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 “ГЕНЕРАТОР ПИВНОЙ КРУЖКИ” ДЛЯ САПР КОМПАС 3D

Проект системы по дисциплине

«Основы разработки САПР»

Выполнил:

Студент гр. 589-3

\_\_\_\_\_\_\_\_\_ Дерябин Н.А.

Руководитель

к.т.н., доцент каф. КСУП

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

(оценка) \_\_\_\_\_\_\_\_\_

Томск 2022 г

Оглавление

[1 ОПИСАНИЕ САПР 3](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583343)

[1.1 Описание САПР «КОМПАС-3D v21»…………...………………………... 3](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583344)

[1.2 Анализ API…………………………………………………………………..4](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583345)

[1.3 Обзор аналогов…………………………………………………………….10](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583346)

[2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ 13](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583347)

[3 ПРОЕКТ СИСТЕМЫ 15](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583348)

[3.1 Диаграмма классов………………………………………………………...15](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583349)

[3.2 Макеты пользовательского интерфейса………………………………….19](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583350)

[Список использованных источников 21](file:///C:\Users\Admin\Downloads\ORSAPR%20Project.docx#_Toc116583351)

# 1 ОПИСАНИЕ САПР

# 1.1 Описание САПР «КОМПАС-3D v21»

**КОМПАС-3D v21** — Одна из программ семейства систем автоматизированного проектирования «Компас», позволяющая создавать трехмерные модели и сборки из них. КОМПАС-3D использует собственное математическое ядро и параметрические технологии, что является особенностью данной программы. В нее, помимо системы трехмерного моделирования, также входит универсальная САПР «Компас-График», созданная для составления чертежей и графиков.

# 1.2 Анализ API

API (англ. Application Programming Interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. Все эти коммуникации происходят с помощью функций, классов, методов, структур, а иногда констант одной программы, к которым могут обращаться другие.

Для КОМПАС-3D созданы две различные версии API - версии 5 и версии 7. К ним разработчик прилагает справочную систему по всем включенным в эту API интерфейсам.

Далее будут приведены самые важные для использования в программе методы и свойства интерфейсов. Наиболее вероятно, программа не ограничится их использованием, а будет также применять более локальные методы и свойства:

Таблица 1.1 – Интерфейсы, используемые при разработке

|  |  |  |
| --- | --- | --- |
| Название интерфейса | Описание интерфейса | |
| KompasObject | Интерфейс API КОМПАС | |
| ksEntity | Интерфейс элемента модели (оси, плоскости, формообразующего элемента) | |
| ksDocument2D | Интерфейс графического документа системы КОМПАС | |
| ksSketchDefinition | Интерфейс параметров эскиза | |
| ksDocument3D | | Интерфейс документа-модели |
| ksPart | | Интерфейс детали или подсборки в составе сборки |
| ksBaseExtrusionDefinition | | Интерфейс параметров основания - элемента выдавливания |
| ksCircularPartArrayDefinition | | Интерфейс операции копирования по окружности |

В нижеописанных таблицах представлены методы, которые будут использоваться при разработке плагина, а также описание входных параметров данных методов (таблицы 1.2 – 1.15).

Таблица 1.2 – Используемые методы интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| ActivateControllerAPI() | bool | Метод для активации API КОМПАС-3D |
| Visible() | bool | Свойство видимости приложения |

Таблица 1.3 – Используемые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |

Таблица 1.4 – Используемые методы интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| ksCircle (double xc, double yc, double rad, long style) | Указатель на окружность – в случае удачного завершения, 0 – в случае неудачи | Создать окружность |

Таблица 1.5 – Описание входных параметров, используемых методов интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| ksCircle (double xc, double yc, double rad, long style) | xc, yc | Координаты центра окружности |
| rad | Радиус окружности |
| style | Стиль линии |

Таблица 1.6 – Используемые методы интерфейса ksSketchDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| BeginEdit() | bool | Войти в режим редактирования эскиза (ksDocument2D) |
| EndEdit() | bool | Выйти из режима редактирования эскиза |

Таблица 1.7 – Используемые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Create (bool invisible, bool typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| GetPart (int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

Таблица 1.8 – Описание входных параметров, используемых методов интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| Create (bool invisible, bool typeDoc) | invisible | Признак режима редактирования документа (TRUE – невидимый режим, FALSE – видимый режим) |
| typeDoc | Тип документа (TRUE – деталь, FALSE – сборка) |
| GetPart (int type) | type | Тип компонента из перечисления: pInPlace\_Part – компонент, редактируемый на месте; pNew\_Part – новый компонент; pEdit\_Part –редактируемый компонент; pTop\_Part – главный компонент, в составе которого находится новый или редактируемый или указанный компонент (например, сборка, в составе которой находится редактируемая деталь) |

Таблица 1.9 – Используемые методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| GetDefaultEntity (short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| NewEntity (short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

Таблица 1.10 – Описание входных параметров, используемых методов интерфейса ksPart

|  |  |
| --- | --- |
| Входной параметр | Описание параметра |
| objType | Тип объекта |

Таблица 1.11 – Используемые типы объектов в методах интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Метод | Тип объекта | Название объекта |
| GetDefaultEntity (short objType) | o3d\_planeXOY | Плоскость XOY |
| o3d\_axisOZ | Ось OZ |
| NewEntity (short objType) | o3d\_sketch | Эскиз |
| o3d\_baseExtrusion | Базовая операция выдавливания |
| o3d\_cutExtrusion | Вырезать выдавливанием |
| o3d\_circularCopy | Операция копирования по концентрической сетке |

Таблица 1.12 – Используемые методы интерфейса ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | bool | Установить параметры выдавливания в одном направлении |
| SetSketch (LPDISPATCH sketch) | bool | Задать указатель на интерфейс эскиза элемента |

Таблица 1.13 – Описание входных параметров, используемых методов интерфейса ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| SetSideParam (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 |
| SetSketch (LPDISPATCH sketch) | sketch | Указатель на интерфейс эскиза ksEntity |

Таблица 1.14 – Используемые методы интерфейса ksCircularPartArrayDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetCopyParamAlongDir (long count, double step, bool, factor, bool dir) | bool | Установить параметры копирования |
| SetAxis (LPDISPATCH axis) | bool | Установить указатель на ось копирования |

Таблица 1.15 – Описание входных параметров, используемых методов интерфейса ksCircularPartArrayDefinition

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| SetCopyParamAlongDir (long count, double step, bool factor, bool dir) | count | Количество копий |
| step | Шаг |
| factor | Признак полного шага |
| dir | Направление |
| SetAxis (LPDISPATCH axis) | axis | Указатель на интерфейс оси ksEntity |

**1.3 Обзор аналогов**

Аналогом для приложения генерации 3D моделей пивной кружки можно посчитать плагин Archimesh для Blender. С помощью данного аддона можно создавать различные поверхности для создания комнаты, а также, что будет рассмотрено в данной работе, различные предметы интерьера. Работа данного плагина будет показана на примере лампы. Меню различных предметов, добавляемых Archimesh в блендер представлено на рисунке 1.1.



Рисунок 1.1 – Меню предметов

Концепция плагина во многом похожа с планируемым приложением. Есть набор параметров, представленных на рисунке 1.2 – размеров примитивов, из которых состоит модель. Если будут меняться параметры, то будет меняться и готовый объект. Изменения объекта в зависимости от параметров представлено на рисунках 1.3 и 1.4.



Рисунок 1.2 – Изменяемые параметры лампы



Рисунок 1.3 – Сгенерированная лампа

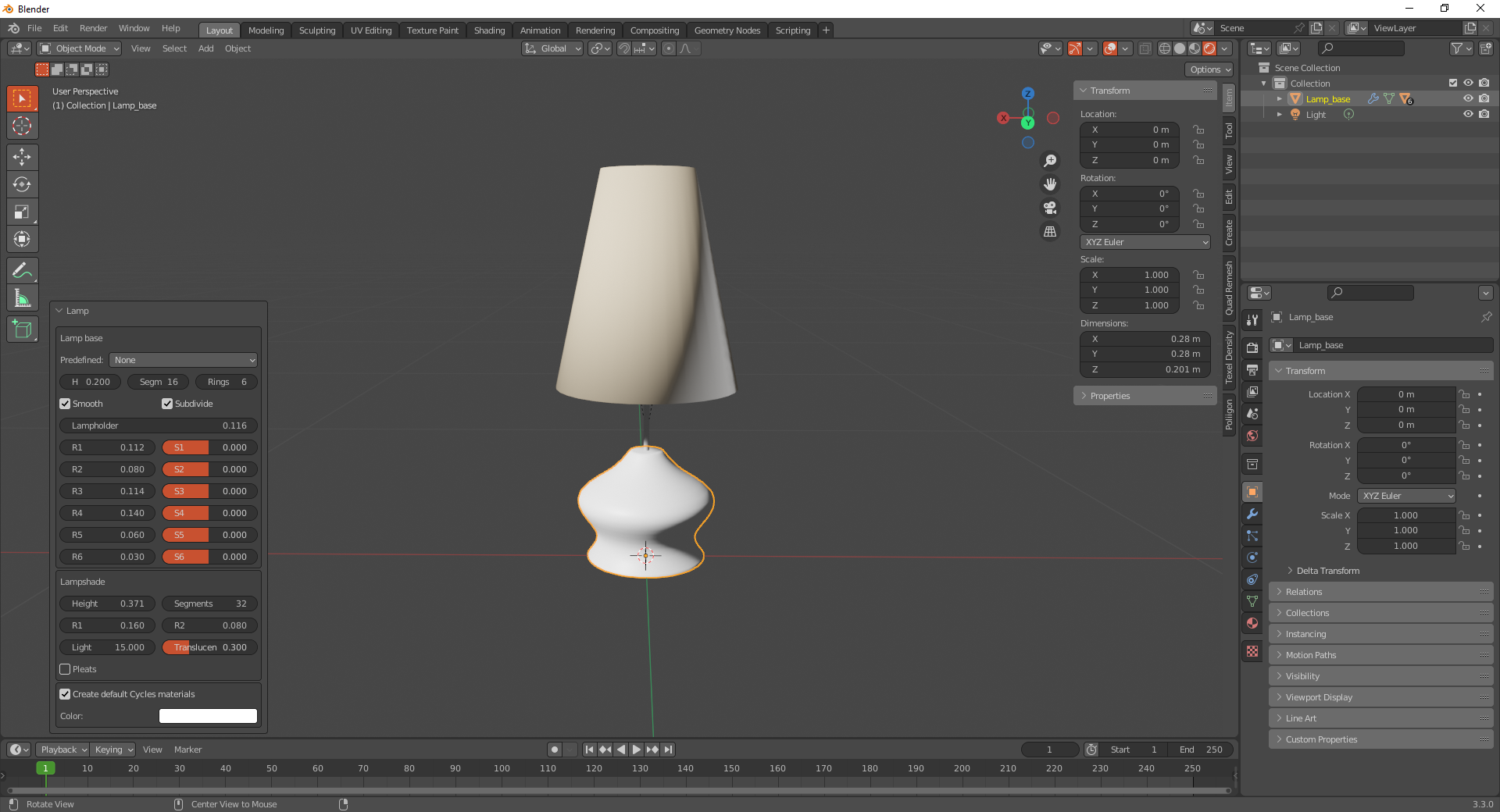


Рисунок 1.4 – Сгенерированная лампа с изменёнными параметрами

# 2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ

Программа предназначена для автоматизации моделирования детали «Пивная кружка»

Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях.

При запуске моделирования с некорректными значениями программа выводит сообщение об ошибке и отменяет построение модели.

При правильно введенных значениях результатом работы программы будет созданная по ним модель пивной кружки.

Изображения моделируемого объекта см. рис. 2.1-2.2:

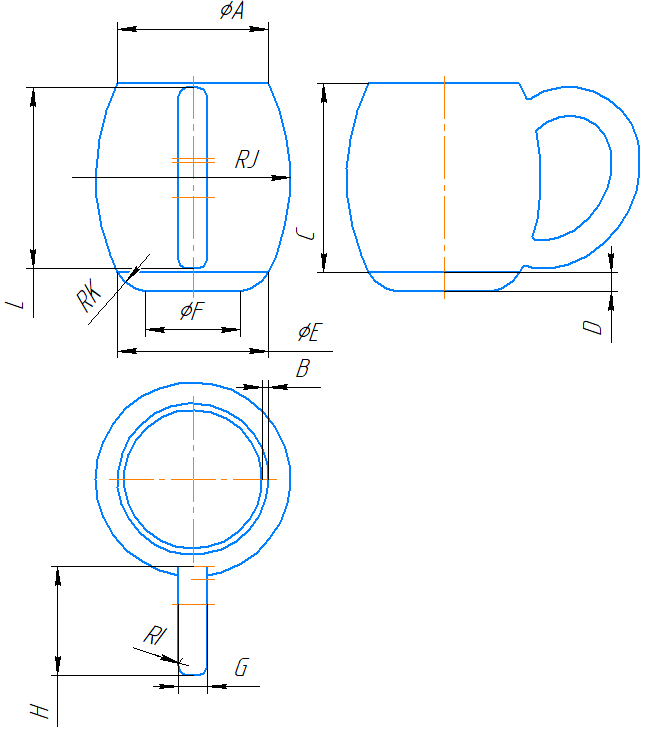


Рисунок 2.1 – Чертёж модели «Пивная кружка»



Рисунок 2.2 – 3D модель пивной кружки

Измеряемые параметры для плагина:

* A – Диаметр внешней части горла кружки (мин. 80 мм, макс. 100 мм);
* B – Толщина стенок кружки (мин. 5 мм, макс. 7 мм);
* C – Высота от горла кружки до дна (мин. 100 мм, макс. 165 мм);
* D – Толщина дна (мин. 10 мм, макс. 16.5 мм). Высота от горла кружки до дна (C) относится к толщине дна (D) как 10 к 1;
* E – Диаметр дна сверху (мин. 80 мм, макс. 100 мм). Отношение диаметра внешней части горла кружки (A) к диаметру дна сверху (E) равно 1 к 1;
* F – Диаметр дна снизу (мин. 50 мм, макс. 70 мм). Диаметр дна сверху (E) больше диаметра дна снизу на 30 мм;

# 3 ПРОЕКТ СИСТЕМЫ

# 3.1 Диаграмма классов

Унифицированный язык моделирования (UML) — это язык моделирования общего назначения, предназначенный для обеспечения стандартного способа визуализации проектирования системы. В 1997 году UML был принят в качестве стандарта Object Management Group (OMG) и с тех пор управляется этой организацией.

Диаграмма классов UML представлена на рисунке 3.1.

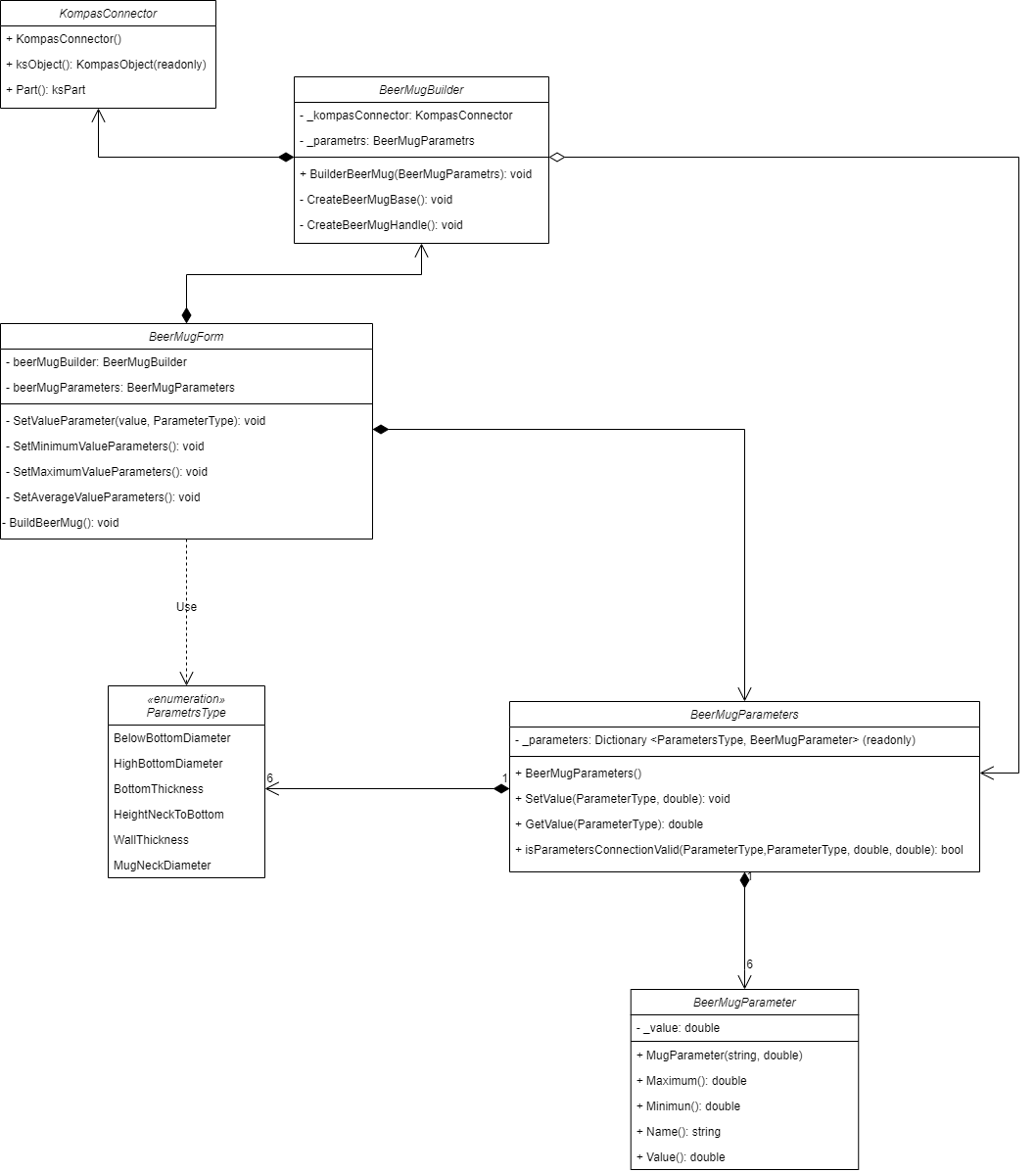


Рисунок 3.1 – Диаграмма классов UML

Далее в таблицах 3.1 – 3.4 представлено описание классов.

Таблица 3.1 – Описание полей, методов, сущностей класса «BeerMugForm»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| * beerMugBuilder: BeerMugBuilder |  | Хранит в себе набор методов для построения пивной кружки |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| * beerMugParameters: beerMugParameters |  | Хранит в себе параметры необходимые для построения пивной кружки |
| * SetValueParameter(value, ParameterType) | void | Устанавливает значение параметра |
| - SetMinimumValueParameters() | void | Устанавливает минимальное значение всех параметров |
| * SetMaximumValueParameters() | void | Устанавливает максимальное значение всех параметров |
| - SetAverageValueParameters() | void | Устанавливает среднее значение всех параметров |
| BuildBeerMug() | void | Построение кружки по заданным параметрам |

Таблица 3.2 – Описание полей, методов, сущностей класса «BeerMugParameters»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| * parameters: Dictionary <ParametersType, BeerMugParameter> |  | Хранит данные о каждом параметре модели из перечисления «BeerMugParameter» |
| + BeerMugParameters() |  | Конструктор для создания экземпляра класса |
| + SetValue(ParameterType, double) | void | Устанавливает значение определённого параметра |
| + GetValue(ParameterType) | double | Возвращает значение определённого параметра |
| + IsParametersConnectionValid (ParameterType, ParameterType, double, double) | bool | Проверка на соотношение размеров параметров |

Таблица 3.3 – Описание полей, методов, сущностей класса «BeerMugParameter»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_value | double | Поле, хранящее значение |
| + MugParameter(string, double) |  | Конструктор для создания экземпляра класса |
| + Maximum() | double | Геттер для возвращение максимального значения параметра |
| + Minimum() | double | Геттер для возвращение минимального значения параметра |
| + Value() | double | Сеттер и геттер для поля, хранящего значение |

Таблица 3.4 – Описание полей, методов, сущностей класса «BeerMugBuilder»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| * kompasConnector: KompasConnector |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| * \_parameters: BeerMugParameters |  | Хранит данные о каждом параметре модели из перечисления «BeerMugParameter» |
| + BuilderBeerMug(BeerMugParameters) | void | Построение пивной кружки по заданным параметрам |
| * CreateBeerMugBase() | void | Построение основы кружки |
| * CreateBeerMugHandle() | void | Построение ручки пивной кружки |

Класс BeerMugParameters хранит в себе данные из перечисления ParametersType. BeerMugParameter хранит в себе текущие, минимальное и максимальное значения. Класс BeerMugBuilder непосредственно строит 3D модель на основе списка параметров из BeerMugParameters. Класс BeerMugForm нужен, чтобы связать представление программы с ее моделью. KompasConnector необходим для связи с Компас 3D.

**3.2** **Макеты пользовательского интерфейса**

На рисунке 3.2 представлен макет пользовательского интерфейса.

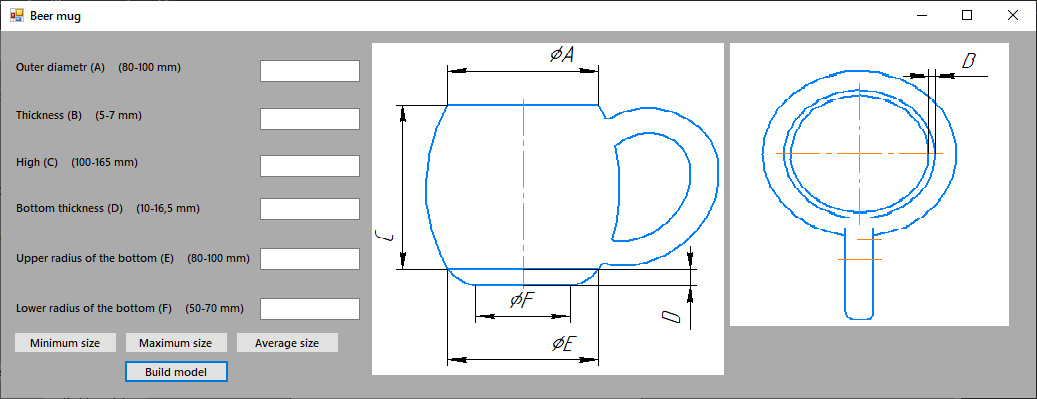


Рисунок 3.2 – Макет пользовательского интерфейса

С помощью данного окна пользователь может изменять параметры будущей 3D модели пивной кружки, где:

* A – Диаметр внешней части горла кружки (мин. 80 мм, макс. 100 мм);
* B – Толщина стенок кружки (мин. 5 мм, макс. 7 мм);
* C – Высота от горла кружки до дна (мин. 100 мм, макс. 165 мм);
* D – Толщина дна (мин. 10 мм, макс. 16.5 мм). Высота от горла кружки до дна (C) относится к толщине дна (D) как 10 к 1;
* E – Диаметр дна сверху (мин. 80 мм, макс. 100 мм). Отношение диаметра внешней части горла кружки (A) к диаметру дна сверху (E) равно 1 к 1;
* F – Диаметр дна снизу (мин. 50 мм, макс. 70 мм). Диаметр дна сверху (E) больше диаметра дна снизу на 30 мм;

Напротив полей ввода находится название компонента, за который поле отвечает и корректные размеры – минимальный и максимальный.

При нажатии на кнопку «Minimum size» будет создана 3D модель с минимальными корректными размерами. При нажатии на кнопку «Average size» будет создана 3D модель со средними корректными размерами. При нажатии на кнопку «Maximum size» будет создана 3D модель с максимальными корректными размерами. При нажатии на кнопку «Build model» будет построена 3D модель по заданным параметрам. Чертёж модели справа необходим для лучшего понимания расположения вводимых размеров.

На рисунке 3.3 представлено окно с некорректно введёнными данными.

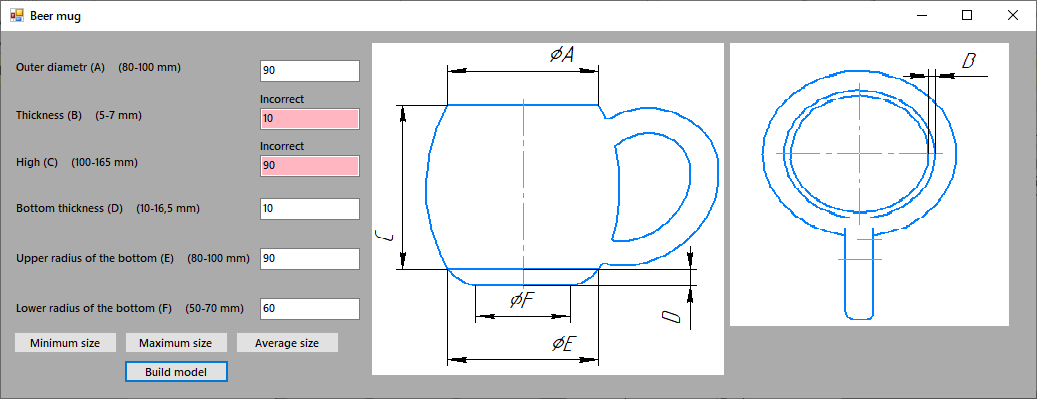


Рисунок 3.3 – Макет пользовательского интерфейса с некорректным вводом данных

В случае некорректного ввода ячейка со значениями изменит цвет и над ней появится надпись «Incorrect».

Список использованных источников

1. Компас (САПР) // <https://kompas.ru/solutions/developers/> (дата обращения: 14.10.2022).
2. Blender manual Add-ons // Blender 3.3 manual URL: <https://docs.blender.org/manual/en/latest/editors/preferences/addons.html> (дата обращения: 14.10.2022).
3. Archimesh // Blender 3.3 Manual URL: <https://docs.blender.org/manual/en/latest/addons/add_mesh/archimesh.html> (дата обращения: 14.10.2022).