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)

## ПРОЕКТ СИСТЕМЫ

## на разработку плагина моделирования держателя для крепления трубы к стене для системы «AutoCAD»

Выполнил:

Студент группы 588-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Гирн П.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Оглавление**

[1 Описание САПР 3](#_Toc98861341)

[1.1 Описание программы 3](#_Toc98861342)

[1.2 Описание API 3](#_Toc98861343)

[2 Описание предмета проектирования 10](#_Toc98861344)

[3 Проект программы 11](#_Toc98861345)

[3.1 Диаграмма классов 11](#_Toc98861346)

[3.2 Макет пользовательского интерфейса 15](#_Toc98861347)

[Список литературы 17](#_Toc98861348)

## 1 Описание САПР

## 1.1 Описание программы

Система автоматизированного проектирования (САПР) или CAD (англ. Computer-Aided Design) – программный пакет, предназначенный для создания чертежей, конструкторской и/или технологической документации и 3D моделей. [1]

AutoCAD — это программное обеспечение автоматизированного проектирования (САПР), с помощью которого архитекторы, инженеры и строители создают точные 2D- и 3D-чертежи. Первая версия системы была выпущена в 1982 году. AutoCAD и специализированные приложения на его основе нашли широкое применение в машиностроении, строительстве, архитектуре и других отраслях промышленности.

AutoCAD включает в себя полный набор инструментов для комплексного трёхмерного моделирования (поддерживается твердотельное, поверхностное и полигональное моделирование). AutoCAD позволяет получить высококачественную визуализацию моделей с помощью системы рендеринга mental ray. Также в программе реализовано управление трёхмерной печатью (результат моделирования можно отправить на 3D-принтер) и поддержка облаков точек (позволяет работать с результатами 3D-сканирования) [2].

## 1.2 Описание API

Среда программирования ObjectARX используется для адаптации и расширения функциональных возможностей AutoCAD и продуктов на его основе. Она обеспечивает непосредственный доступ к структурам базы данных AutoCAD, графической системе и определениям встроенных команд.

В состав ObjectARX SDK входит также управляемый API, который часто называют AutoCAD .NET API. Для адаптации и расширения функциональных возможностей AutoCAD и продуктов на его основе может применяться любой язык программирования, поддерживающий .NET. Обеспечивается непосредственный доступ к структурам базы данных AutoCAD, определениям встроенных команд и другим внутренним программным элементам [3].

Все графические объекты, которые отображаются на экране, унаследованы от класса Entity. Класс Entity наследуется от класса DBObject, который является базовым классом для всех объектов, находящихся в базе данных Database (таблица 1.4).

Каждая сущность в базе данных имеет свой уникальный идентификатор, который представлен классом ObjectId.

В таблице 1.1 представлен класс Application, предоставляющий доступ к объекту программы.

Таблица 1.1 – Методы и свойства класса Application.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| DocumentManager | DocumentManager | Свойство, хранящее менеджер документов. |
| ShowModalWindow(Window) | bool? | Метод для открытия окна |

В таблице 1.2 представлен класс DocumentManager, предоставляющий доступ к текущему документу.

Таблица 1.2 – Методы и свойства класса DocumentManager.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| MdiActiveDocument | Document | Свойство, хранящее открытый документ. |

В таблице 1.3 представлен класс Document, который хранит сведения об открытом документе AutoCAD.

Таблица 1.3 – Методы и свойства класса Document.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| Database | Database | Свойство, хранящее базу данных текущего документа. |

В таблице 1.4 представлен класс Database, который позволяет работать с базами данных.

Таблица 1.4 – Методы и свойства класса Database.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| BlockTableId | ObjectId | Свойство, хранящее идентификатор в базе, соответствующий BlockTable |
| TransactionManager | TransactionManager | Свойство, которое предоставляет доступ к TransactionManager для текущей базы данных |

В таблице 1.5 представлен класс TransactionManager, который позволяет работать с транзакциями.

Таблица 1.5 – Методы и свойства класса TransactionManager.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| StartTransaction() | Transaction | Начало транзакции |

В таблице 1.6 представлен класс Transaction, который используется для чтений данных из баз, и сохранения изменений.

Таблица 1.6 – Методы и свойства класса Transaction.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| GetObject(ObjectId id,  OpenMode mode) | DBObject | Получение объекта по его идентификатору. OpenMode объявляет права доступа к объекту (для чтения, записи, уведомления) |
| AddNewlyCreatedDBObject  (DBObject obj, bool add) | void | Добавление нового объекта в транзакцию. Нужно для того, чтобы данные об объекте сохранились |

Продолжение таблицы 1.6.

|  |  |  |
| --- | --- | --- |
| Commit() | void | Сохранение данных, отправление коммита. |

В таблице 1.7 представлен класс BlockTableRecord, который используется для записи данных в таблицу блоков.

Таблица 1.7 – Методы и свойства класса BlockTableRecord.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| AppendEntity(Entity entity) | ObjectId | Добавление сущности в таблицу |

В таблице 1.8 представлен класс Solid3d, который используется для работы с трехмерными фигурами.

Таблица 1.8 – Методы и свойства класса Solid3d.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| CreateFrustum(double height, double radiusAlongX, double radiusAlongY, double topRadius) | void | Создание цилиндра |
| Extrude(Region region, double height, double taperAngle) | void | Выдавливание 3d объекта из 2d рисунка |
| BooleanOperation(BooleanOperationType operation, Solid3d solid) | void | Выполнение булевых операций между двумя 3d фигурами. К булевым операциям относятся объединение, вычитание и пересечение 3d фигур. |
| CheckInterference(Solid3d otherSolid) | bool | Проверяет, пересекаются ли фигуры. |

В таблице 1.9 представлен класс Circle, который используется для работы с кругом.

Таблица 1.9 – Методы и свойства класса Circle.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| Center | Point3d | Центр круга |
| Dispose() | void | Освобождение ресурсов |

В таблице 1.10 представлен класс DBObjectCollection, представляющие коллекцию объектов DBObject.

Таблица 1.10 – Методы и свойства класса DBObjectCollection.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| Add(DBObject) | int | Добавление объекта DBObject в коллекцию |

В таблице 1.11 представлен класс Region, который используется для работы с двумерной областью.

Таблица 1.11 – Методы и свойства класса Region.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| Area | double | Площадь поверхности региона |
| Dispose() | void | Освобождение ресурсов |
| BooleanOperation(  BooleanOperationType operation, Region otherRegion) | void | Выполнение булевых операций между двумя регионами. К булевым операциям относятся объединение, вычитание и пересечение областей |
| CreateFromCurves(  DBObjectCollection curveSegmants) | DBObjectCollection | Создание области из каждого замкнутого цикла, образованного входным массивом объектов. |

В таблице 1.12 представлен класс Polyline, который используется для работы ломаной линией.

Таблица 1.11 – Методы и свойства класса Polyline.

|  |  |  |
| --- | --- | --- |
| **Название метода** | **Тип** | **Описание** |
| AddVertexAt(int index,  Point2d pt, double bulge, double startWidth, double endWidth) | void | Добавление вершин |

## 1.3 Обзор аналогов

Одним из аналогов является библиотека стандартных изделий. Библиотека стандартных изделий – библиотека трехмерных моделей стандартных изделий для вставки в сборку.

Ключевые возможности:

* Удобный поиск элементов по наименованиям и значениям атрибутов;
* Наглядное представление элементов при помощи трехмерных моделей;
* Быстрый доступ к содержанию, размещенному на тематических вкладках;
* Формирование индивидуальных списков избранных элементов;
* Создание типовых крепежных соединений с автоматическим подбором размеров элементов;
* Экспорт графических представлений в файлы различных графических форматов.

Модульная структура приложения позволяет пользователю самостоятельно определить требуемый для своих нужд перечень стандартных изделий.

Приложение способно построить крепёжные изделия по стандартам ГОСТ: болты, винты, гайки, шайбы, шпильки, заклепки и прочие крепежные изделия. [4]

На рисунке 1.1 изображено окно библиотеки «Стандартные изделия».



Рисунок 1.1 – Окно библиотеки «Стандартные Изделия»

## 2 Описание предмета проектирования

Предметом проектирования является модель держателя для крепления трубы к стене. Данная модель имеет 5 основных параметров:

1. Высота H должна быть от 15 мм до 40 мм;
2. Диаметр D должен быть не меньше 20 мм и не больше D2 / 2 мм;
3. Диаметр D1 должен быть не меньше 10 мм и не больше D-2.5 мм;
4. Диаметр D2 должен быть не меньше, чем 60 мм и не больше 90 мм;
5. Диаметр D3 должен быть не меньше 4 мм и не больше, чем (D2 – D)/4 мм.

Чертеж модели показан на рисунке 2.1.

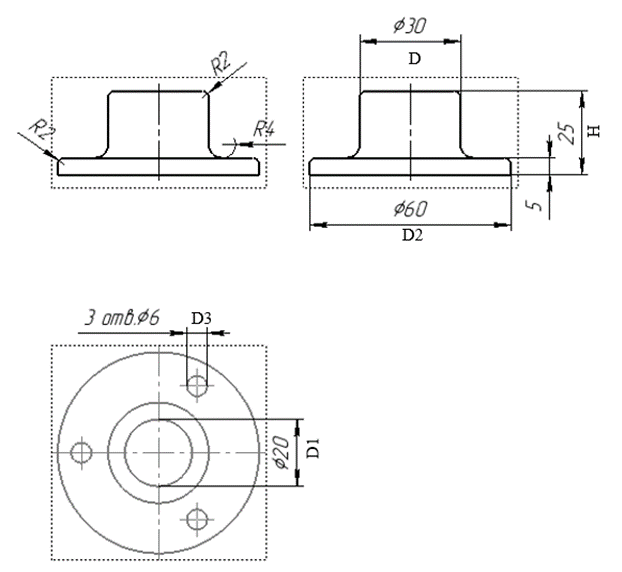


Рисунок 2.1 – Чертеж модели

## 3 Проект программы

## 3.1 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними [5].

Диаграмма классов представлена на рисунке 3.1.

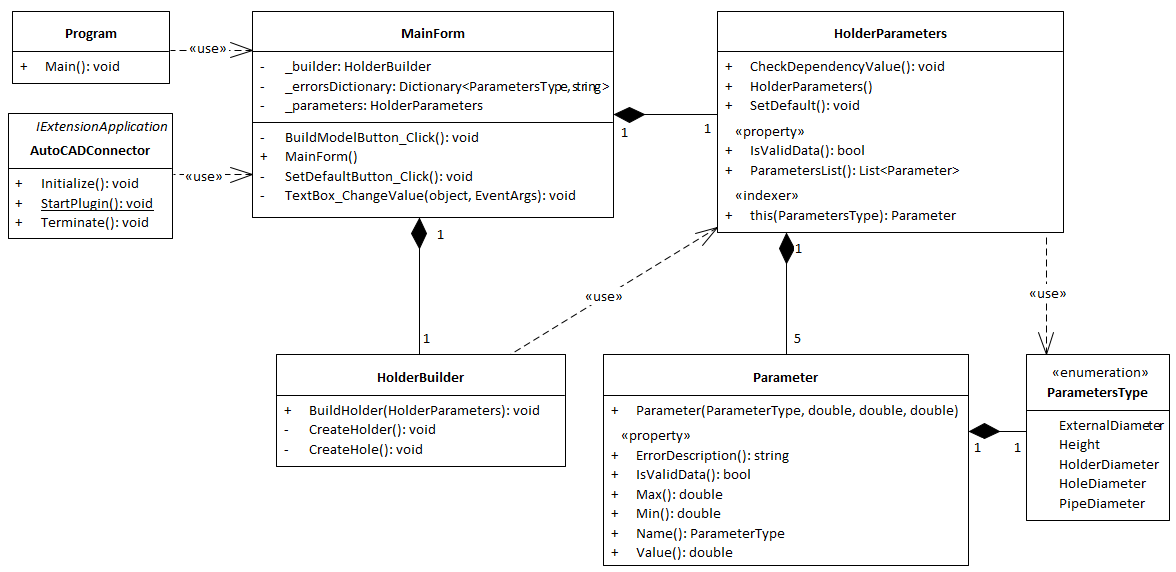


Рисунок 3.1 – UML-диаграмма классов

Класс «Program», использует «MainForm» для обработки действий в графическом интерфейсе. Класс «HolderBuilder» содержит в себе методы создания 3D модели в «AutoCAD». Класс «HolderParameters» хранит значения, введенные в графическом интерфейсе. Класс «AutoCADConnector» производит запуск программы «AutoCAD» и строит объект в этой программы.

Класс AutoCADConnector реализует интерфейс IExtensionApplication, необходимый для запуска плагина в программе AutoCAD.

Класс Parameter хранит данные о каждом параметре модели из перечисления ParametersType. Параметр хранит минимальное и максимальное значение параметра, а также текущее значение.

Класс HolderParameters хранит список параметров, а также методы обработки зависимых параметров.

Для построения модели используется класс HolderBuilder. Метод Build строит 3D объект на основании списка параметров.

Свойства и методы данных классов представлены в таблицах 3.1 – 3.5.

Таблица 3.1 – Класс HolderBuilder.

|  |  |  |
| --- | --- | --- |
| Название | Тип входных параметров | Описание |
| BuildHolder  (HolderParameters) | HolderParameters – параметры модели | Метод построения модели по заданным параметрам |

Таблица 3.2 – Класс AutoCADConnector.

|  |  |
| --- | --- |
| Название | Описание |
| Initialize() | Методы интерфейса IExtensionApplication, без реализации |
| Terminate() |
| StartPlugin() | Команда, которая вызывается в AutoCAD для запуска плагина |

Таблица 3.3 – Класс Parameter.

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Name | ParameterType | Название параметра |
| Min | double | Минимальное значение параметра |
| Max | double | Максимальное значение параметра |
| Value | double | Текущее значение параметра. При изменении свойства происходит валидация значения. Если значение введено некорректно, флаг IsValidData устанавливается в false и выбрасывается исключение, содержащее описание ошибки. |

Продолжение таблицы 3.3.

|  |  |  |
| --- | --- | --- |
| IsValidData | bool | Флаг, сообщающий о корректности введенных данных. |
| ErrorDescription | string | Свойство, хранящее описание ошибки |

Таблица 3.4 – Класс HolderParameters.

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| ParametersList | List<Parameter> | Свойство, хранящее список параметров |
| this(ParameterType) | Parameter | Индексатор, позволяющий получить параметр по его названию |
| SetDefault() | void | Метод для задания значений по умолчанию |
| CheckDependency  Value() | void | Метод для изменения ограничений у зависимых параметров |
| IsValidData | bool | Флаг, сообщающий о корректности введенных данных. Если хотя бы у одного параметров IsValidData установлен в false, текущий флаг тоже устанавливается в false. |

Таблица 3.5 – Класс MainForm.

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| \_builder | HolderBuilder | Экземпляр класса, используемого для построения модели. |
| \_parameters | HolderParameters | Список параметров модели. |
| \_errorsDictionary | Dictionary  <ParametersType,  string> | Словарь, хранящий сообщение об ошибке для каждого параметра |
| SetDefault  Button\_Click  (object, EventArgs) | void | Обработчик события нажатия кнопки «Установить значения по умолчанию». Вызывается метод SetDefault() для объекта \_parameters. |
| BuildButton\_Click  (object, EventArgs) | void | Обработчик события нажатия кнопки «Построить». Вызывается метод Build() для объекта \_builder. |
| TextBox\_ChangeValue  (object, EventArgs) | void | Обработчик события ввода значения в текстбокс. Устанавливается свойство Value для соответствующего параметра. В случае ошибки, блокируется кнопка «Построить», меняется цвет текстбокса. |

## 3.2 Макет пользовательского интерфейса

Макет пользовательского интерфейса создан с помощью Windows Form. Перед пользователем представлены поля для ввода параметров (в мм) детали с указанием ограничений для каждого параметра. Также присутствуют кнопка для построения модели, и кнопка для установки параметров по умолчанию. В правой части представлен чертеж детали с обозначением всех параметров.

На рисунке 3.2 представлен макет пользовательского интерфейса.

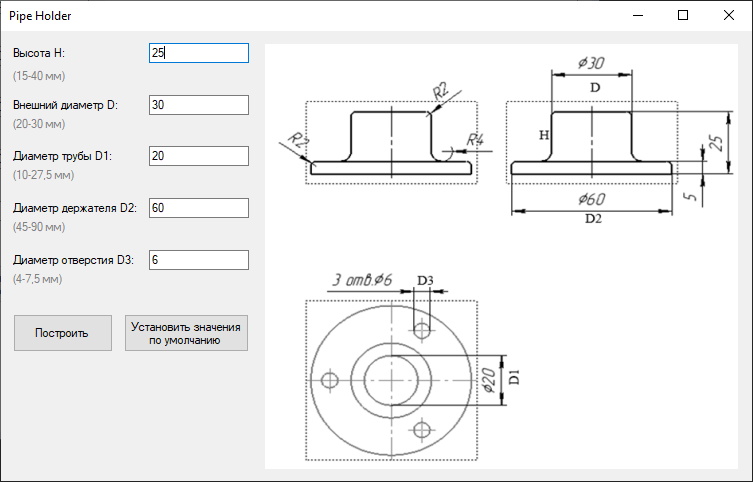


Рисунок 3.2 — Макет пользовательского интерфейса

Проверка правильности ввода значений проводится по ходу заполнения полей. Если поле заполнено неправильно, то оно выделяется красным цветом, во всплывающей подсказке отображается сообщение об ошибке и блокируется кнопка для построения (рисунок 3.3).

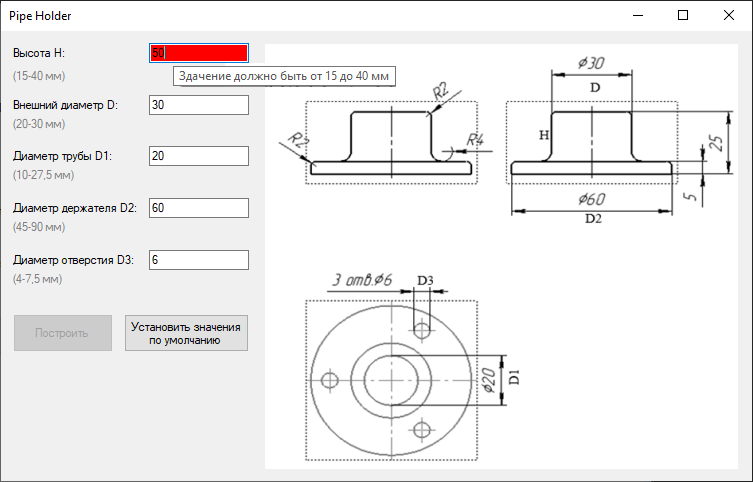


Рисунок 3.4 – Подсказка об ошибке

## Список литературы

1. Общие сведения о САПР [Электронный ресурс]. – Режим доступа: https://dic.academic.ru/dic.nsf/ruwiki/1132874, свободный (дата обращения: 25.11.2021).

2. AutoCAD – Википедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/AutoCAD (дата обращения: 22.03.2022).

3. Разработка приложений для AutoCAD [Электронный ресурс]. URL: https://www.autodesk.ru/autodesk-developer-network/software-platform-russian/develop-autocad (дата обращения: 22.03.2022).

4. Стандартные изделия для КОМПАС [Электронный ресурс]. – Режим доступа: https://ascon.ru/products/17/review/, свободный (дата обращения: 12.11.2021).

5. Мартин Фаулер. UML. Основы. / Мартин Фаулер; пер. с англ. А. Петухова – 3-е издание. – Спб: Символ-Плюс, 2004 – 192с.