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 «РАСЧЕСКА» ДЛЯ «КОМПАС-3D»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

|  |  | Студент гр. 580-3  \_\_\_\_\_\_\_\_\_\_\_ В.В Ковалев  “ ” 2023 г. |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  оценка | Проверил доцент каф. КСУП, Кандидат технических наук  \_\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев  “ ” 2023 г. |

Томск 2023

Содержание

[1 Описание САПР 3](#_heading=h.gjdgxs)

[1.1 Информация о выбранной САПР 3](#_heading=h.30j0zll)

[1.2 Описание API 4](#_heading=h.1fob9te)

[2 Обзор аналогов плагина 9](#_heading=h.3znysh7)

[3 Описание предмета проектирования 10](#_heading=h.2et92p0)

[4 Проект системы 12](#_heading=h.tyjcwt)

[4.1 Диаграмма классов 12](#_heading=h.3dy6vkm)

[4.2 Макеты пользовательского интерфейса 15](#_heading=h.1t3h5sf)

[Список источников 17](#_heading=h.4d34og8)

# **1 Описание САПР**

# 1.1 Информация о выбранной САПР

КОМПАС 3-D - это компьютерная система автоматизированного проектирования в трехмерном пространстве, разработанная компанией ASCON.[1]

Основные возможности КОМПАС 3-D включают:

1. Создание и редактирование трехмерных моделей объектов с использованием специализированных инструментов.

2. Создание и редактирование технической документации: чертежей, схем, спецификаций и пр.

3. Расчет и анализ параметров объектов, включая прочностные расчеты и статический анализ.

4. Обмен данными с другими программами и форматами файлов, такими как AutoCAD, STEP, IGES, и др.

5. Возможность коллективной работы над проектами с использованием сетевых возможностей.

Аналогами КОМПАС 3-D могут являться следующие САПР:

1. AutoCAD - одна из самых распространенных САПР, обладающая широкими возможностями в области трехмерного моделирования и создания технической документации. КОМПАС 3-D был специально разработан для инженерных и промышленных предприятий, в то время как AutoCAD является более универсальным инструментом, который используется во многих отраслях, включая архитектуру, строительство и дизайн[2].

2. CATIA - мощная система для трехмерного моделирования и разработки продуктов разных отраслей промышленности. КОМПАС 3-D предоставляет возможности для 3D-моделирования и проектирования, в то время как CATIA предлагает более широкий спектр инструментов, охватывающих не только 3D-моделирование, но и анализ, симуляцию, проектирование электронных систем и т. д.[3]

3. SolidWorks - САПР, специализирующаяся на трехмерном моделировании и инженерном анализе. SolidWorks имеет более развитую систему моделирования, включая возможность создавать сложные контуры и поверхности, а также интегрировать внешние расчеты и анализ, в то время как Компас 3D более удобен для решения простых инженерных задач и быстрого создания чертежей.[4]

4. Siemens NX - комплексная система трехмерного моделирования, анализа и симуляции от компании Siemens. Siemens NX является более мощным инструментом для сложного инженерного проектирования и обладает более высокими возможностями в области поверхностного моделирования и интеграции с другими системами. КОМПАС 3-D ориентирован на простые инженерные задачи и обеспечивает простоту использования и скорость создания чертежей.[5]

Выбор конкретной САПР должен основываться на требованиях и предпочтениях пользователя, а также на особенностях конкретной области применения. КОМПАС 3-D может быть предпочтительным выбором в случаях, когда требуется сочетание относительно простого интерфейса и широких функциональных возможностей.

# 1.2 Описание API

API КОМПАС 3-D - это набор программных интерфейсов (API), доступных разработчикам для создания и управления моделями, созданными в программном обеспечении компас 3D.

API КОМПАС 3-D предоставляет разработчикам доступ к функциям моделирования, редактирования и визуализации трехмерных объектов. С его помощью разработчики могут создавать и изменять геометрические фигуры, применять различные техники моделирования (например, создание граничных поверхностей, сглаживание, вырезание и т. д.) и управлять пространственными свойствами объектов (например, положение, поворот, масштабирование).

API КОМПАС 3-D также позволяет разработчикам взаимодействовать с другими функциями программного обеспечения компас 3D, такими как создание чертежей и спецификаций, импорт и экспорт данных и управление привязками и параметрами.

С использованием API КОМПАС 3-D разработчики могут создавать собственные приложения и инструменты, расширяя возможности программного обеспечения КОМПАС 3-D и адаптируя его под свои потребности. Это позволяет автоматизировать определенные задачи моделирования, повысить производительность и облегчить совместную работу над проектами.

В таблицах 1.1-1.5 представлены основные свойства и методы интерфейсов KompasObject, IPart и ksDocument3D, ksDocument2D, [ksEntity](about:blank).

Таблица 1.1 – Методы интерфейса KompasObject

| Метод | Возвращаемое значение | Описание |
| --- | --- | --- |
| Document3D() | Указатель на интерфейс документа трёхмерной модели ksDocument3D. | Даёт возможность получить указатель на интерфейс трёхмерного документа (детали или сборки). |
| Visible() |  | Свойство видимости приложения. |
| Quit() |  | Метод для завершения программы Kompas-3D. |
| ActivateControllerAPI() |  | Метод для активации контроллера API. |

Окончание таблицы 1.1

| Метод | Возвращаемое значение | Описание |
| --- | --- | --- |
| ksDocument2D() |  | Интерфейс событий графического документа, события интерфейса позволяют контролировать состояние документа. |

Таблица 1.2 – Методы интерфейса IPart

| Метод | Входные параметры | Возвращаемое значение | Описание |
| --- | --- | --- | --- |
| GetDefaultEntity(short objType) | objType –тип объекта. | | Указатель на интерфейс [ksEntity](about:blank) или [IEntity](about:blank). | | --- | | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию. |
| GetPart(int type) | type – тип объекта. | Указатель на интерфейс компонента [ksPart](about:blank) или [IPart](about:blank). | Получить указатель на интерфейс компонента в соответствии с заданным типом. |
| NewEntity(short objType) | objType – тип объекта. | Указатель на интерфейс [ksEntity](about:blank) или [IEntity](about:blank). | Создать новый интерфейс объекта и получить указатель на него. |

Таблица 1.3 – Методы интерфейса ksDocument3D

| Метод | Входные параметры | Возвращаемое значение | Описание |
| --- | --- | --- | --- |
| Create (bool invisible, bool \_typeDoc) | invisible – признак режима редактирования документа  (TRUE – невидимый режим,  FALSE – видимый режим),  typeDoc – тип документа  (TRUE – деталь,  FALSE – сборка). | TRUE – в случае успешного завершения. | Дает возможность создать пустой документ (деталь или сборку) |
| GetPart(int type) | type – тип компонента из перечисления Типы компонентов. |  | Получить указатель на интерфейс компонента в соответствии с заданным типом |

Таблица 1.4 – Методы интерфейса ksDocument2D

| Метод | Входные параметры | Возвращаемое значение | Описание |
| --- | --- | --- | --- |
| long ksLineSeg (double x1, double y1, double x2, double y2, long style) | x1, y1 - координаты первой точки отрезка, x2, y2 - координаты второй точки отрезка, style - стиль линии. | указатель на отрезок - в случае удачного завершения, 0 - в случае неудачи. | Метод для создания отрезка. |

Таблица 1.5 —Методы интерфейса [ksEntity](about:blank)

| Метод | Входные параметры | Возвращаемое значение | Описание |
| --- | --- | --- | --- |
| ksBossExtrusionDefinition(BOOL forward, short type, double depth, double draftValue, BOOL draftOutward); | Forward- - направление выдавливания: TRUE - прямое направление, FALSE - обратное направление, type - тип выдавливания, depth - глубина выдавливания, draftValue - угол уклона, draftOutward - направление уклона: FALSE - уклон наружу, TRUE - уклон внутрь. | TRUE – в случае успешного завершения. | Метод выдавливает эскиз в одном направлении |
| ksCutExtrusionDefinition (BOOL forward, short type, double depth, double draftValue, BOOL draftOutward); | Forward- - направление выдавливания: TRUE - прямое направление, FALSE - обратное направление, type - тип выдавливания, depth - глубина выдавливания, draftValue - угол уклона, draftOutward - направление уклона: FALSE - уклон наружу, TRUE - уклон внутрь. | TRUE – в случае успешного завершения. | Метод вырезания выдавливанием эскиз в одном направлении |

# **2 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[6].

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.

# **3 Описание предмета проектирования**

Предметом проектирования является расческа. Расчёска — индивидуальный предмет быта для расчесывания волос, массажа головы и нанесения воды на волосы.[11]

Назначение разрабатываемого плагина обусловлено быстрым моделированием расчесок разных типов. Благодаря данному расширению, каждую расческу можно создать индивидуально под каждого клиента. На рисунке 2.1 представлена модель расчески.

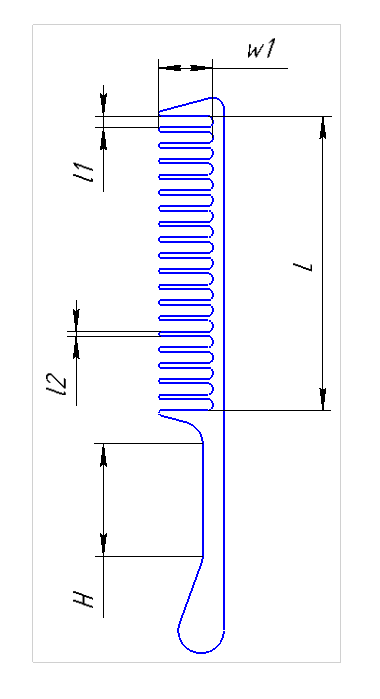


Рисунок 3.1 — Модель расчески с размерами

Изменяемые параметры для плагина(также все обозначения показаны на рис. 2.1):

* длина ручки расчески H (30 — 300мм);
* общее количество зубьев n (5 - 100)
* длина зубчиков расчески w (10 - 50мм)
* расстояние между зубчиками расчески l1 (1 - 30мм)
* ширина зубчиков расчески l2 (1 - 5мм)
* длина секции с зубчиками рассчитывается из формулы L = n \* (l1 + l2) + l1

# **4 Проект системы**

# 4.1 Диаграмма классов

Диаграммы UML (Unified Modeling Language) представляют собой графические инструменты, используемые для моделирования и визуализации различных аспектов системы или программного обеспечения. Они позволяют разработчикам и аналитикам представить идеи, концепции и структуру системы, а также её поведение и взаимодействие с внешним окружением. Диаграммы UML широко используются в разработке программного обеспечения и помогают в понимании, проектировании и документировании системы.[8]

На рисунке 4.1 представлена диаграмма классов.

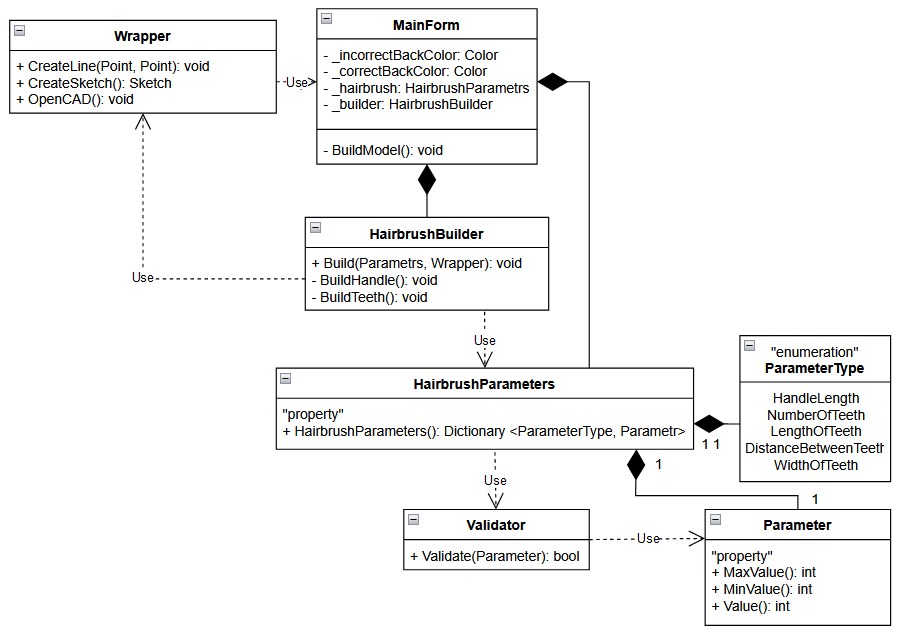


Рисунок 4.1 - Диаграмма классов

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

Parameters – класс, хранящий в себе параметры модели;

HairbrushBuilder – класс строитель модели;

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

Таблица 4.1 Свойства класса Parameter

| Название | Тип данных | Описание |
| --- | --- | --- |
| MaxValue | int | Максимальное значение параметра |
| MinValue | int | Минимальное значение параметра |
| Value | int | Значение параметра |

Таблица 4.2 Методы класса Validator

| Название | Входные параметры | Тип возвращаемых данных | Описание |
| --- | --- | --- | --- |
| Validate | Parameter | bool | Проверяет параметр на соответствие ограничений |

Таблица 4.3 Свойства класса HairbrushParameters

| Название | Тип данных | Описание |
| --- | --- | --- |
| HairbrushParameters | Dictionary | Коллекция объектов класса Parameter, внутри которого описаны ограничения и само значение |

Таблица 4.4 Методы класса HairbrushBuilder

| Название | Входные параметры | Тип возвращаемых данных | Описание |
| --- | --- | --- | --- |
| Build | HairbrushParameters, Wrapper | void | Построение полной модели расчески. |
| BuildHandle | - | void | Построение рукоятки расчески |
| BuildTeeth | - | void | Построение зубьев расчески |

Таблица 4.5 Поля класса MainForm

| Название | Тип данных | Описание |
| --- | --- | --- |
| \_incorrectBackColor | Color | Цвет, который отображается при некорректном вводе |
| \_correctBackColor | Color | Цвет, который отображается при корректном вводе |
| \_hairbrush | HairbrushParametrs | Экземпляр класса HairbrushParametrs |
| \_builder | HairbrushBuilder | Экземпляр класса HairbrushBuilder |

Таблица 4.6 Методы класса MainForm

| Название | Входные параметры | Тип возвращаемых данных | Описание |
| --- | --- | --- | --- |
| BuildModel | - | void | Построение полной модели расчески в Компас 3D |

# 4.2 Макеты пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров. При запуске программы в полях для ввода параметров отсутствуют значения. Пользователь может менять данные параметры (рисунок 4.2).

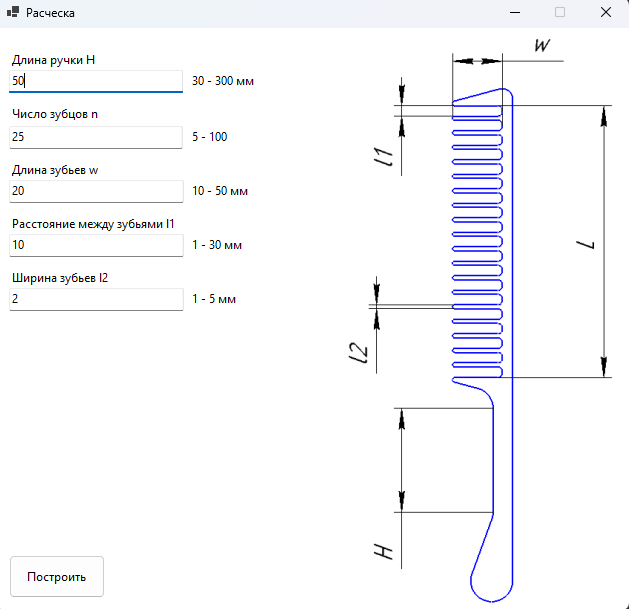


Рисунок 4.2 - Макет пользовательского интерфейса

Поле, где было введено некорректное значение изменит цвет на светло-красный (рисунок 4.3).

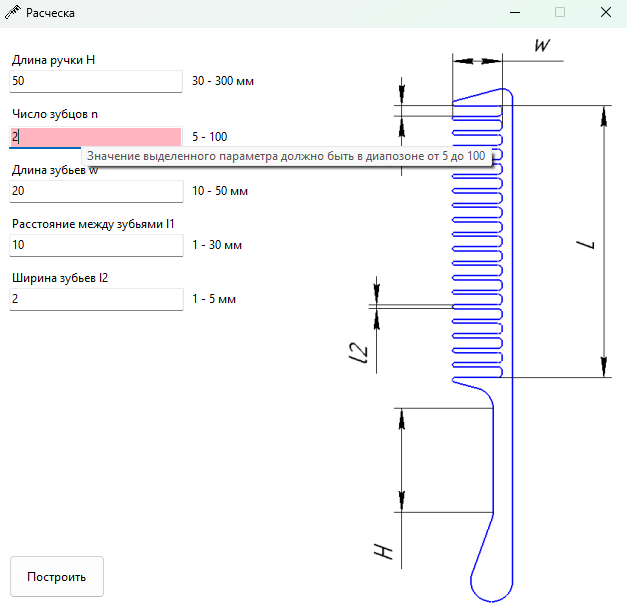


Рисунок 4.3 – Поля с некорректными параметрами

# **Список источников**

1. Компас 3D [Электронный ресурс]. – Режим доступа: https://kompas.ru/kompas-3d/about/ Дата обращения (15.10.2023)

2. AutoCad [Электронный ресурс]. – Режим доступа: https://www.autodesk.com/products/autocad/overview Дата обращения (15.10.2023)

3. CATIA [Электронный ресурс]. – Режим доступа: https://www.3ds.com/ru/produkty-i-uslugi/catia/ Дата обращения (15.10.2023)

4. SolidWorks [Электронный ресурс]. – Режим доступа: https://www.solidworks.com/ Дата обращения (15.10.2023)

5. Siemens NX[Электронный ресурс]. – Режим доступа: https://plm.sw.siemens.com/en-US/nx/ Дата обращения (15.10.2023)

6. K3D-Parametric[Электронный ресурс]. – Режим доступа: https://k3d.tech/ Дата обращения (15.10.2023)

7. Кидрук Максим. КОМПАС-3D V10 на 100% / М. Кидрук. – СПб.: Питер, 2009 – 560 с.

8. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. Изд: Символ-Плюс, 2011, с.192 (3-е издание)

9. Язык UML. Руководство пользователя. Изд: ДМК Пресс, 2015, с.496

10. Применение UML 2.0 и шаблонов проектирования. Введение в объектно- ориентированный анализ, проектирование и итеративную разработку. Изд: Вильямс, 2013, с.739 (3-е издание).

11. Расческа [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%87%D1%91%D1%81%D0%BA%D0%B0 Дата обращения (15.10.2023)