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 «ШАХМАТНЫЙ КОРОЛЬ» ДЛЯ САПР «КОМПАС 3D v18»

Проект системы по лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент группы 588-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соловьева А.Д.

«\_\_» \_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент КСУП:

\_\_\_\_\_\_\_\_\_ Калентьев A. А.

«\_\_» \_\_\_\_\_\_\_\_2021 г.

Томск 2021

1. **Описание САПР**
   1. **Описание программы**

Компас 3D – это комплексная система автоматизированного проектирования (САПР). Она направлена не только на создание объемных цифровых вариантов изделий, но и на разработку чертежей, проектирование различных систем (в том числе кабельных) и создание соответствующей документации[1].

**Функции**

* Твердотельное и параметрическое 3D моделирование;
* Построение чертежей и технической документации;
* Возможность проектирования изделий из листового металла;
* Возможность учесть всевозможные допуски, усадку, свойства материала и технологию производства будущего изделия.

**Особенности программы**

* Собственное ядро;
* Русскоязычный интерфейс;
* Интеграция с другими программами;
* Поддержка различных файловых форматов;
* Возможность проектирования трубопроводов, кабелей и кабельных систем;
* Встроенный модуль для создания электрических цепей.

**Преимущества**

* Простота в освоении;
* Обширная библиотека стандартизированных изделий;
* Доступная цена;
* Масштабное и продуманное проектирование в 2D;
* Возможность учета свойств большого количества материалов.

**Недостатки**

* Случаются проблемы при импорте 3D моделей из других программ;
* Проектировать в 3D сложнее, чем в 2D;
* Плохо реализована возможность реализации;
* Не слишком хорошо оформлена система поверхностного моделирования.
  1. **Описание API**

Сегодня встречаются задачи, решение которых не реализовано в CAD-системах. Чаще всего это очень узкоспециализированные задачи, которые встречаются на каком-то конкретном предприятии или подотрасли. Для решения подобных задач есть возможность использовать КОМПАС-3D как платформу и на базе него создать свое приложение, которое позволит автоматизировать решение таких задач. Для создания таких приложений в КОМПАС-3D есть открытый API[2].

Главным интерфейсом API системы КОМПАС является KompasObject. Основные методы этого интерфейса представлены в таблице 1.1.

Таблица 1.1– Основные методы интерфейса KompasObject

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| ksError() | Выводит сообщение об ошибке |
| Document2D() | Позволяет получить указатель на интерфейс графического документа (чертежа или фрагмента) |
| Document3D() | Позволяет получить указатель на интерфейс графического документа (чертежа или сборки) |
| GetMatematic2D() | Возвращает указатель на интерфейс для работы с математическими функциями в графическом документе |
| ksDetachKompasLibrary() | Отключает библиотеку |
| Quit() | Закрывает КОМПАС |

Свойства интерфейса ksDocument3D представлены в таблице 1.2.

Таблица 1.2 – Основные методы интерфейса ksDocument3D

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| Close() | Позволяет закрыть документ |
| Create() | Дает возможность создать пустой документ (деталь или сборку) |
| DeleteObject() | Позволяет удалить трехмерный объект |
| GetPart() | Возвращает указатель на интерфейс компонента (детали или подсборки) в сборке |
| IsActive() | Дает возможность проверить, активен ли документ |
| RebuildDocument() | Позволяет перестроить документ |

Метод ksDocument3D::GetPart возвращает указатель на интерфейс детали или компонента сборки – ksPart. Свойства и методы этого интерфейса управляют состоянием компонентов сборки, они почти полностью дублируют команды контекстного меню и панели свойств, доступные пользователю при работе с тем или иным компонентом. Свойства интерфейса ksPart представлены в таблице 1.3.

Таблица 1.3 – Свойства интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Тип данных** | **Описание** |
| excluded | WordBool | Определяет, исключен ли из расчета компонент |
| fileName | WideString | Имя файла, из которого вставлен документ |
| fixedComponent | WordBool | Определяет, является ли компонент зафиксированным |
| hidden | WordBool | Задает видимость компонента (скрыт или нет) |

Продолжение таблицы 1.3

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Тип данных** | **Описание** |
| name | WideString | Имя компонента в дереве построений |
| StandartComponent | WordBool | Определяет, является ли данный компонент стандартным (библиотечным элементом) |

* 1. **Обзор аналогов**

**Houdini**

Houdini - Масштабный программный пакет для создания персонажей и сцен для кино, телевидения и видеоигр, в том числе в виртуальной реальности.

Houdini поддерживает полигональное, сплайновое и физическое моделирование, частицы, вокселы, а также анимацию. В новой версии появилась процедурная генерация персонажей, интерактивные кисти для рисования тканей, проводов и мягких тел, быстрые алгоритмы рендеринга.[3]

На рисунке 1.1 представлен интерфейс приложения Houndini.

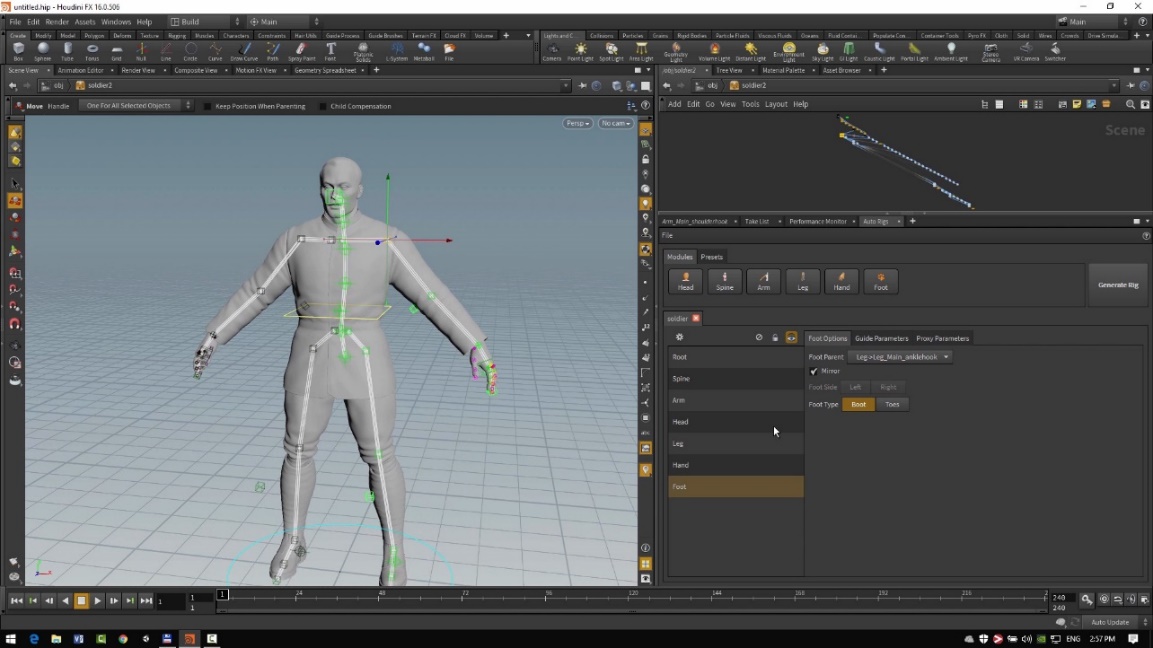


Рисунок 1.1 – Интерфейс Houndini.

1. **Описание предмета проектирования**

**Король.** Шахматная фигура, с крестом на голове, похожая на корону.

К изменяемым параметрам фигуры относятся:

1. Высота фигуры: от 40 мм до 1м;
2. Высота тела: от 20мм до 40мм высоты фигуры;
3. Высота креста: от 10мм до 15мм высоты фигуры;
4. Диаметр основания: от 20мм до 30мм высоты фигуры;
5. Диаметр верхней части головы: от 10мм до 15мм высоты фигуры;
6. Диаметр нижней части головы: от 11мм до 16мм высоты фигуры;
7. Высота поддержки креста: от 1мм до 3мм высоты фигуры.

Ограничение: Диаметр верхней и нижней части головы не может быть больше диаметра основания.

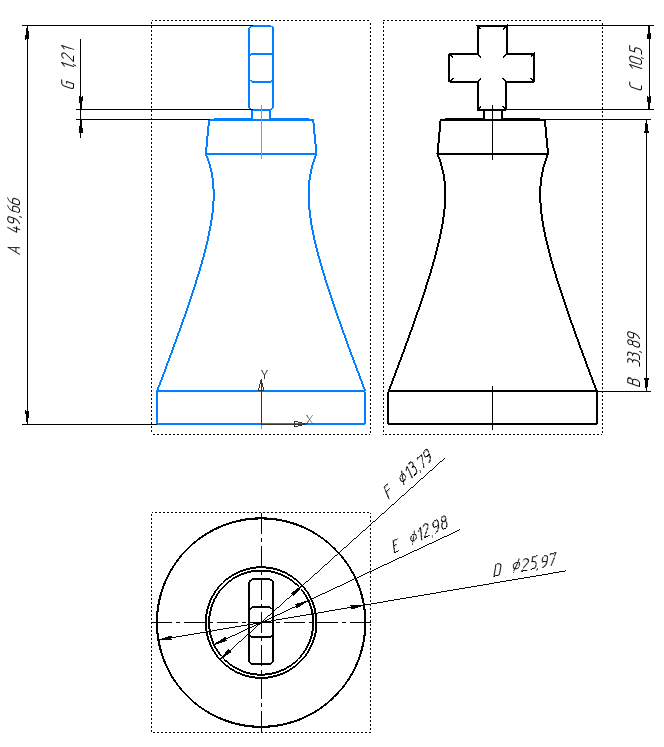


Рисунок 2.1 – Чертеж модели с обозначениями.

1. **Проект диаграммы**
   1. **Диаграмма классов**

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами [4].

Диаграмма классов для данного проекта представлена на рисунке 3.1.

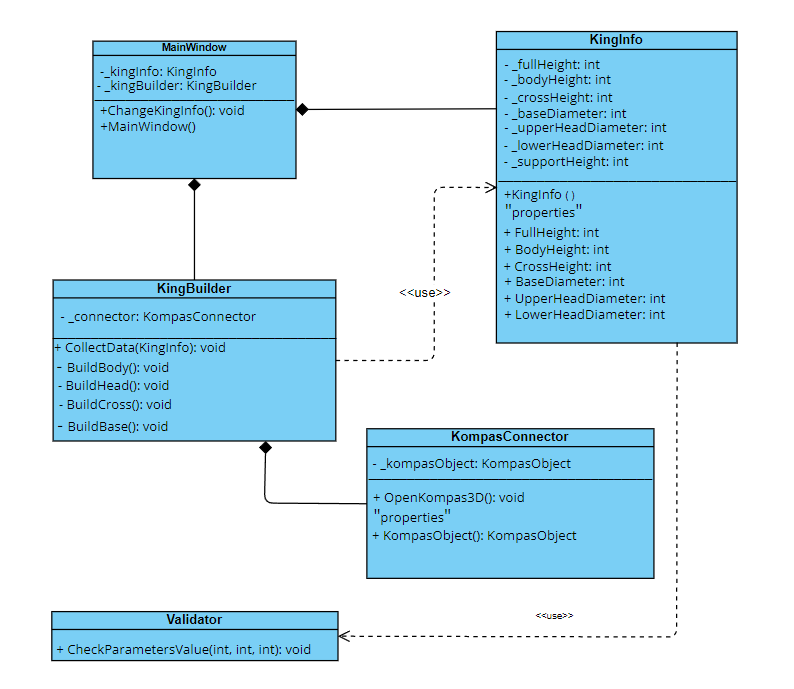


Рисунок 3.1 – UML-диаграмма классов.

В таблице 3.1 представлено описание полей и методов класса MainWindow.

Таблица 3.1 – Описание полей и методов класса MainWindow.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| \_kingInfo: KingInfo | Данные для построения короля |
| - \_kingBuilder: KingBuilder | Построение модели |
| MainWindow() | Конструктор |
| ChangeKingInfo() | Изменение данных для построения короля |

В таблице 3.2 представлено описание свойств и методов класса KingInfo.

Таблица 3.2 – Описание свойств и методов класса KingInfo.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| FullHeight: int | Полная высота фигуры |
| BodyHeight: int | Высота верхнего основания |
| CrossHeight: int | Высота креста основания |
| BaseDiameter: int | Диаметр основания |
| UpperHeadDiameter: int | Диаметр верха головы |
| LowerHeadDiameter: int | Диаметр низа головы |
| SupportHeight: int | Высота поддержки головы |
| KingInfo() | Конструктор |

В таблице 3.3 представлено описание полей и методов класса KingBuilder.

Таблица 3.3 – Описание полей и методов класса KingBuilder.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| + CollectData(KingInfo): void | Сбор данных из KingInfo |
| - BuildBody(): void | Построение тела |
| - BuildHead(): void | Построение головы |
| - BuildCross(): void | Построение креста |
| - BuildBase(): void | Построение основания |

В таблице 3.4 представлено описание полей и методов класса Validator.

Таблица 3.4 – Описание полей и методов класса Validator.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| + CheckParametersValue(int, int): void | Проверка введеных параметров |

В таблице 3.5 представлено описание полей и методов класса KompasConnector.

Таблица 3.5 – Описание свойств и методов класса KompasConnector.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| OpenKompas3D(): void | Открытие Компаса |
| KompasObject() | Получение KompasObject |
| \_kompasObject: KompasObject | Интерфейс API-системы КОМПАС |

* 1. **Макеты пользовательского интерфейса**

Макеты пользовательского интерфейса представлены на следующих рисунках.

Первоначальный вид приложения представлен на рисунке 3.2.

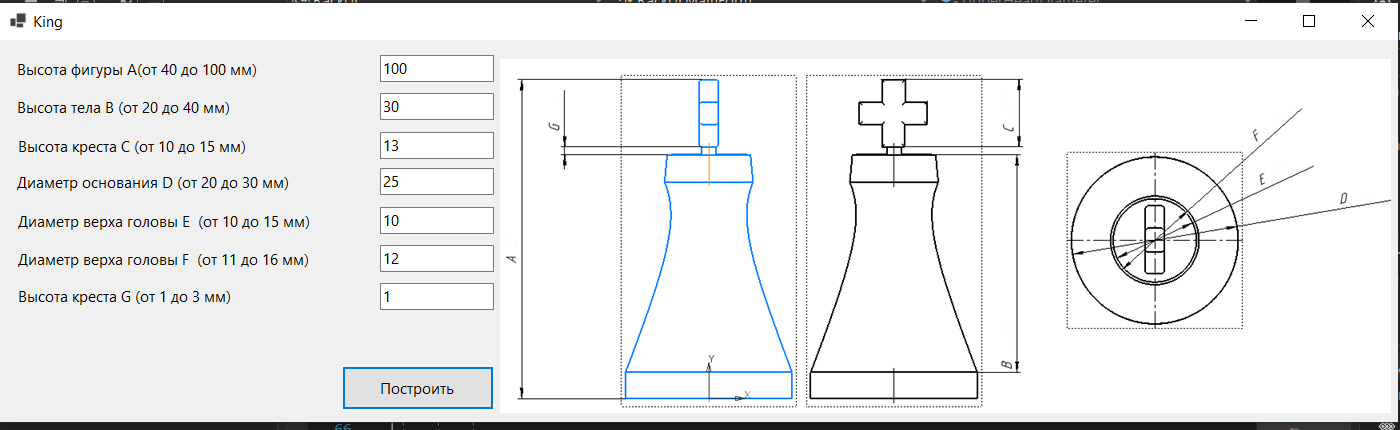


Рисунок 3.2 – Первоначальный вид приложения.

В разрабатываемом приложении предусмотрена валидация значений в поле для ввода. При вводе значения, удовлетворяющего соответствующему диапазону, текстовое поле остается белым цветом. В иных случаях – окрашивается в красный.

Пример работы валидации представлен на рисунках 3.3 и 3.4.

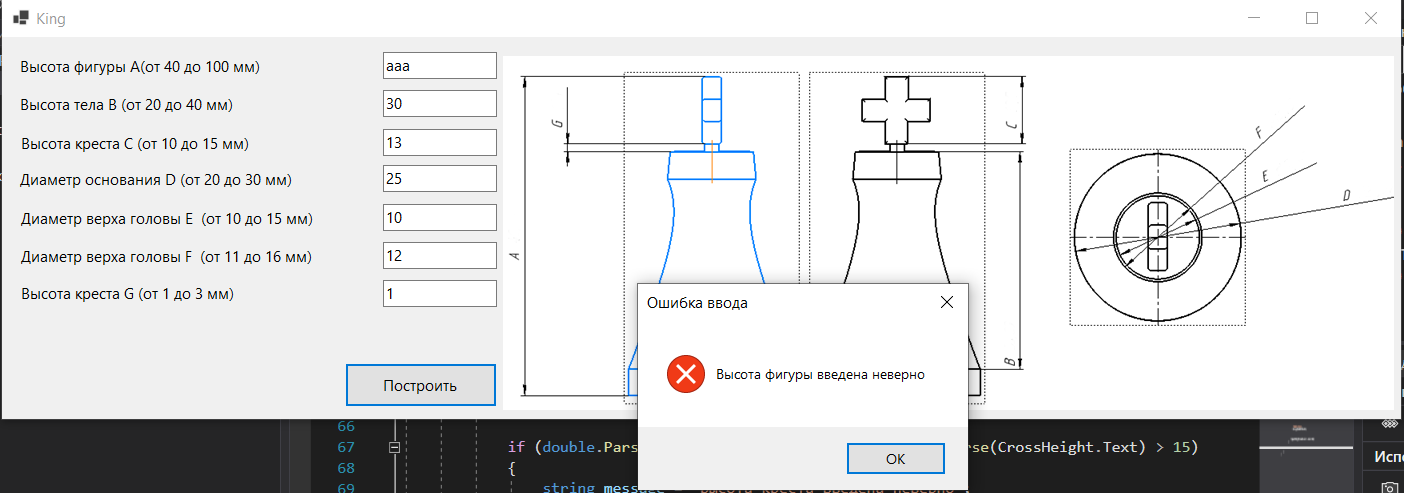


Рисунок 3.3 – Пример работы валидации.

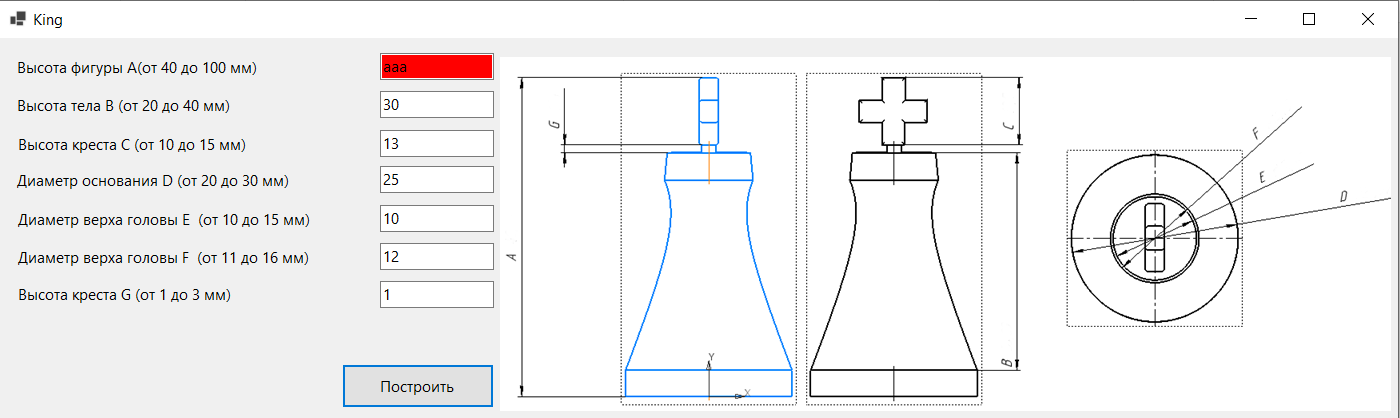


Рисунок 3.3 – Пример работы валидации.

**Список использованных источников**

1. Обзор Компас 3D|Обзор программы для новичков [Электронный ресурс]: 3DDEVICE.COM – режим доступа к статье: <https://3ddevice.com.ua/blog/3d-printer-obzory/obzor-kompas-3d/> (дата обращения 10.12.2021)
2. Базовые интерфейсы API системы КОМПАС [Электронный ресурс]: PROGRAMMING-LANG.COM – режим доступа к статье: <http://programming-lang.com/ru/comp_soft/kidruk/1/j196.html> (дата обращения 10.12.2021)
3. Программы для 3D-моделирования [Электронный ресурс]: - режим доступа к статье: <https://lifehacker-ru.turbopages.org/lifehacker.ru/s/programmy-dlya-3d-modelirovaniya/> (дата обращения 12.12.2021)
4. Введение в UML от создателей языка. / Г. Буч, Д. Рамбо, И. Якобсон, 2-е изд. – ДМК Пресс, 2015 – 496 с. (дата обращения 10.12.2021)