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 "КРЫШКА" ДЛЯ САПР КОМПАС-3D V20

Проект системы по лабораторному проекту

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 588-2

\_\_\_\_\_\_\_\_\_Е.И. Белов

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_А.А. Калентьев

# Описание САПР

## Описание программы

САПР (Система автоматизированного проектирования) — автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из комплекса технических, программных и других средств автоматизации его деятельности. [1]

Компас — семейство систем автоматизированного проектирования, универсальная система автоматизированного проектирования, позволяющая в оперативном режиме выпускать чертежи изделий, схемы, спецификации, таблицы, инструкции, расчётно–пояснительные записки, технические условия, текстовые и прочие документы. Изначально система ориентирована на оформления документации в соответствии с ЕСКД, ЕСТД, СПДС и международными стандартами, но этим возможности системы не ограничиваются. [2]

## Описание API

API (англ. Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. [3]

Взаимодействие внешнего приложения или подключаемого модуля с системой КОМПАС осуществляется посредством программных интерфейсов, называемых API. В КОМПАС на данный момент существуют API двух версий: API 5 и API 7.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс можно с помощью экспортной функции CreateKompasObject(). Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы. В таблице 1.1 приведены методы интерфейса KompasObject.

Таблица 1.1 – Методы интерфейса KompasObject

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название | | Тип входных параметров | | Возвращаемое значение | | Описание |
| Document3D() | |  | | Указатель на интерфейс документа трехмерной модели ksDocument3D | | Даёт возможность получить указатель на интерфейс трёхмерного документа (детали или сборки) |
| ActiveDocument3D() | |  | | Указатель на интерфейс документа трехмерной модели ksDocument3D | | Дает возможность получить указатель на активный трехмерный документ |
| GetParamStruct() | | structType – тип интерфейса параметров | | Указатель на интерфейс указанного типа из StructType2D | | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| Visible |  | | bool | | Свойство видимости приложения | |
| GetMathematic2D() |  | | Указатель на интерфейс ksMathematic2D | | Метод для получения указателя на интерфейс для работы с математическими функциями | |

Графические документы имеют собственный интерфейс – ksDocument2D, со своими специфическими свойствами и методами. С помощью функций, присутствующих в ksDocument2D, создаются изображения в эскизах трехмерных операций. Свойства (члены данных) этого интерфейса позволяют динамически управлять настройками любого трехмерного документа системы из модуля. Наиболее используемые из них приведены в таблице 1.2.

Таблица 1.2 – Методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| Create () | invisible-признак режима редактирования документа (TRUE-невидимый режим, FALSE – видимый режим), typeDoc – тип документа | TRUE – в случае успешного завершения | Дает возможность создать пустой документ (деталь или сборку) |
| GetPart() | Type – тип компонента из перечисления | Указатель на интерфейс компонента ksPart или IPart | Позволяет получить указатель на интерфейс компонента в сборке |
| UpdateDocumentParam() |  | TRUE – в случае успешного завершения. | Позволяет обновить настройки документа |

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.

Таблица 1.3 – Методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| EntityCollection() | objType – тип объектов | В случае успеха указатель на интерфейс ksEntityCollection или IEntityCollection, в случае неудачи – NULL. | Формирует динамический массив трехмерных объектов и возвращает указатель на его интерфейс |
| GetPart() | type – тип компонента из перечисления | Указатель на интерфейс компонента ksPart или IPart | Получить указатель на интерфейс компонента |
| NewEntity() | objType – тип объекта. | Указатель на интерфейс ksEntity или IEntity. | Создает интерфейс нового трехмерного объекта и возвращает указатель на него |
| GetDefaultEntity() | objType – тип объекта. | Указатель на интерфейс ksEntity или IEntity. | Возвращает указатель на интерфейс объекта, создаваемого системой в трехмерном документе по умолчанию |

## Обзор аналогов

**Mechanical Desktop Power Pack**

Mechanical Desktop Power Pack – это мощная, динамично развивающаяся инженерная система автоматизации проектирования самых разнообразных объектов – от простейших деталей и узлов до сложных машиностроительных механизмов. [4]

Она позволяет:

* автоматизировать процесс разработки изделия;
* управлять взаимным расположением элементов конструкции с автоматическим обновлением модели и чертежа в процессе внесения в них изменений;
* работать с трёхмерными поверхностями и строить сложные модели с произвольными пространственными формами;
* создавать сложные параметрические твердотельные модели, формообразующие кривые, пространственные NURBES–поверхности и тонкие оболочки;
* генерировать плоские проекции, формировать чертежи изделия;
* управление размерами деталей и узлов и многое другое.

На рисунке 1.1 представлен пример создания крышки в программе Mechanical Desktop Power Pack.



Рисунок 1.1 – Пример интерфейса Mechanical Desktop Power Pack и готовой модели

# Описание предмета проектирования

Крышка – это деталь тело вращения. Конструкция детали представляет собой взаимное пересечение цилиндрических поверхностей. Крышка предназначена для обеспечения герметизации корпуса редуктора, предотвращения вытекания масла и попадания в корпус редуктора абразивных частиц, а также фиксации наружного кольца подшипника. Вследствие этого крышка должна обладать необходимой герметичностью и прочностью.

Основным предназначением торцевой крышки корпуса является фиксация подшипника в посадочном месте разъемного корпуса. [5]

Модель крышки представлена на рисунке 2.1.

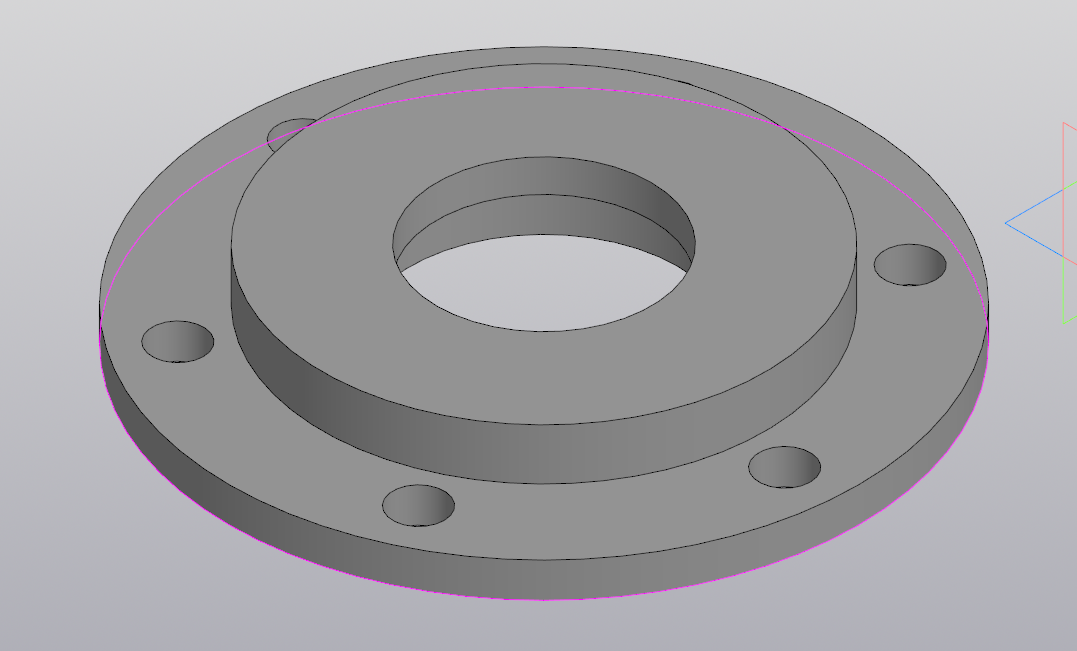


Рисунок 2.1 – Модель крышки

Параметры крышки:

* диаметр крышки A (рисунок 2.2): 50 мм – 500мм;
* диаметр малого ступенчатого отверстия крышки B (рисунок 2.2): 17 мм – 170 мм;
* диаметр большого ступенчатого отверстия крышки C (рисунок 2.2): 20 мм – 200 мм;
* диаметр малых отверстий D (рисунок 2.2): 2 мм – 40 мм;



Рисунок 2.2 – Чертёж крышки: вид снизу

* диаметр внешней ступени E (рисунок 2.3): 35 мм – 350 мм;



Рисунок 2.3 – Чертёж крышки: вид сверху

* толщина крышки F (рисунок 2.4): 6 мм – 68 мм;
* высота ступени крышки G (рисунок 2.4): 4 мм – 40 мм;
* высота внутренней ступени крышки H (рисунок 2.4): 5 мм – 50 мм;



Рисунок 2.4 – Чертёж крышки: вид сбоку

* расстояние между внешней ступень крышки (Е) малыми отверстиями (D) должно быть больше 5 мм;
* расстояние между малым ступенчатым отверстием (B) и большим ступенчатым отверстием (C) должно быть больше 5 мм;
* расстояние между малыми отверстиями (D) и диаметром крышки (A) должно быть больше 5 мм;
* расстояние между большим ступенчатым отверстием (C) и внешней ступени крышки (E) должно быть больше 15 мм;
* расстояние между высотой большого ступенчатого отверстия (E) и высотой крышки (F) должно быть больше 2 мм.

# Проект программы

## Описание технических и функциональных аспектов проекта

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML.

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML – моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот. [6]

При использовании UML были простроена диаграмма классов.

## Диаграмма классов

Унифицированный язык моделирования (UML) – это семейство графических нотаций, в основе которого лежит единая метамодель. Он помогает в описании и проектировании программных систем, в особенности систем, построенных с использованием объектно–ориентированных (ОО) технологий. [6]

На рисунке 3.1 представлена диаграмма классов.

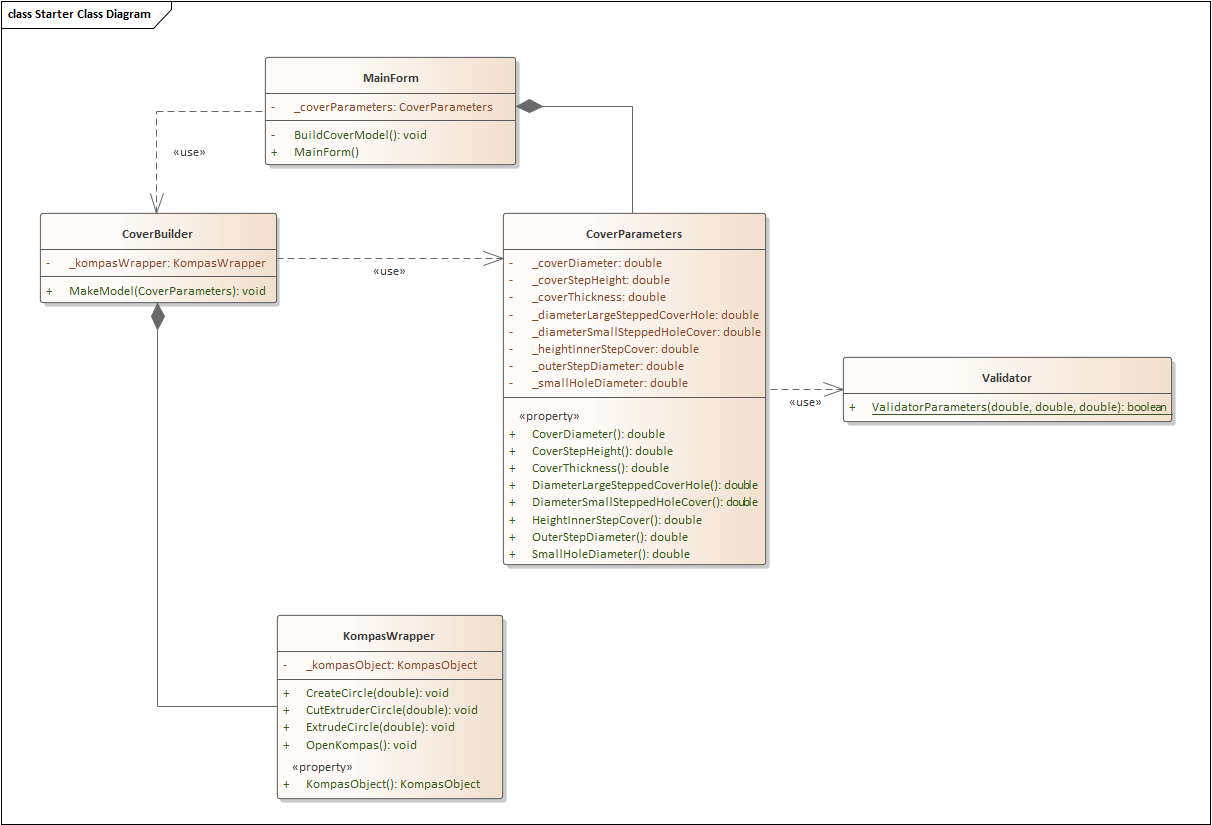


Рисунок 3.1 – Диаграмма классов

Класс MainForm содержит в себе экземпляр класса CoverParameters. CoverParameters хранит в себе параметры. CoverBuilder обладает методами для построения моделей. Класс KompasWrapper содержит в себе методы для работы с САПР Компас 3D. Класс Validator проверяет вводимые данные.

## Макет пользовательского интерфейса

Макет пользовательского интерфейса (рисунок 3.2) представляет собой форму для ввода параметров крышки. Построение модели осуществляется путем нажатия на кнопку «Build». При попытке ввода некорректных значений, в поле для ввода будет изменять свой цвет.

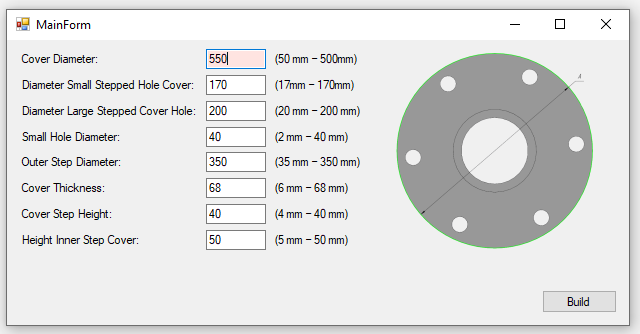


Рисунок 3.2 – Макет пользовательского интерфейса

После нажатия на кнопку «Build» при введенных некорректных значениях, появится окно, приведенное на рисунке 3.3.



Рисунок 3.3 – Окно ошибки

# Список использованной литературы

1. САПР [Электронный ресурс]. – Режим доступа: <https://www.autodesk.ru/solutions/cad-software> (дата обращения 17.11.2021);
2. Компас (САПР) [Электронный ресурс] ­– Режим доступа: <https://kompas.ru/kompas-3d-viewer/download/> (дата обращения 17.11.2021)
3. API – Библиотека обучающей и информационной литературы. [Электронный ресурс] – Режим доступа: <http://www.k2x2.info/kompyutery_i_internet/kompas_3d_v10_na_100/p9.php> (дата обращения 30.10.2021);
4. Кудрявцев Е.М. Mechanical Desktop Power Pack Основы работы в системе. Изд: ДМК Пресс, 2012 г., 535 с.: ил. (Серия «Проектирование»).
5. Разработка технологического процесса детали Крышка. [Электронный ресурс] – Режим доступа: <https://vmasshtabe.ru/mashinostroenie-i-mehanika/tm/razrabotka-tehnologicheskogo-protsessa-detali-kryishka.html> (дата обращения 30.10.2021);
6. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. Изд: Символ–Плюс, 2011, с.192   
   (3-е издание);