каждым успешным построением фигуры производилась запись результатов в текстовый файл «StressTest.txt».



Рисунок 6.2 – Зацикливание перестроения фигуры

На графике, изображенном на рисунке 6.3 ось «Х» - количество построенных деталей, ось «Y» - количество потребляемой оперативной памяти. На графике, изображенном на рисунке 6.4 в текущей главе, ось «X» – время в минутах, ось «Y» – количество построенных деталей..

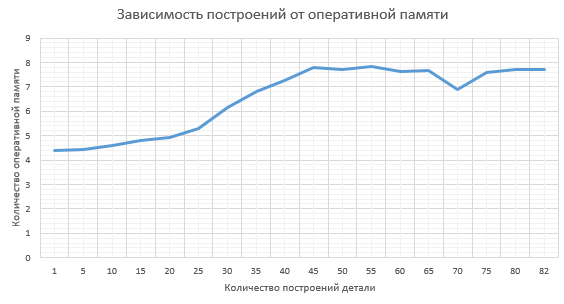


Рисунок 6.3 – График зависимости памяти от количества деталей

Из графика видно, что зависимость является линейной пока оперативная память на загружена практически полностью. На графике имеются падения, то есть происходит уменьшение нагрузки на оперативную память. Это обусловлено устройством оперативной памяти, что для её регенерации периодически приостанавливается обращение, это снижает среднюю скорость обмена и понижает нагрузку [12].

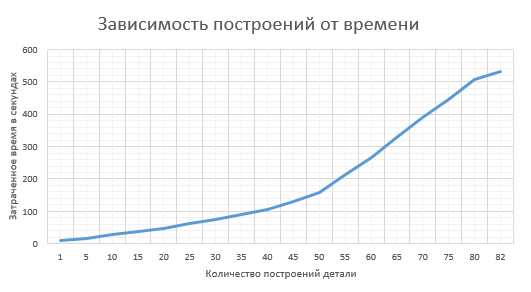


Рисунок 6.4 – График зависимости времени от количества построенных деталей.

Из графика видно, что зависимость количества необходимого времени от числа деталей является линейной с изменением угла наклона в течение теста. Это связано с тем, что при большой загрузке оперативной памяти и центрального процессора падает скорость построения одной детали. Поэтому увеличивается коэффициент наклона прямой.

**Заключение**

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API, функциональное и нагрузочное тестирование и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D моделей «Топор» в САПР Kompas 3D, и проведено функциональное и нагрузочное тестирование плагина.

**Список использованных источников**

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 3.05.2023).
2. Visual Studio [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 3.05.2023).
3. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 3.05.2023).
4. PTC Creo [Электронный ресурс]. — Режим доступа: <https://junior3d.ru/article/creo.html> (дата обращения 3.05.2023).
5. Топор – Википедия. [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BF%D0%BE%D1%80> (дата обращения 3.05.2023).
6. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 3.05.2023).
7. NUnit [Электронный ресурс]. – URL: https://nunit.org/ (дата обращения: 3.05.2023).
8. Юнит-тестирование для чайников [Электронный ресурс]. – URL: https://habr.com/ru/post/169381/ (дата обращения: 3.05.2023).
9. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 3.05.2022).
10. Нагрузочное тестирование [Электронный ресурс]. – URL: https://habr.com/ru/company/jugru/blog/329174/ (дата обращения: 03.05.2023).