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»

|  |  |
| --- | --- |
|  | Выполнил:  Студент гр. 580-1  \_\_\_\_\_\_\_\_\_\_\_ Ершов И.А.  « » 2023 г. |
| \_\_\_\_\_\_\_  (оценка) | Проверил  к.т.н., доцент каф. КСУП  \_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.  « » 2023 г. |

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc153884422)

[1 ВВЕДЕНИЕ 3](#_Toc153884423)

[2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_Toc153884424)

[2.1 Описание предмета проектирования 4](#_Toc153884425)

[2.2 Выбор инструментов и средств реализации 5](#_Toc153884426)

[2.3 Назначение плагина 6](#_Toc153884427)

[3 ОБЗОР АНАЛОГОВ ПЛАГИНА 7](#_Toc153884428)

[4 ОПИСАНИЕ РЕАЛИЗАЦИИ 8](#_Toc153884429)

[5 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 12](#_Toc153884430)

[6 ТЕСТИРОВАНИЕ ПЛАГИНА 15](#_Toc153884431)

[6.1 Функциональное тестирование 15](#_Toc153884432)

[6.2 Модульное тестирование 16](#_Toc153884433)

[6.3 Нагрузочное тестирование 17](#_Toc153884434)

[7 ЗАКЛЮЧЕНИЕ 20](#_Toc153884435)

[8 СПИСОК ИСТОЧНИКОВ 21](#_Toc153884436)

# 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, оценить технологические и экономические характеристики производства, 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 «Киянка» для системы автоматизированного проектирования КОМПАС-3D с помощью интегрированной среды разработки Visual Studio 2022 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, на языке программирования C#. Плагин на основе входных параметров, интегрируясь с системой «КОМПАС-3D», строит модель «Киянка» [3].

Сроки реализации данного проекта:

* выбор темы + git-репозиторий (18.09.23 - 24.09.23);
* составление технического задания (25.09.23 - 08.10.23);
* составление проекта системы (09.10.23 - 22.10.23);
* прототип плагина (20.11.23 - 03.12.23);
* готовый плагин (04.12.23 - 31.12.23);

Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры забора, такие как:

* форма бойка;
* Радиус скругления
* диаметр бойка;
* ширину бойка;
* высоту бойка;
* длину бойка;
* высоту рукоятки;
* диаметр рукоятки.

В ходе анализа реализации плагина были выявлены проблемы в сложно читаемой документации API для САПР «КОМПАС-3D». Из положительных сторон можно выделить, что в открытом доступе есть множество различных примеров кода по правильному использованию API.

## 2.1 Описание предмета проектирования

Кия́нка – столярный молоток из дерева твёрдых пород. Внешним видом киянка похожа на молоток, только с увеличенным бойком. Еще одним отличием является то, что она сделана из дерева или резины, а не из металла. Боёк киянки может быть выполнен в форме прямоугольника или цилиндра. Белый резиновый боёк, в отличие от чёрного, не оставляет следов после своих ударов [4].

На рисунке 2.1 представлен чертёж киянки.

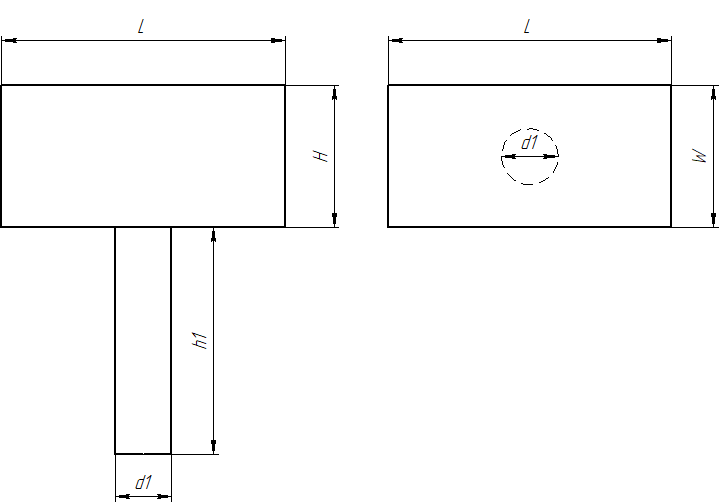


Рисунок 2.1 – Чертёж киянки с размерами, вид сбоку и сверху

Параметры киянки:

* форма бойка (прямоугольная или цилиндрическая);
* радиус скругления цилиндрической формы бойка (0 — 10мм);
* диаметр цилиндрической формы бойка (50 — 100мм);
* ширина прямоугольной формы бойка W (50 — 100мм);
* длина бойка L (75 — 200мм);
* высота прямоугольной формы бойка H (50 — 100мм);
* длина рукоятки h1 (100 — 250мм);
* диаметр рукоятки d1 (25 — 75мм);
* длина бойка L не меньше его ширины W;
* диаметр рукоятки D не больше ширины бойка W.

## 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 6.0, библиотеки для Kompas 3D [5].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [6].

Для реализации пользовательского интерфейса был использован WPF [7].

## 2.3 Назначение плагина

Назначение плагина обусловлено быстрым моделированием киянок разных размеров и типов. Благодаря данному расширению, мастера по киянкам могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# 3 ОБЗОР АНАЛОГОВ ПЛАГИНА

Для быстрой разработки графической документации была разработана библиотека «Инструмент». Данная библиотека работает в среде системы Компас-3D. Основные функции библиотеки «Инструмент»:

1. быстрый и наглядный поиск необходимого инструмента в базе данных;
2. быстрое вычерчивание различных инструментов в системе Компас-3D в соответствии с их точными размерами, согласно ГОСТ, ОСТ или ТУ.

Основные геометрические параметры инструментов хранятся в базе данных инструментов и полностью соответствуют ГОСТ. Для того чтобы можно было автоматически начертить инструмент любого типоразмера в графической системе, для каждого его вида разработана своя математическая модель. С помощью математических моделей и происходит расчет всех недостающих размеров инструмента, необходимых для его вычерчивания. Библиотека «Инструмент» может использоваться технологами для быстрого создания операционных эскизов в системе Компас-3D. Она значительно экономит время при создании графических документов, в которых присутствуют различные инструменты. [8]



Рисунок 3.1 – Интерфейс библиотеки

# 4 ОПИСАНИЕ РЕАЛИЗАЦИИ

UML диаграмма классов представляет собой графическую интерпретацию классов системы, их атрибутов, методов и взаимосвязей между ними [9].

На рисунке 4.1 представлена диаграмма классов.

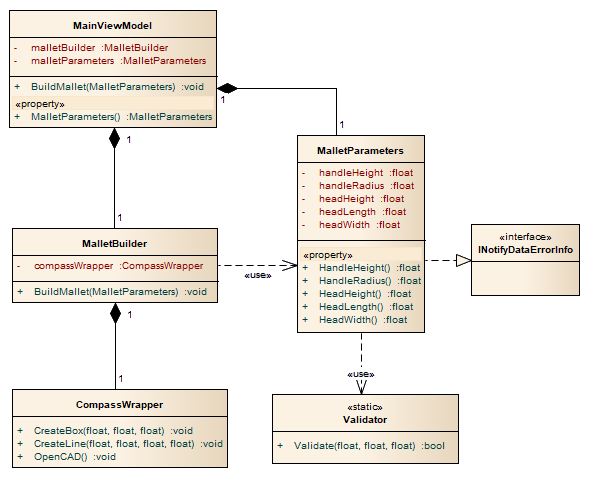


Рисунок 4.1 – Изначальная диаграмма классов

Разберем основные классы проекта:

 –  MainWindow – является главным окном приложения. Хранит в себе ViewModel;

 –  MalletParameters класс, хранящий в себе все параметры модели;

 –  CompassWrapper – класс обертка API САПР. В нем находятся все нужные методы создания примитивов и документов, которые пригодятся для построения модели.

 –  Model хранит часть моделей бизнес-логики: валидаторы, классы, связанные с объектом построения;

 –  ViewModel связывает пользовательский интерфейс приложения с бизнес логикой приложения.

 –  View хранит в себе пользовательский интерфейс плагина;

 –  MalletBuilder класс отвечающий за строительство модели киянки.

В итоговом проекте созданы следующие классы и методы, которые отображены на итоговой диаграмме классов (рисунок 4.2).

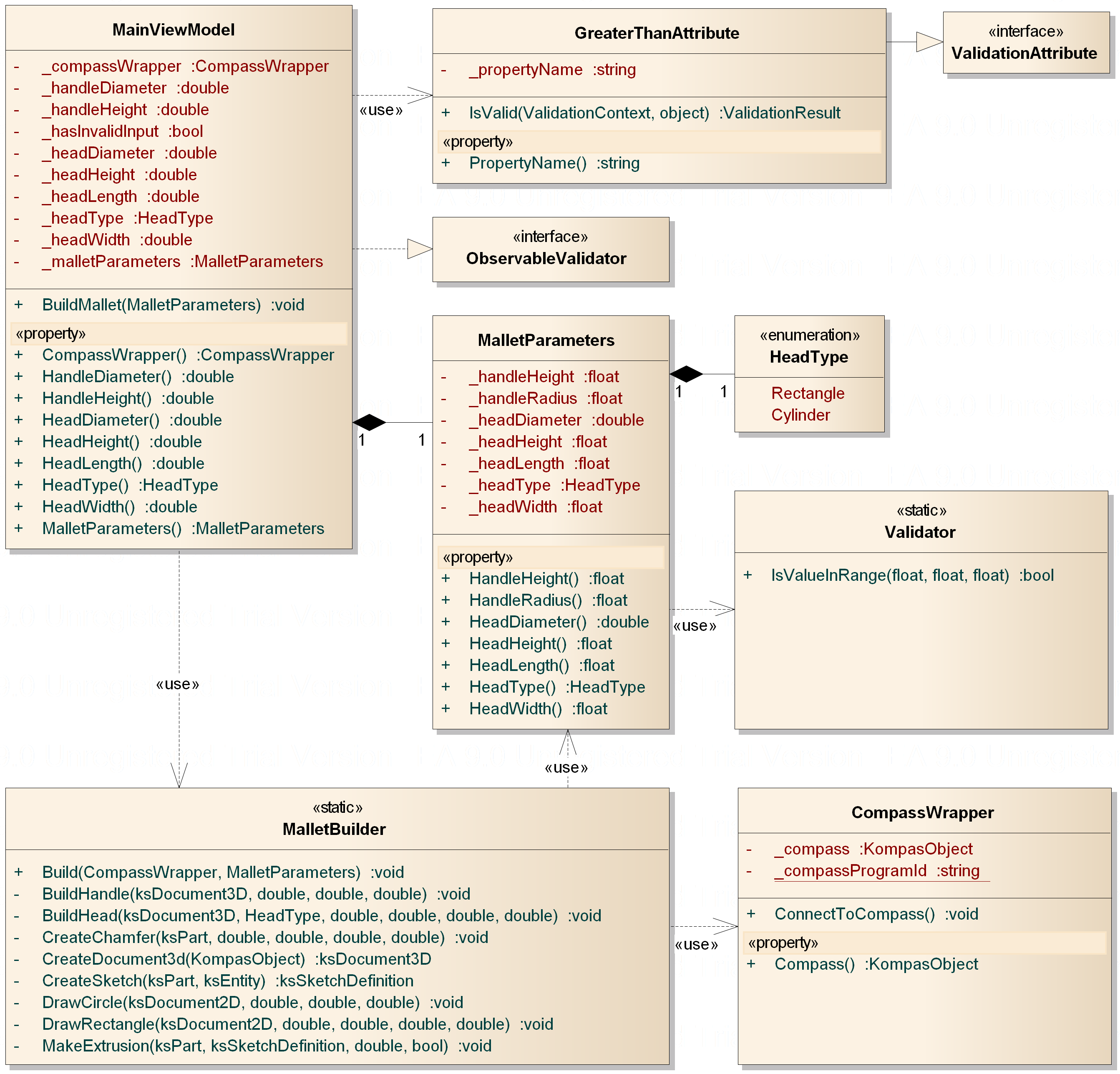


Рисунок 4.2 – Итоговая диаграмма классов

Разберем основные изменения классов проекта после реализации:

 –  Был добавлен интерфейс ValidationAttribute для того, чтобы программа могла валидировать зависимые параметры при построении модели «Киянка».

 –  Также были реализованы класса от этого интерфейса — GreaterThanAttribute.

 –  Интерфейс INotifyDataErrorInfo был заменён на ObservableValidator, для валидации параметров в MainViewModel.

 –  MalletBuilder обзавёлся дополнительными методами для построения киянки.

 –  Связь между MainViewModel и MalletBuilder заменена с композиции на использование.

–  Были также добавлены новые параметры для модели, для построения киянок разных форм.

# 5 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ

Макет пользовательского интерфейса включает в себя окно, в котором пользователь вводит параметры для дальнейшего построения модели «Киянка» в САПР «КОМПАС-3D».

На рисунке 5.1 представлен пользовательский интерфейс плагина.

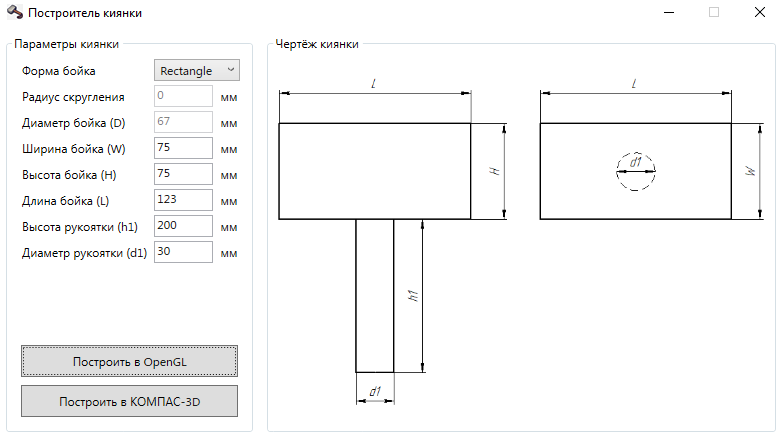


Рисунок 5.1 – Пользовательский интерфейс плагина

Если все данные были введены правильно, то после нажатия кнопки "Построить" модель киянки будет создана.

Результат построений представлен на рисунке 5.2.

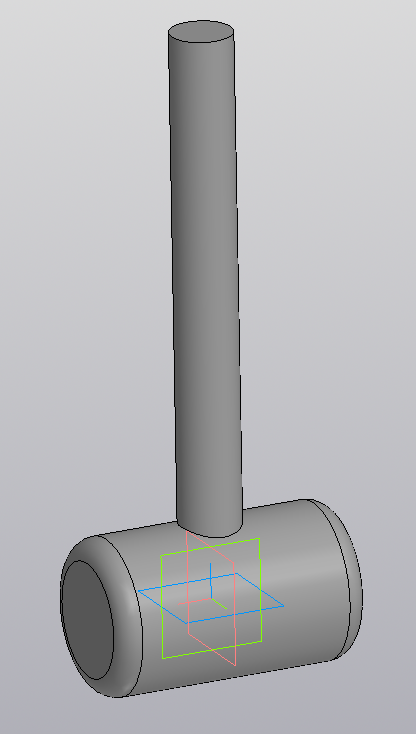
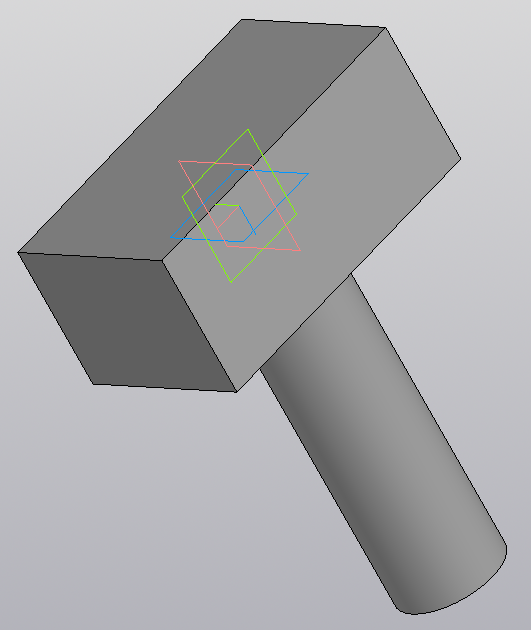


Рисунок 5.2 – Модель киянок в КОМПАС-3D

В случае некорректного ввода параметров модель, кнопка для построения, станет не доступна, и пользователь получит сообщение об ошибке при наведении мышки на неправильно заполненное поле (Рисунок 5.3).

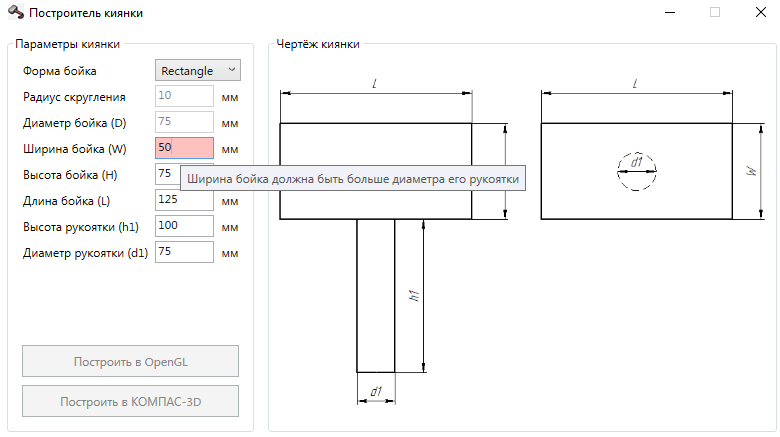


Рисунок 5.3 – Реакция плагина на некорректный ввод

Возможные варианты ошибок и их решений перечислены в таблице 5.1.

Таблица 5.1 – Ошибки и их варианты исправления

|  |  |
| --- | --- |
| **Текст ошибки** | **Варианты исправления** |
| Параметр A должен быть больше параметра B | Задать значение для параметра A больше чем для параметра B |
| Параметр A должен быть задан в следующем диапазоне: [\* - \*] | Задать значение для параметра A входящее в требуемый диапазон |
| Числовое поле параметра A содержит недопустимый символ | Удалить в поле параметра A недопустимый символ |

Также построение киянки было реализовано на C++, с использованием графического API OpenGL v1.1 [10]. Результаты построения представлены на рисунке 5.4.

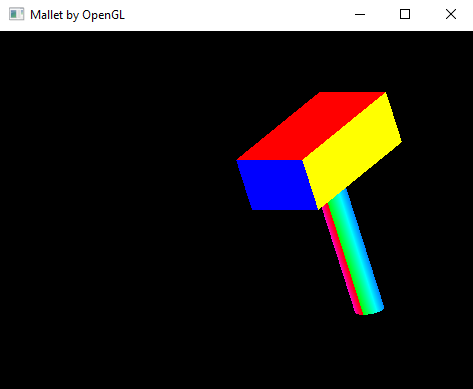
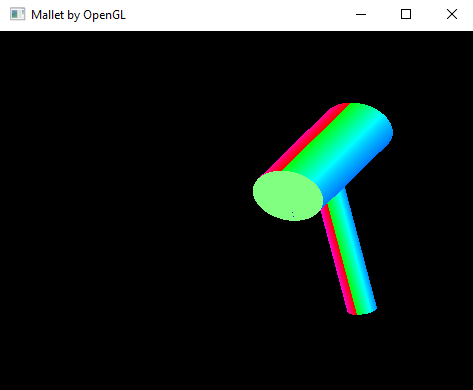
 

Рисунок 5.4 – Модели киянок нарисованные с помощью OpenGL v1.1

# 6 ТЕСТИРОВАНИЕ ПЛАГИНА

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

## 6.1 Функциональное тестирование

При функциональном тестировании проверялась корректность работы плагина «Киянка», а именно соответствие полученного результата в виде трехмерной модели с входными параметрами [11].

Результаты тестирование минимальных (ширина бойка 50 мм, высота бойка 50 мм, длина бойка 50 мм, высота рукоятки 100 мм, диаметр рукоятки 25 мм) и максимальных (ширина бойка 100 мм, высота бойка 100 мм, длина бойка 200 мм, высота рукоятки 250 мм, диаметр рукоятки 75 мм) параметров модели «Киянка» представлена на рисунке 6.1.

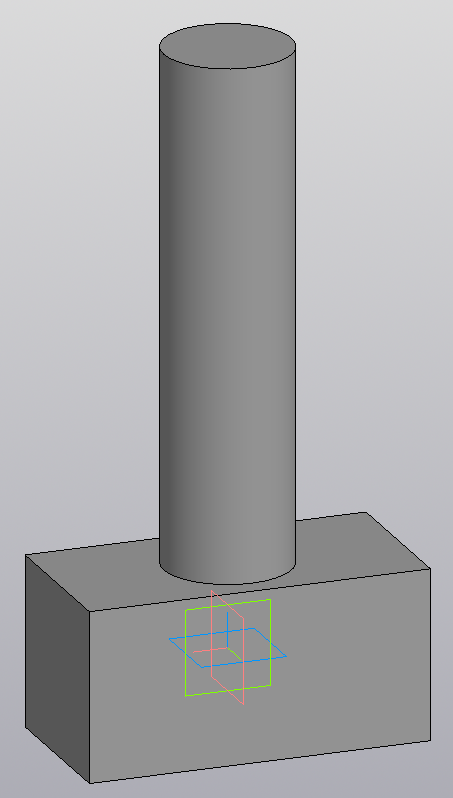
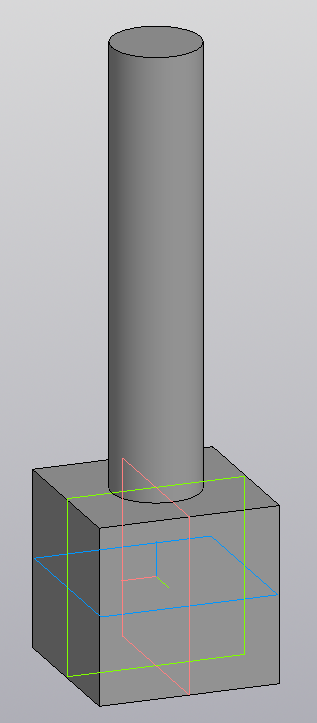


Рисунок 6.1 – Результаты построения минимальных (слева) и максимальных (справа) параметров модели «Киянка»

## 6.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit проведено модульное тестирование [12]. проверялись открытые поля и методы.

На рисунке 6.2 представлены результаты модульного тестирования классов проектов, а именно: Validator, MalletParameters.

Степень покрытия проектов — сто процентов. Было написано 22 теста.

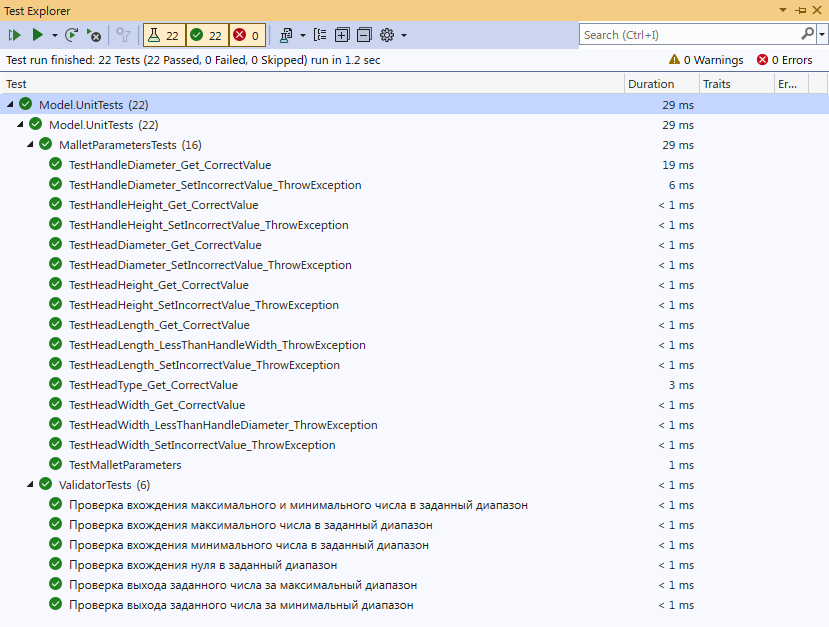


Рисунок 6.2 – Результаты модульного тестирования

## 6.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [15]. Тестирование производилось на ПК со следующей конфигурацией:

* ЦП AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
* 8 ГБ ОЗУ (6,94 GB usable);
* Windows 64 bit

На рисунке 6.3 представлен код для проведения нагрузочного тестирования. В нём использовался секундомер («Stopwatch»), который засекал время от начала построения. После успешного построения фигуры производилась запись результатов в текстовый файлы «log.txt».

|  |
| --- |
| var stopWatch = new Stopwatch();  stopWatch.Start();  var malletParameters = new MalletParameters(  HeadType.Rectangle, 0, 75, 75, 75, 125, 175, 50);  var compassWrapper = new CompassWrapper();  var streamWriter = new StreamWriter("log.txt", true);  var count = 0;  while (true)  {  const double gigabyteInByte = 0.000000000931322574615478515625;  MalletBuilder.Build(compassWrapper, malletParameters);  var computerInfo = new ComputerInfo();  var usedMemory =  (computerInfo.TotalPhysicalMemory - computerInfo.AvailablePhysicalMemory)  \* gigabyteInByte;  streamWriter.WriteLine(  $"{++count}\t{stopWatch.Elapsed:hh\\:mm\\:ss}\t{usedMemory}");  streamWriter.Flush();  } |

Рисунок 6.3 – Код для нагрузочного тестирования

На рисунке 6.4, 6.5 и 6.6 представлено тестирование зацикленного построения модели со следующими параметрами:

* длина бойка: 125 мм;
* ширина бойка: 75 мм;
* высота бойка: 75 мм;
* высота рукоятки: 175 мм;
* диаметр рукоятки: 50 мм;

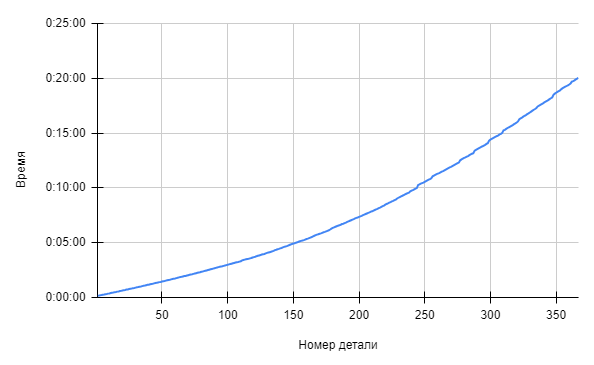


Рисунок 6.4 – График зависимости времени от количества построенных деталей со средними параметрами

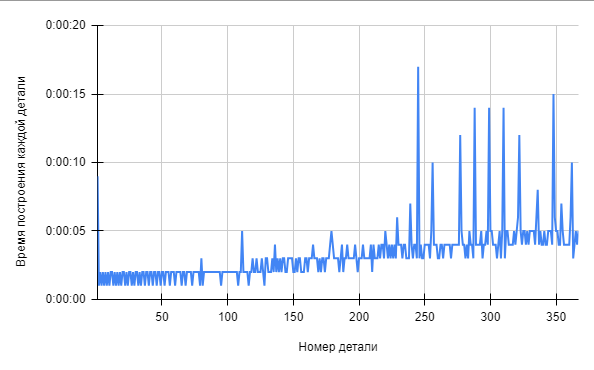


Рисунок 6.5 – График зависимости времени построения одной детали от количества деталей



Рисунок 6.6 – График зависимости загруженности памяти от количества построенных деталей со средними параметрами

Исходя из вышеуказанных графиков на рисунках 6.4, 6.5 и 6.6, можно сделать следующие выводы.

Во-первых, объем оперативной памяти, затрачиваемый плагином на построение трехмерных моделей «Киянка», линейно увеличивается до достижения предела объема оперативной памяти. По окончании свободного места оперативная память частично очищается, после чего операционная система начинает использовать файл подкачки для компенсации недостатка оперативной памяти.

Во-вторых, после построения триста шестидесяти семи деталей работа «КОМПАС-3D», экстренно завершается. Вероятно, это связано с тем, что доступная для работы оперативная память (включая файл подкачки) закончилась.

В-третьих скорость построения в САПР «КОМПАС-3D», увеличивается экспоненциально.

# 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 «Киянка» в САПР «КОМПАС-3D», и проведено функциональное и нагрузочное тестирование плагина.

# Список источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 10.12.2023).
2. Visual Studio [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 10.12.2023).
3. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 22.10.2023).
4. Киянка – Википедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D1%8F%D0%BD%D0%BA%D0%B0 (дата обращения: 18.10.2023).
5. КОМПАС-3D для разработчиков [Электронный ресурс]. – URL: https://kompas.ru/solutions/developers/ (дата обращения: 10.12.2023).
6. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 10.12.2023).
7. Что такое Windows Presentation Foundation (WPF) [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022 (дата обращения: 10.12.2023).
8. Библиотека «Инструмент» для «Компас-3D» [электронный ресурс].– URL: <http://www.insoftmach.ru/Instrument.html> (дата обращения 15.12.2023).
9. UML. Основы / Фаулер, М. – 3-е изд., пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
10. OpenGL [Электронный ресурс]. – URL: https://www.opengl.org/ (дата обращения: 10.12.2023).
11. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 10.12.2023).
12. Юнит-тестирование для чайников [Электронный ресурс]. – URL: <https://habr.com/ru/post/169381/> (дата обращения: 10.12.2023).