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 «Половник» ДЛЯ «КОМПАС 3D V20»**

Проект системы по лабораторному проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 588-3

\_\_\_\_\_\_\_\_\_Зорина Д.С.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП:

\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Томск 2022

**Оглавление**

[1. Описание САПР 3](#_Toc86356821)

[1.1. Описание программы 3](#_Toc86356822)

[1.2. Описание API 3](#_Toc86356823)

[1.3. Обзор аналогов 7](#_Toc86356824)

[2. Описание предмета проектирования 8](#_Toc86356825)

[3. Проект программы 9](#_Toc86356826)

[3.1 Макет пользовательского интерфейса 10](#_Toc86356827)

[Список литературы 13](#_Toc86356828)

# 1. **Описание САПР**

## **1.1. Описание программы**

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий и сотни тысяч профессиональных пользователей, благодаря сочетанию проектирования изделий любой сложности и простоты освоения. Особенностью продукта является использование российского геометрического ядра C3D и собственных программных технологий. КОМПАС-3D обеспечивает поддержку файлов наиболее распространённых CAD-систем (SolidWorks, Autodesk Inventor, Solid Edge, Creo, NX, Catia), что позволяет организовывать совместную работу со смежными организациями и заказчиками, использующими другие программные продукты.[1]

## 1.2. Описание API

В КОМПАС-3D на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда, очевидно, что обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) 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 (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа). [2]

Ниже в таблице 1.1 представлены свойства и методы интерфейса KompasObject, которые будут использованы при разработке плагина.

Таблица 1.1 – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трёхмерного документа (детали или сборки) |
| GetDynamicArray(long type) | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| GetParamStruct (short structType) | StructType2D | Метод для получения указателя на интерфейс структуры параметров объекта нужного типа |
| visible | bool | Cвойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksDocument3D, которые будут использованы при разработке плагина

Таблица 1.2 – Некоторые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool typeDoc) | bool | Метод для создания документа-модели (детали или сборки) |
| UpdateDocumentParam() | bool | Метод для активизации изменённых параметров продукта |
| GetPart(int type) | ksPart | Метод для получения указателя на интерфейс компонента в соответствии с заданным типом |

В таблице 1.3 представлены методы интерфейса ksPart, которые будут использованы при разработке плагина

Таблица 1.3 – Некоторые методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection (short objType) | ksEntityCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity (short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity (short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.4 представлены методы интерфейса ksEntity, которые будут использованы при разработке плагина

Таблица 1.4 – Некоторые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnknown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблицах 1.5 и 1.6 представлены методы и свойства интерфейса ksPlaneOffsetDefinition, которые будут использованы при разработке плагина

Таблица 1.5 – Свойства интерфейса ksPlaneOffsetDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| direction | bool | Направление смещения от базовой плоскости |
| offset | double | Смещение (расстояние) от базовой плоскости |

Таблица 1.6 – Методы интерфейса ksPlaneOffsetDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных данных | Тип выходных данных | Описание |
| SetPlane | plane - указатель на интерфейс плоскости ksEntity или IEntity. | TRUE – в случае успешного завершения | Изменить указатель на интерфейс базовой плоскости |
| GetPlane | - | ksEntity | Получить указатель на интерфейс базовой плоскости |

## 1.3. Обзор аналогов

**Model ChemLab**

Model ChemLab - это продукт, включающий в себя интерактивное моделирование и рабочее пространство лабораторной записной книжки с отдельными областями для теории, процедур и наблюдений студентов. Обычно используемое лабораторное оборудование используется для моделирования. [3]

Интерфейс приложения представлен на рисунке 1.1



Рисунок 1.1 – Интерфейс приложения Model ChemLab

# 2. Описание предмета проектирования

Предметом проектирования является модель половника. Данная модель имеет 6 основных параметров

1. Внешний диаметр = 50 мм ≤ А ≤ 150 мм
2. Внутренний диаметр = 30 мм ≤B≤ A – 4 мм
3. Ширина ручки = 2 мм ≤ С ≤ мм
4. Толщина ручки = 1 мм ≤ D ≤ (A – B) / 2 мм
5. Длина ручки = 5 мм ≤ E ≤ 200 мм
6. Радиус скругления ручки = 1 мм ≤ F ≤ С / 2 мм

На рисунке 2.1 представлен чертёж половника.

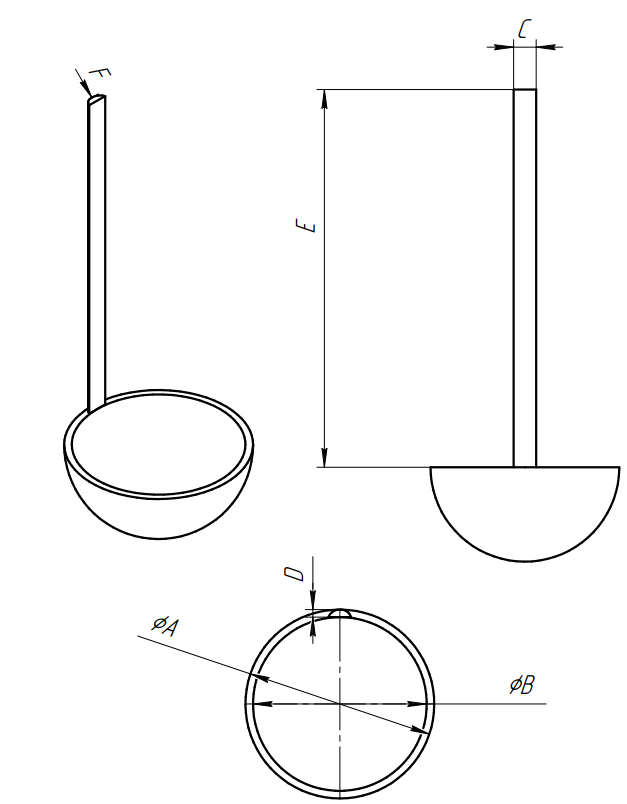


Рисунок 2.1 – Чертёж половника.

# 3. Проект программы

## 3.1 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами.[4] Целью создания диаграммы классов является графическое представление статической структуры элементов системы. Диаграмма классов плагина представлена на рисунке 3.1.

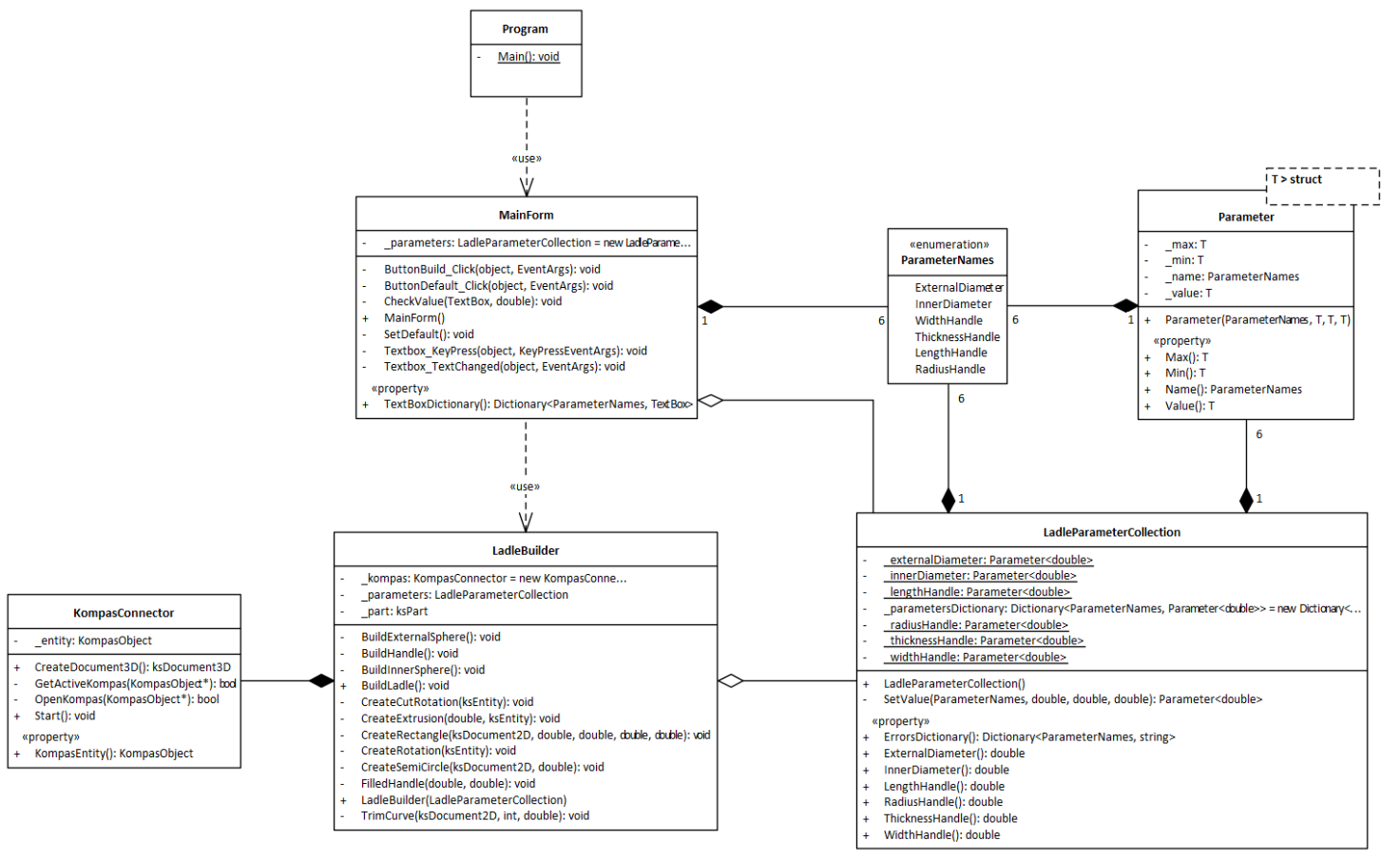


Рисунок 3.1 – Диаграмма классов плагина «Половник»

* MainForm – является главным элементом управления для обработки действий в графическом интерфейсе, в классе находятся метод для демонстрации корректного и некорректного ввода в текстовые поля (CheckValue()), также обработчики ввода с клавиатуры (Textbox\_KeyPress, Textbox\_TextChanged);
* LadleParametersCollection – содержит параметры половника для построения, которые проверяются на правильность с помощью свойств использующих методы шаблонного класса;
* LadleBuilder – выполняет построение детали, в данном классе находятся поля и методы, которые будут использовать API КОМПАС-3D, для построения основных примитивов(CreateSemiCircle(), CreateRectangle() и т.д.), с помощью которых будет строиться деталь в методе BuildLadle();
* KompasConnector – класс связи с КОМПАС – 3D, который будет вызывать метод Start() при каждом построении новой детали. В данном методе сначала будет происходить вызов метода GetActiveKompas() для получения экземпляра объекта КОМПАС – 3D из таблицы всех запущенных объектов. В случае провала, будет вызываться метод OpenKompas(), который будет открывать КОМПАС – 3D. Указатели в данных методах нужены, чтобы не запускать каждый раз новый КОМПАС – 3D, а активировать уже открытый. Метод CreateDocument3D() будет запускать окно создания 3D-модели для построения в нём детали;
* ParameterNames – перечисление имён параметров половника, для предотвращения ошибок и удобной валидации;
* Parameter – данный шаблонный класс будет создаваться для каждого параметра, нужен для хранения в себе основной информации о параметре, при этом в методе Max() максимальное значение будет сверяться с минимальным, чтобы точно быть больше него, а метод Value() будет проверять введенное значение на соответствие заданному диапазону, т.е. больше минимально возможного значения и меньше максимального возможного значения.

## 3.1 Макет пользовательского интерфейса

Плагин представляет собой пользовательскую форму с полями для ввода 6 геометрических параметров: «Внешний диаметр», «Внутренний диаметр», «Ширина ручки», «Толщина ручки», «Длина ручки», «Радиус скругления ручки». Ниже располагаются кнопки для запуска построения и возврата к стандартным параметрам. Справа находится изображение, демонстрирующее пример модели.

Макет пользовательского интерфейса представлен на рисунке 3.1.

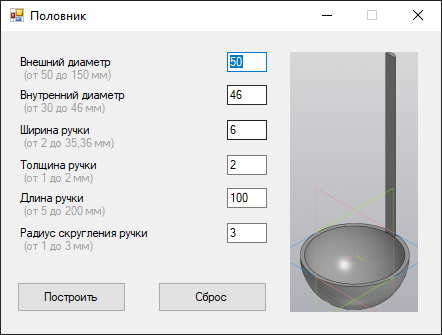


Рисунок 3.1 – Макет пользовательского интерфейса.

При вводе параметров, выходящих за диапазон, соответствующее поле подсвечивается красным цветом (рис. 3.2). При нажатии на кнопку «Построить запускается построение модели. При попытке запустить построение с неверными параметрами будет выведено соответствующее сообщение. Пример сообщения представлен на рисунке 3.3.

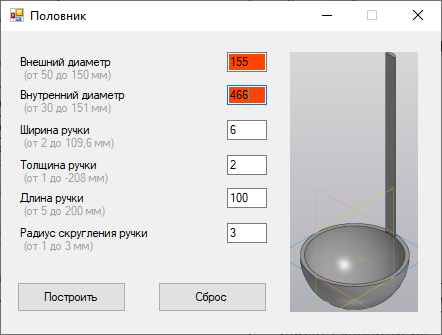


Рисунок 3.2 – Пример ввода неверных данных.

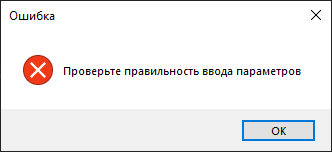


Рисунок 3.3 – Пример сообщения о некорректности данных.

# Список литературы

1. Система трёхмерного моделирования КОМПАС-3D [электронный ресурс]. – URL: <https://ascon.ru/products/7/review/> (дата обращения 26.10.2021).
2. Базовые интерфейсы API системы КОМПАС [электронный ресурс]. – URL: <https://it.wikireading.ru/23741> (дата обращения 26.10.2021).
3. Model ChemLab [электронный ресурс]. – URL: <https://www.modelscience.com/products.html> (дата обращения 27.10.2021).
4. Фаулер М. UML. Основы. 3-е издание / М.Фаулер. – 3-е изд., пер. с англ. – СПб.: Символ-Плюс, 2019. – 192 с. (дата обращения 09.11.2021).