Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «БУТЫЛКА»**

**ДЛЯ «Autodesk AutoCAD»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 580-2

\_\_\_\_\_\_\_\_\_ Сеченов В.В

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Томск, 2023

# 1 Описание САПР

## **1.1 Информация о выбранной САПР**

AutoCAD – это ведущее в отрасли коммерческое программное обеспечение для САПР.

AutoCAD используется AEC (Architecture, Engineering и Construction) для создания и оптимизации 2D- и 3D-проектов. AutoCAD – это широко используемая программа, которая может помочь вам составить проектную документацию, изучить дизайнерские идеи, визуализировать концепции с помощью фотореалистичных визуализаций и смоделировать, как дизайн работает в реальном мире.

AutoCAD был впервые выпущен в декабре 1982 года в качестве настольного приложения. В 2010 году AutoCAD был выпущен в виде мобильного и веб-приложения под названием AutoCAD 360 [1].

Данная САПР была выбрана, так как AutoCAD является одной из САПР, позволяющей создавать плагины с использованием языка C# и имеет обширную и исчерпывающую документацию к API.

## **1.2 Описание API**

API (Application Programming Interface) – это набор протоколов, которые позволяют различным программным компонентам взаимодействовать и передавать данные [2].

Для работы AutoCAD существует ObjectARX. Это большой набор библиотек, которые используются для создания приложений для AutoCAD. Готовые библиотеки с разработанным приложением затем подключаются в AutoCAD.

Ниже в таблицах представлено описание классов (методы, свойства, константы и поля), которые будут использоваться при создании приложения.

Таблица 1.1 – Используемые свойства класса Application

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| DocumentManager | DocumentManager | Свойство, предоставляющее доступ к документам |

Таблица 1.2 – Используемые методы класса Polyline2d

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Polyline2d | Poly2dType type, Point3dCollection vertices, double elevation, bool closed, double startWidth, double endWidth, DoubleCollection bulges | Polyline2d | Конструктор класса |

Таблица 1.3 – Используемые методы класса Point3d

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Point3d | double x, double y, double z | Point3d | Конструктор класса |

Таблица 1.4 – Используемые методы класса Circle

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Circle | Point3d center, Vector3d normal, double radius | Circle | Конструктор класса |

Таблица 1.5 – Используемые методы класса Region

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| CreateFromCurves | DBObjectCollection | Region[] | Создает набор объектов Region из замкнутых контуров |

Таблица 1.6 – Используемые свойства класса Region

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Normal | Vector3d | Свойство, возвращающее координаты вектора нормали объекта Region |

Таблица 1.7 – Используемые методы класса DBObjectCollection

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| DBObjectCollection | - | DBObjectCollection | Конструктор класса |
| Add | DBObject value | void | Добавить объект в коллекцию |
| Clear | - | void | Удаляет все содержимое коллекции |

Таблица 1.8 – Используемые методы класса Point3dCollection

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Point3dCollection | Point3d[] values | Point3dCollection | Конструктор класса |

Таблица 1.9 – Используемые методы класса Solid3d

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Solid3d | - | Solid3d | Конструктор класса |
| Extrude | Region region,  double height,  double taperAngle | void | Создает сплошную область путем выдавливания, на расстояние height с углом сужения taperAngle. |
| BooleanOperation | BooleanOperationType operation, Solid3d solid | void | Выполняет логическую операцию между твердыми телами |
| ShellBody | SubentityId[] subentityIds, double offsetDistance | void | Преобразует твердое тело в тонкостенную твердую оболочку со смещением граней наружу от твердого тела при заданных положительных расстояниях смещения и внутрь при заданных отрицательных расстояниях. |

Таблица 1.10 – Используемые методы класса Transaction

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| AddNewlyCreatedDBObject | DBObject object, bool add | void | Помещает или удаляет из транзакции новый объект, который уже добавлен в базу данных |
| Commit | - | void | Фиксирует изменения, внесенные во все DBObjects, открытые во время транзакции, а затем закрывает их. |
| GetObject | ObjectId id, OpenMode mode, bool openErased, bool | DBObject | Вызывает функцию Open() верхней транзакции, передавая все полученные ею аргументы. |

Таблица 1.11 – Используемые методы класса BlockTableRecord

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| AppendEntity | Entity entity | ObjectId | Добавляет объект в базу данных и запись таблицы блоков. |

Таблица 1.12 – Используемые методы класса DoubleCollection

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| DoubleCollection | double[] values | DoubleCollection | Конструктор класса |

Таблица 1.13 – Используемые методы класса SymbolUtilityServices

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| GetBlockModel  SpaceId | Database databasePointer | ObjectId | Возвращает ObjectId предопределенной записи таблицы символов для указанной базы данных. |

## **1.3 Обзор аналогов плагина**

Аналогом является плагин для создания бутылки Wine Bottle для программы для 3D дизайна и архитектурного проектирования SketchUp [3].



Рисунок 1.1 – Интерфейс плагина Wine Bottle

# 2 Описание предмета проектирования

Бутылка – ёмкость для долговременного хранения жидкостей, высокий сосуд преимущественно цилиндрической формы и с узким горлом, удобным для закупоривания пробкой.

На рисунке 2.1 представлен чертеж модели бутылки с параметрами из технического задания.



Рисунок 2.1 – Чертеж модели бутылки

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

* длина основной части L (10 – 250 мм);
* ширина основной части W (10 – 250 мм);
* высота основной части H1 (10 – 250 мм);
* высота горлышка H2 (10 – 40 мм);
* радиус горлышка R (5 – 20 мм).

***Зависимости изменяемых параметров***:

* высота горлышка H2 от высоты основной части H1:
* радиус горлышка R от ширины W:
* радиус горлышка R от длины L:

# 3 Проект системы

## **3.1 Диаграмма классов**

Диаграмма классов (class diagram) показывает набор классов, интерфейсов и коопераций, а также их связи. Диаграммы этого вида чаще всего используются для моделирования объектно-ориентированных систем. Предназначены для статического представления системы. Диаграммы классов, включающие активные классы, представляют статическое представление процессов системы [4].

На рисунке 3.1 представлена диаграмма классов плагина.

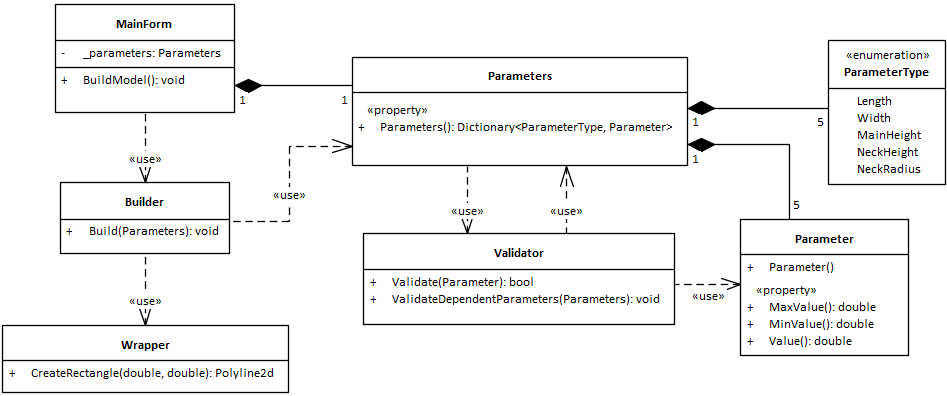


Рисунок 3.1 – Диаграмма классов плагина

## **3.2 Макеты пользовательского интерфейса**

На рисунках 3.2-3.5 представлен пользовательский интерфейс

с разными вариантами введенных данных.



Рисунок 3.2 – Макет пользовательского интерфейса



Рисунок 3.3 – Интерфейс с неправильно введенными параметрами



Рисунок 3.4 – Интерфейс при попытке построения фигуры с неправильно введенными независимыми параметрами



Рисунок 3.5 – Интерфейс при попытке построения фигуры с неправильно введенными зависимыми параметрами

# 4 Список использованных источников

1. Introduction to AutoCAD [Электронный ресурс]: сайт Iowa State Unifersity. URL: <https://iastate.pressbooks.pub/visualgraphiccomm/chapter/chapter-1/> (дата обращения: 21.10.2023).
2. What is API? [Электронный ресурс]: официальный сайт Postman. URL: [https://www.postman.com/what-is-an-api/](https://vk.com/away.php?to=https%3A%2F%2Fwww.postman.com%2Fwhat-is-an-api%2F&cc_key=) (дата обращения: 21.10.2023).
3. Wine Bottle [Электронный ресурс]: официальный сайт разработчика. URL: [http://www.nxtrender.com/wk/Wine\_Bottle.htm](https://vk.com/away.php?to=http%3A%2F%2Fwww.nxtrender.com%2Fwk%2FWine_Bottle.htm&cc_key=) (дата обращения: 21.10.2023).
4. Буч, Г. Язык UML. Руководство пользователя. 2-е изд. [Текст]/Г. Буч, Д. Рамбо, И. Якобсон. – М.: ДМК Пресс, 2006. – 496 с.