Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

РАЗРАБОТКА ПЛАГИНА «Расческа» ДЛЯ «КОМПАС-3D»

Пояснительная записка по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 580-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ковалев В. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Томск 2023

# **Оглавление**

[Оглавление 2](#_30j0zll)

[Введение 3](#_we4wuj8mrofg)

[2 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_1fob9te)

[2.1 Описание предмета проектирования 5](#_3znysh7)

[2.2 Выбор инструментов и средств реализации 6](#_2et92p0)

[2.3 Назначение плагина 6](#_tyjcwt)

[3 ОБЗОР АНАЛОГОВ ПЛАГИНА 7](#_3dy6vkm)

[4 ОПИСАНИЕ РЕАЛИЗАЦИИ 8](#_1t3h5sf)

[5 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 10](#_4d34og8)

[6 ТЕСТИРОВАНИЕ ПЛАГИНА 13](#_2s8eyo1)

[6.1 Функциональное тестирование 13](#_17dp8vu)

[6.2 Модульное тестирование 14](#_3rdcrjn)

[6.3 Нагрузочное тестирование 15](#_26in1rg)

[Заключение 18](#)

[Список источников 19](#_y8xv1ovhfiw3)

# Введение

В современном мире автоматизация процессов является важным аспектом в различных сферах деятельности. Одной из таких сфер является проектирование в программе Kompas 3D. Для упрощения и ускорения процесса построения расчесок, было решено разработать плагин, который позволит автоматизировать этот процесс.

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Расческа» для системы автоматизированного проектирования КОМПАС-3D с помощью интегрированной среды разработки Visual Studio 2022 Community [1].

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 техническим заданием требовалось разработать плагин, на языке программирования C#. Плагин на основе входных параметров, интегрируясь с системой «КОМПАС-3D», строит модель «Расческа» [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 расчески, такие как:

* Длина ручки;
* Количество зубьев;
* Длина зубьев;
* Расстояние между зубьями;
* Ширина зубьев;
* Ширина ручки.

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**

Расчёска — индивидуальный предмет быта для расчесывания волос, массажа головы и нанесения воды на волосы. Также есть расчёски для шерсти животных. Существует два основных типа расчёски — гребень и щётка для волос. [2].

На рисунке 2.1 представлен чертёж расчески.

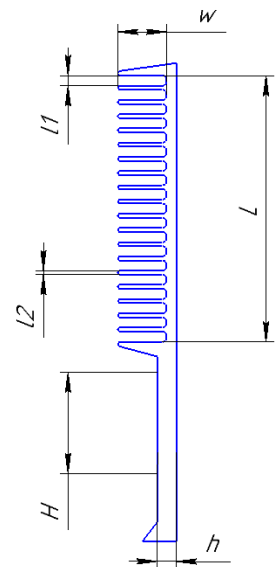


Рисунок 2.1 – Чертёж расчески с размерами, вид сбоку.

Параметры расчески:

* длина ручки расчески H (50 — 200мм);
* ширина ручки расчески h (10 - 30мм)
* общее количество зубьев n (20 - 50)
* длина зубчиков расчески w (20 - 50мм)
* расстояние между зубчиками расчески l1 (1 - 10мм)
* ширина зубчиков расчески l2 (1 - 3мм)
* длина секции с зубчиками рассчитывается из формулы L = n \* (l1 + l2) + l1

## **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 7.0, библиотеки для Kompas 3D [3].

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [4].

Для реализации пользовательского интерфейса был использован WinForms[5].

## **2.3 Назначение плагина**

Назначение разрабатываемого плагина обусловлено быстрым моделированием расчесок разных типов. Благодаря данному расширению, каждую расческу можно создать индивидуально под каждого клиента.

# 3 ОБЗОР АНАЛОГОВ ПЛАГИНА

K3D-Parametric - это библиотека для языка программирования Python, которая предоставляет возможности для создания и визуализации трехмерных геометрических моделей с использованием параметрического подхода.

Основная идея параметрического моделирования заключается в том, чтобы описать объект в виде набора параметров, которые могут быть выразимы в математической форме. Это позволяет легко изменять и контролировать различные характеристики моделей, такие как размеры, форма, текстура и другие свойства.

K3D-Parametric предоставляет набор инструментов и функций для создания и манипулирования параметрическими моделями. Библиотека поддерживает различные типы объектов, такие как примитивы (как кубы, сферы, цилиндры и т. д.), а также более сложные структуры, такие как поверхности Безье, сплайны и другие.

# 4 ОПИСАНИЕ РЕАЛИЗАЦИИ

UML диаграмма классов представляет собой графическую интерпретацию классов системы, их атрибутов, методов и взаимосвязей между ними [6].

На рисунке 4.1 представлена диаграмма классов.

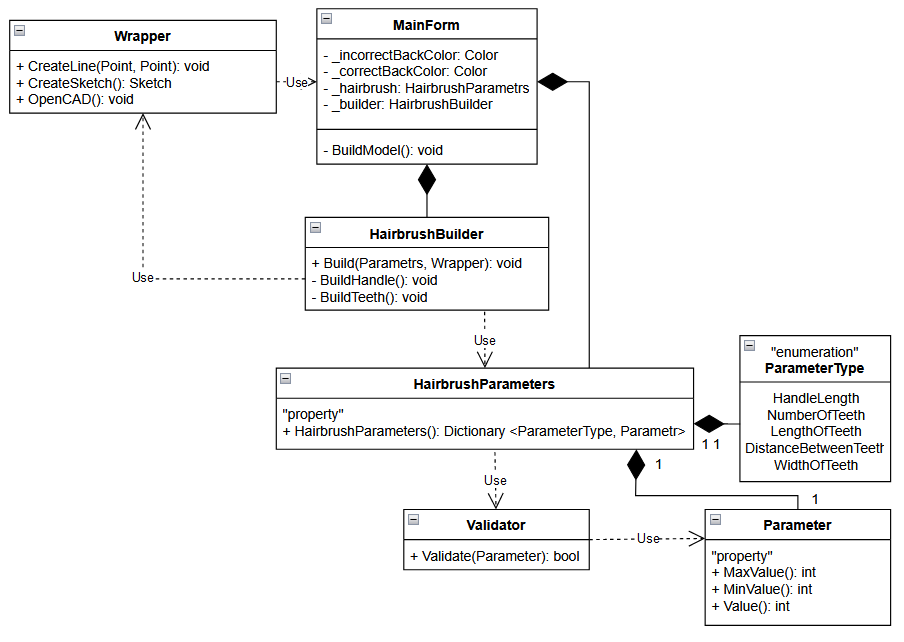


Рисунок 4.1 – Изначальная диаграмма классов

Разберем основные классы проекта:

 - MainForm – является главным окном приложения. Хранит в себе параметры (Parameters) и объект класса строителя модели (Builder);

- Parameters – класс, хранящий в себе параметры модели;

- TableBuilder – класс строитель модели;

- Wrapper – класс обёртка API КОМПАС. В нем находятся все нужные методы создания примитивов и документов, которые пригодятся для построения модели.

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

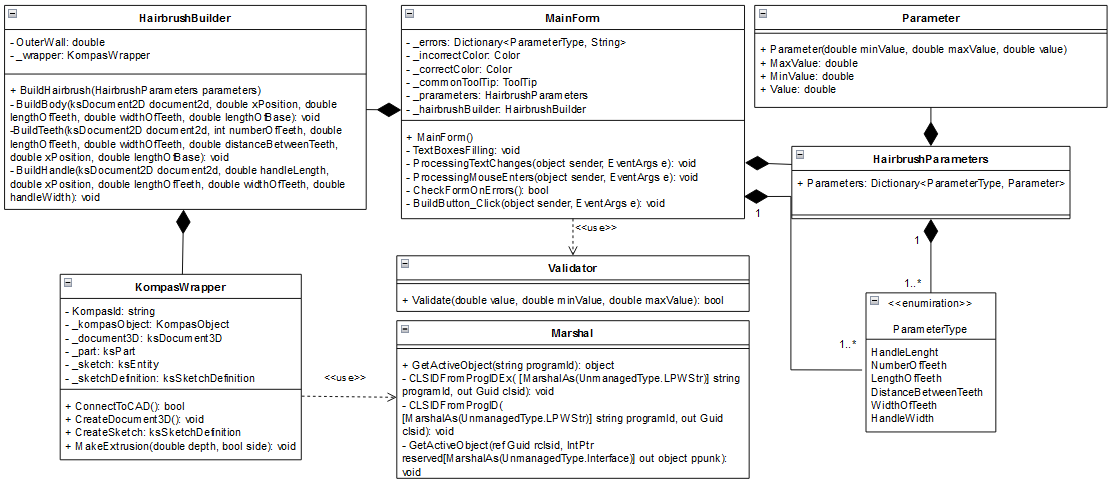


Рисунок 4.2 – Итоговая диаграмма классов

Разберем основные изменения классов проекта после реализации:

 –  Был добавлен новый класс Marshal, который Нужен для реализации метода GetActiveObject, который доступен только в .Net Framework.

 –  Был добавлен новый параметр, который позволяет регулировать ширину расчески.

 –  Были изменены названия для некоторых методов, также класс Wrapper теперь называется KompasWrapper.

 –  В классе KompasWrapper были добавлены новые методы и поля, такие как CreateDocument3D, MakeExtrusion и поля \_part, \_kompasObject и другие.

 –  Также изначально предполагалось запускать плагин непосредственно из Компас 3Д, но по итогу было реализован запуск САПР из плагина.

# 5 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ

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

На рисунке 5.1 представлен пользовательский интерфейс плагина.

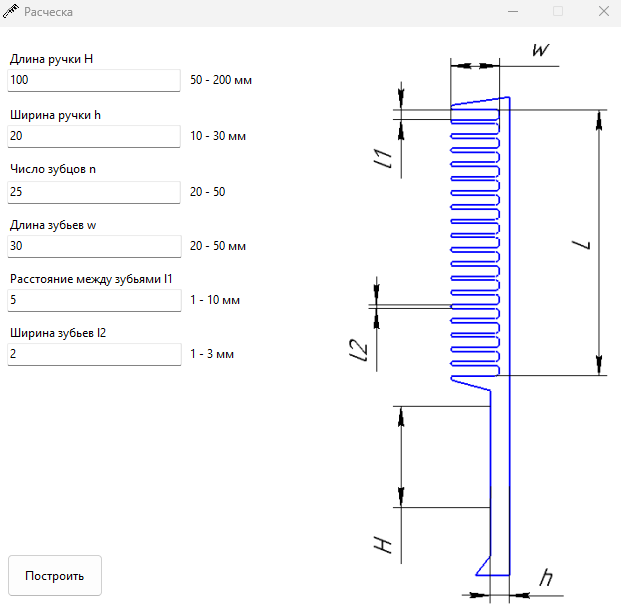


Рисунок 5.1 – Пользовательский интерфейс плагина

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

Результат построений представлен на рисунке 5.2.

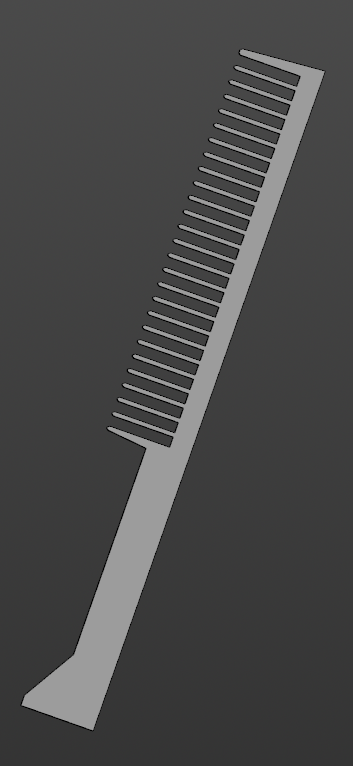


Рисунок 5.2 – Модель расчески в КОМПАС-3D

В случае некорректного ввода параметров модель, при нажатии на кнопку построения будет представлен список ошибок, и пользователь получит сообщение об ошибке при наведении мышки на неправильно заполненное поле (Рисунок 5.3).

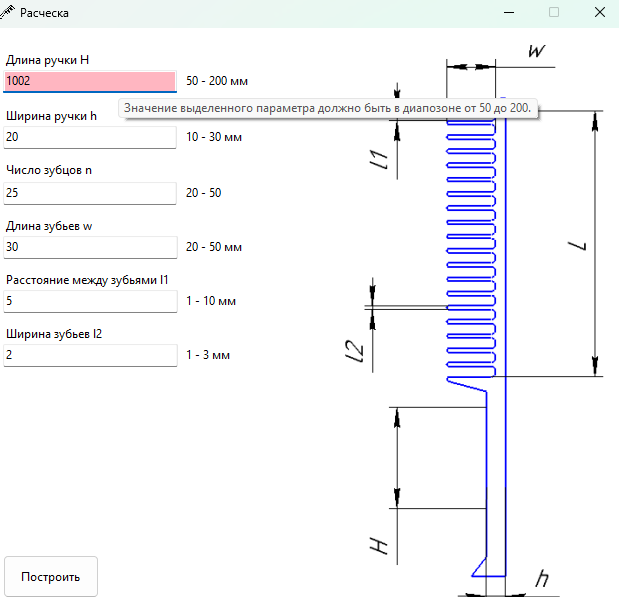


Рисунок 5.3 – Реакция плагина на некорректный ввод

Возможные варианты ошибок и их решений перечислены в таблице 5.1.

Таблица 5.1 – Ошибки и их варианты исправления

| **Текст ошибки** | **Варианты исправления** |
| --- | --- |
| Значение выделенного параметра должно быть в диапазоне от Min до Max | Ввести значение параметра в диапазоне от Min до Max |
| Некорректный формат данных | Убрать все символы из строки, не относящиеся к числам |

# 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 «Расческа», а именно соответствие полученного результата в виде трехмерной модели с входными параметрами.

Результаты тестирование минимальных (длина ручки 50 мм, ширина ручки 10 мм, число зубцов 20, длина зубцов 20 мм, расстояние между зубьями 1мм, ширина зубьев 1 мм) и максимальных (длина ручки 200 мм, ширина ручки 30 мм, число зубцов 50, длина зубцов 50 мм, расстояние между зубьями 10мм, ширина зубьев 3 мм) параметров модели «Расческа» представлена на рисунке 6.1.

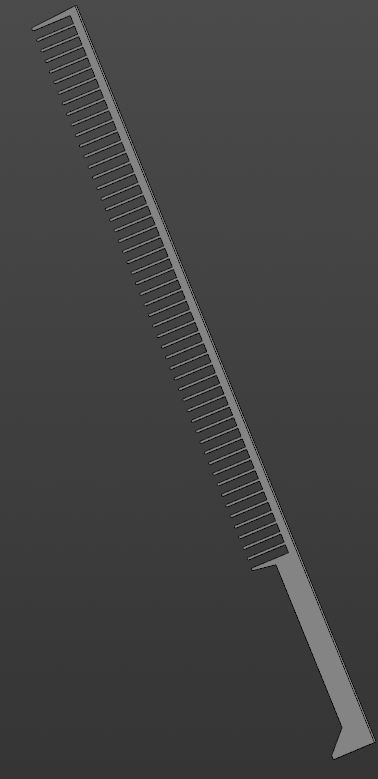
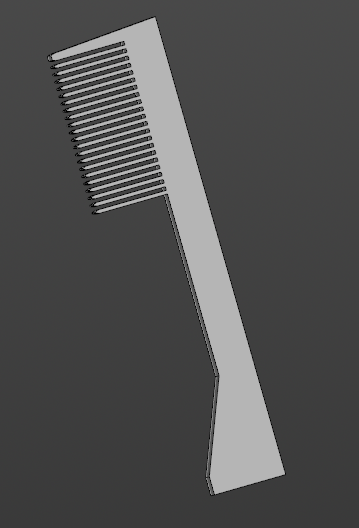


Рисунок 6.1 – Результаты построения минимальных (слева) и максимальных (справа) параметров модели «Расческа»

## **6.2 Модульное тестирование**

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit проведено модульное тестирование [7]. проверялись открытые поля и методы.

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

Степень покрытия проектов — сто процентов. Было написано 18 тестов.

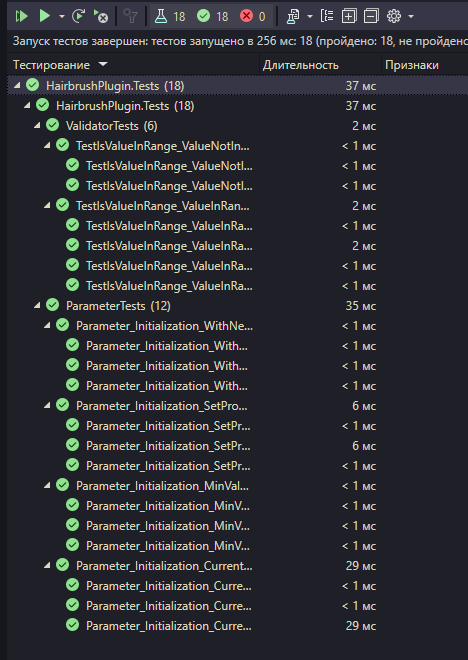


Рисунок 6.2 – Результаты модульного тестирования

## 

## **6.3 Нагрузочное тестирование**

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

* ЦП Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
* 16 ГБ ОЗУ (15.9 GB usable);
* Windows 64 bit

Тестирование зацикленного построения модели со следующими параметрами:

* длина ручки расчески 100 мм
* ширина ручки расчески 20 мм
* общее количество зубьев 25
* длина зубчиков расчески 30 мм
* расстояние между зубчиками расчески 5 мм
* ширина зубчиков расчески 2 мм

На рисунке 6.4 представлен график зависимости памяти ОЗУ от построения модели, а на рисунке 6.5 представлен график зависимости времени от построения модели.

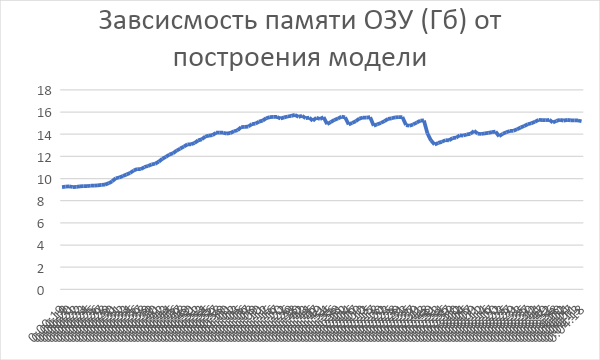


Рисунок 6.4 –График зависимости памяти ОЗУ от построения модели

Из рисунка 6.4 видно, что память изменяется линейно, но с некоторыми скачками вниз. Это связано с тем, что для регенерации памяти происходит остановка обмена данными, что снижает нагрузку на оперативную память. В конце график не сильно идет вверх, это связано с тем, что, когда на оперативную память идет сильная нагрузка, ОС начинает использовать файл подкачки для разгрузки оперативной памяти.



Рисунок 6.5 – График зависимости времени от построения модели

Из рисунка 6.5 видно, что время построения моделей изменяется линейно. Это связано с увеличением нагрузки на оперативную память и центральный процессор с каждым построением модели. За счет этого время построения модели увеличивается.

# Заключение

В ходе выполнения лабораторных работ были выполнены следующие действия:

- составлено техническое задание;

- составлен проект системы;

- реализован плагин;

- реализована дополнительная функциональность;

- составлена пояснительная записка.

Основные сложности возникли в ходе составления проекта системы, а конкретнее, с проектированием архитектуры плагина. В связи с чем далее пришлось несколько изменить и добавить новые методы и свойства классов.

Также в ходе выполнения лабораторных работ были получены навыки проектирования программных средств, составления проектной документации, тестирования ПО и работы с API Компас-3D.

В результате выполнения всех лабораторных был реализован плагин, позволяющий автоматизировать построение расчесок в САПР Компас-3D.

# Список источников

1. Visual Studio [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 26.12.2023).
2. КОМПАС-3D. Официальный сайт САПР КОМПАС [Электронный ресурс]. — Режим доступа: https://kompas.ru/ (дата обращения 26.12.2023).
3. Расческа – Википедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%87%D1%91%D1%81%D0%BA%D0%B0 (дата обращения: 18.10.2023).
4. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 10.12.2023).
5. Что такое WinfowsForms [Электронный ресурс]. – URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0 (дата обращения: 10.12.2023).
6. UML. Основы / Фаулер, М. – 3-е изд., пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
7. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 10.12.2023).