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 «ПИСЬМЕННЫЙ СТОЛ» ДЛЯ САПР AUTOCAD 2022**

Проект системы  
по дисциплине «Основы разработки САПР»

Студент гр. 589-1

\_\_\_\_\_\_\_\_\_ Д.С. Ахроменко

« 23 » сентября 2022

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ А.А. Калентьев  
« 23 » сентября 2022

Томск 2022

Оглавление

[Введение 3](#_Toc114561738)

[1 ОПИСАНИЕ САПР 5](#_Toc114561739)

[1.1 Описание программы 5](#_Toc114561740)

[1.2 Описание API 6](#_Toc114561741)

[1.3 Обзор аналогов 13](#_Toc114561742)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 16](#_Toc114561743)

[3 ПРОЕКТ ПРОГРАММЫ 19](#_Toc114561744)

[3.1 Диаграмма классов 19](#_Toc114561745)

[3.2 Макеты пользовательского интерфейса 22](#_Toc114561746)

[Заключение 26](#_Toc114561747)

[Список использованных источников 27](#_Toc114561748)

# Введение

Автоматизация проектирования занимает особое место среди информационных технологий. Во-первых, автоматизация проектирования – это синтетическая дисциплина, составными частями которой являются многие другие современные информационные технологии: здесь используются персональные компьютеры и рабочие станции, вычислительные сети и телекоммуникационные технологии, методы вычислительной математики, статистики, математического программирования, дискретной математики, искусственного интеллекта и т.д. А во-вторых, практически любому инженеру-разработчику необходимо знание основ автоматизации проектирования и умение работать со средствами САПР, поскольку в настоящее время проектные подразделения, конструкторские бюро и офисы насыщены компьютерами [1].

На сегодняшний день проектирование осуществляется за счет применения систем автоматизированного проектирования (САПР), которые решают весь комплекс задач – от анализа технического задания до разработки полного объема конструкторской и технологической документации. Это достигается за счет объединения современных технических средств и математического обеспечения, параметры и характеристики которых выбираются с максимальным учетом особенностей задач проектно-конструкторского процесса [2].

САПР представляет собой крупные организационно-технические системы, состоящие из комплекса средств автоматизации проектирования, взаимосвязанного с проектными подразделениями конкретной организации. Основная цель создания САПР – повышение эффективности труда инженеров, включая:

* сокращение трудоемкости проектирования и планирования;
* сокращение сроков проектирования;
* сокращение себестоимости проектирования и изготовления, уменьшение затрат на эксплуатацию;
* повышение качества и технико-экономического уровня результатов проектирования;
* сокращение затрат на натурное моделирование и испытания.

Как показывает практический опыт, предприятия, ведущие разработку без САПР или лишь с малой степенью их использования, оказываются неконкурентоспособными как из-за больших материальных и временных затрат на проектирование, так и из-за невысокого качества проектов [1].

В каждой крупной САПР есть собственные средства для разработки, которые предоставляют разработчикам возможность расширить функциональность данной системы под нужды конкретного проекта. Данным средством является программный интерфейс приложения (Application Programming Interface, API) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой [3]. API упрощает процесс программирования при создании приложений, абстрагируя базовую реализацию и предоставляя только объекты или действия, необходимые разработчику.

Расширение функциональности в основном подразумевает разработку плагина или библиотеки на основе предоставляемого API. Плагин (англ.   
plug-in, от plug in – «подключать») – это независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и (или) использования ее   
возможностей [4].

В качестве САПР, которая предоставляет API и для которой поставлена задача разработки плагина, выбрана система Autodesk AutoCAD 2022.

# ОПИСАНИЕ САПР

## Описание программы

AutoCAD – это двух- и трехмерная система автоматизированного проектирования и черчения, разработанная американской компанией Autodesk [5]. 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. С помощью данной системы архитекторы, инженеры и строители создают точные 2D-чертежи и 3D-модели [6].

Ранние версии AutoCAD оперировали небольшим числом элементарных объектов, таким как круги, линии, дуги и текст, из которых составлялись более сложные объекты. Однако на современном этапе возможности AutoCAD весьма широки и намного превосходят первоначальные.

В области двумерного проектирования AutoCAD по-прежнему позволяет использовать элементарные графические примитивы для получения более сложных объектов. Кроме того, программа предоставляет обширные возможности работы со слоями и аннотативными объектами (размерами, текстом, обозначениями). Использование механизма внешних ссылок (XRef) позволяет разбивать чертеж на составные файлы, за которые ответственны различные разработчики, а динамические блоки расширяют возможности автоматизации 2D-проектирования обычным пользователем без использования программирования.

Также AutoCAD включает в себя полный набор инструментов для комплексного трехмерного моделирования: поддерживается твердотельное, поверхностное и полигональное моделирование. В программе реализовано управление трехмерной печатью и поддержка облаков точек (возможность работы с результатами 3D-сканирования) [5].

## Описание API

AutoCAD представляет собой гибкую платформу разработки специализированных приложений, предназначенных для автоматизированного проектирования. Открытая архитектура позволяет адаптировать AutoCAD под конкретные задачи пользователей. Образцами такой адаптации могут служить САПР на базе AutoCAD для различных отраслей проектирования, а также тысячи надстроек, разработанных участниками сети Autodesk Developer Network [7].

**ObjectARX SDK**

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.

**AutoCAD .NET API**

В состав ObjectARX SDK также входит 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 [7].

Основными строительными блоками AutoCAD .NET API являются объекты. Каждый объект представляет собой конкретную часть AutoCAD. В AutoCAD .NET API существует множество различных типов объектов. Некоторые из объектов, представленных в AutoCAD .NET API:

* графические объекты (линии, дуги, текст, размеры);
* параметры стиля, такие как слои, типы линий и размерные стили;
* организационные структуры (вид, видовой экран);
* чертеж и сама программа AutoCAD.

Объекты структурированы иерархически, корневым объектом является AutoCAD Application. Эту иерархическую структуру часто называют «объектной моделью» (Object Model).

На рисунке 1.1 показаны взаимоотношения объекта Application и сущностей, каждая из которых (Entity) находится в BlockTableRecord, например в пространстве модели – Model Space [8].



Рисунок 1.1 – Иерархия объектов в AutoCAD

Объект Application – корневой объект AutoCAD .NET API. Из объекта Application можно получить доступ к главному окну AutoCAD, а также к любому открытому чертежу. Поскольку есть доступ к чертежу, можно взаимодействовать и с его объектами (за работу с открытыми файлами чертежей отвечает объект Document).

Объект Document, который на самом деле является чертежом AutoCAD, – это часть объекта DocumentCollection. Document обеспечивает доступ к объекту базы данных Database, связанному с объектом Document.

Объект Database содержит все графические и большинство неграфических объектов AutoCAD. К графическим объектам относятся видимые объекты, которые составляют чертеж: линии, окружности, растровые изображения и т.д. Другими словами, пользователь может видеть такой объект на экране и манипулировать им. Неграфическими объектами являются невидимые (информационные) объекты, такие как слои, типы линий, размерные стили, таблицы стилей [8].

Принцип создания и работы плагина для AutoCAD:

* создание проекта «Библиотека классов» (Class Library);
* добавление ссылок на необходимые библиотеки AutoCAD .NET API (из ObjectARX SDK);
* написание программного кода плагина;
* загрузка созданного плагина в AutoCAD (запуск AutoCAD и выполнение команды NETLOAD);
* отладка плагина (при необходимости).[9]

При создании плагинов для AutoCAD основными используемыми пространствами имен являются:

* **Autodesk.AutoCAD.ApplicationServices** – позволяет получить доступ к приложению AutoCAD;
* **Autodesk.AutoCAD.EditorInput** – позволяет получить доступ к редактору (Editor) AutoCAD;
* **Autodesk.AutoCAD.DatabaseServices** – предоставляет доступ к базе данных и сущностям AutoCAD;
* **Autodesk.AutoCAD.Geometry** – группирует все, что относится к геометрии в AutoCAD;
* **Autodesk.AutoCAD.Runtime** – отвечает за регистрацию   
  команд [9, 10].

В таблицах 1.1 – 1.22 представлены необходимые свойства и методы классов и интерфейсов AutoCAD .NET API, которые будут использованы при разработке плагина «Письменный стол».

Таблица 1.1 – Свойства класса Application

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| DocumentManager | DocumentCollection | Свойство, предоставляющее доступ к объекту DocumentManager |
| MainWindow | Window | Свойство, предоставляющее доступ к главному окну AutoCAD |

Таблица 1.2 – Методы класса Application

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные аргументы | Возвращаемое значение | Описание |
| ShowModelessWindow | Form formToShow | void | Метод для отображения окна WPF, передаваемого в виде параметра formToShow, в немодальном режиме. Использование данного метода вместо стандартного метода Show() предотвращает неожиданное поведение программы |

Таблица 1.3 – Свойства класса DocumentCollection

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| MdiActiveDocument | Document | Свойство, предоставляющее доступ к активному документу с расширением MDI |

Таблица 1.4 – Свойства класса Document

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| DocumentManager | DocumentCollection | Свойство, предоставляющее доступ к объекту DocumentManager |
| MainWindow | Window | Свойство, предоставляющее доступ к главному окну AutoCAD |

Таблица 1.5 – Методы класса Document

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные аргументы | Возвращаемое значение | Описание |
| ShowModelessWindow | Form formToShow | void | Метод для отображения окна WPF, передаваемого в виде параметра formToShow, в немодальном режиме. Использование данного метода вместо стандартного метода Show() предотвращает неожиданное поведение программы |

Таблица 1.6 – Свойства класса Database

Таблица 1.7 – Методы класса Database

Таблица 1.8 – Свойства класса TransactionManager

Таблица 1.9 – Методы класса TransactionManager

Таблица 1.10 – Свойства класса Transaction

Таблица 1.11 – Методы класса Transaction

Таблица 1.12 – Свойства класса Database

Таблица 1.13 – Методы класса Database

Таблица 1.14 – Свойства класса BlockTableRecord

Таблица 1.15 – Методы класса BlockTableRecord

Таблица 1.16 – Свойства класса Solid3D

Таблица 1.17 – Методы класса Solid3D

Таблица 1.18 – Свойства класса Circle

Таблица 1.19 – Методы класса Circle

Таблица 1.20 – Свойства класса Region

Таблица 1.21 – Методы класса Region

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Application | | |
| DocumentManager | DocumentCollection | Свойство, предоставляющее доступ к объекту DocumentManager |
| MainWindow | Window | Свойство, предоставляющее доступ к главному окну AutoCAD |
| ShowModelessWindow (Form formToShow) | void | Метод для отображения окна WPF, передаваемого в виде параметра formToShow, в немодальном режиме. Использование данного метода вместо стандартного метода Show() предотвращает неожиданное поведение программы |
| DocumentCollection | | |
| MdiActiveDocument | Document | Свойство, предоставляющее доступ к активному документу с расширением MDI |
| Document | | |
| Database | Database | Свойство, возвращающее объект базы данных, который используется данным документом |
| Editor | Editor | Свойство, предоставляющее доступ к |

Продолжение таблицы 1.1

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Document | | |
|  |  | редактору (Editor), связанному с данным документом |
| Window | Window | Свойство, предоставляющее доступ к окну, в котором открыт данный документ |
| LockDocument() | DocumentLock | Метод, блокирующий документ. Используется для предотвращения конфликтов с другими запросами в базе данных при взаимодействии с AutoCAD через немодальное диалоговое окно |
| Database | | |
| BlockTableId | ObjectId | Свойство, возвращающее идентификатор объекта в таблице BlockTable базы данных |
| TransactionManager | TransactionManager | Свойство, предоставляющее доступ к TransactionManager для текущей базы данных |
| TransactionManager | | |
| StartTransaction() | Transaction | Метод, начинающий новую транзакцию |
| Transaction | | |
| AddNewlyCreatedObject (DBObject obj, bool add) | void | Метод, который добавляет объект в транзакцию или удаляет его из транзакции |
| GetObject(ObjectId id,  OpenMode mode) | DBObject | Метод для получения объекта, на данный момент открытого в транзакции, по его идентификатору id. Аргумент mode объявляет права доступа к объекту (чтение, запись, уведомление) |

Продолжение таблицы 1.1

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Transaction | | |
| Commit() | void | Метод, который фиксирует изменения, внесенные во все объекты DBObject, открытые в течение транзакции, после чего закрывает их |
| BlockTableRecord | | |
| AppendEntity(Entity entity) | ObjectId | Метод, добавляющий сущность entity в базу данных и в таблицу блоков |
| Solid3D | | |
| CreateBox(double legnthAlongX, double lengthAlongY, double lengthAlongZ) | void | Метод для создания прямоугольного твердотельного примитива с центром тяжести в начале координат WCS. Аргумент legnthAlongX представляет длину прямоугольного твердотельного примитива, аргумент lengthAlongY – его ширину, аргумент lengthAlongZ – его высоту |
| Extrude(Region region, double height, double taperAngle) | void | Метод, создающий твердое тело посредством выдавливания области region на заданное расстояние по высоте height с заданным углом конусности taperAngle |
| BooleanOperation( BooleanOperationType operation, Solid3d solid) | void | Метод для выполнения булевых операций между двумя 3D-объектами. Аргумент operation представляет тип булевой операции (объединение, пересечение, вычитание) |
| Circle | | |
| Center | Point3d | Свойство, предоставляющее доступ к центру окружности |

Окончание таблицы 1.1

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Circle | | |
| Diameter | void | Свойство, предоставляющее доступ к диаметру окружности |
| Dispose() | void | Метод для освобождения ресурсов объекта Circle |
| DBObjectCollection | | |
| Add(DBObject value) | int | Метод для добавления объекта value в данную коллекцию |
| Region | | |
| Dispose() | void | Метод для освобождения ресурсов объекта Region |
| CreateFromCurves(  DBObjectCollection curveSegments) | DBObjectCollection | Статический метод для создания набора объектов Region из замкнутых циклов, представленных кривыми, которые содержатся в коллекции curveSegments. Созданные объекты возвращаются в виде массива областей DBObjectCollection |
| AddVertexAt(int index,  Point2d pt, double bulge, double startWidth, double endWidth) | void | Метод для добавления вершин к полилинии. Аргумент index представляет индекс вершины (начиная с 0), перед которой необходимо вставить новую вершину. Аргумент pt представляет точку расположения вершины. Аргумент bulge представляет выпуклость вершины. Аргумент startWidth представляет начальную ширину вершины, аргумент endWidth – конечную ширину вершины |

## Обзор аналогов

**Auto-Furniture**

Плагин Auto-Furniture – инструмент от компании FurnitureSoft, который работает с базой данных параметрических мебельных блоков для САПР AutoCAD. Данная программа упрощает и ускоряет проектирование мебели (рисунок 1.2).



Рисунок 1.2 – Интерфейс программы Auto-Furniture

В Auto-Furniture можно выбрать блок, наиболее близкий к проектируемому мебельному изделию, и задать ему размеры. Все изменения немедленно отражаются в окне предварительного просмотра. После нажатия кнопки «Вставить» (Paste) выбранный блок с заданными размерами вставляется в то место документа AutoCAD, которое указывает пользователь. Существует опция предварительного просмотра объекта с разных ракурсов (сверху, справа, спереди) для более удобной работы с программой.

Кроме размеров, плагин позволяет изменять и некоторые другие параметры мебельных изделий: типы полок, выдвижных ящиков, стеллажей; количество ящиков, полок или дверей. Также в Auto-Furniture есть дополнительные настройки для таких предметов, как кухни и   
шкафы-купе [11].

Примеры мебельных изделий, созданных в AutoCAD с помощью данного плагина, приведены на рисунке 1.3.



Рисунок 1.3 – Примеры мебельных изделий, созданных с помощью   
Auto-Furniture в САПР AutoCAD

**SymbolDesigner**

SymbolDesigner – инструмент для создания параметрических объектов в AutoCAD, разработанный компанией CAD Division. Параметрические объекты САПР рассматриваются в качестве скриптов языка PPM, а геометрические объекты – в качестве результатов выполнения скриптов языка PPM (рисунок 1.4).



Рисунок 1.4 – Интерфейс программы SymbolDesigner

Язык PPM также предоставляет средства для создания различных аннотаций, сложных размеров и любых других 2D-объектов. Следовательно, параметрический объект можно определить в точных единицах измерения.

SymbolDesigner позволяет ускорить процесс создания параметрических объектов, так как предоставляет возможность найти и исправить любые проблемы при проектировании модели; пользователь может визуализировать свой эскиз и просмотреть окончательные результаты сценария; можно также автоматизировать создание геометрических объектов с помощью встроенных инструментов. Кроме того, SymbolDesigner позволяет изменять параметры и точки параметров. Таким образом, можно наблюдать поведение параметрического объекта САПР в реальном времени [12].

# ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Предметом проектирования является письменный стол.

Письменный стол [13] – стол для письменных работ и аналогичных занятий сидя. Под рабочей поверхностью письменного стола находятся выдвижные ящики, в которых обычно хранятся канцелярские принадлежности и документы.

Трехмерная модель письменного стола может применяться как для согласования его внешнего вида с заказчиком, так и для визуализации, изготовления рабочих чертежей и т. д.

К изменяемым параметрам модели относятся:

* + длина столешницы L1 (от 800 до 1200 мм);
  + ширина столешницы B (от 1/2 \* L1 до 750 мм);
  + высота столешницы H1 (от 30 до 40 мм);
  + тип ножек (с круглым или квадратным основанием);
  + высота ножек H2 (от 690 до 740 мм);
  + диаметр основания ножки D (от 50 до 70 мм), если выбрано круглое основание;
  + длина основания ножки A (от 50 до 70 мм), если выбрано квадратное основание;
  + количество ящиков для канцелярии N (от 3 до 5 шт);
  + длина ящиков для канцелярии L2 (от 250 до L1 \* 1/3 мм).

Параметры плагина имеют следующие зависимости:

* минимальное значение параметра B должно быть не меньше 1/2 значения параметра L1;
* выбор типа ножек определяет форму и размер их основания (для круглых ножек – диаметр, для квадратных ножек – длина);
* максимальное значение параметра L2 не должно превышать 1/3 значения параметра L1.

На рисунках 2.1 – 2.2 представлена 3D-модель письменного стола.



Рисунок 2.1 – 3D-модель письменного стола с круглыми основаниями ножек



Рисунок 2.2 – 3D-модель письменного стола с квадратными основаниями ножек

Чертеж моделируемого объекта приведен на рисунке 2.3.





Рисунок 2.3 – Чертеж письменного стола

# ПРОЕКТ ПРОГРАММЫ

## Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами [14]. Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы.

Диаграмма классов плагина «Письменный стол» представлена на рисунке 3.1.

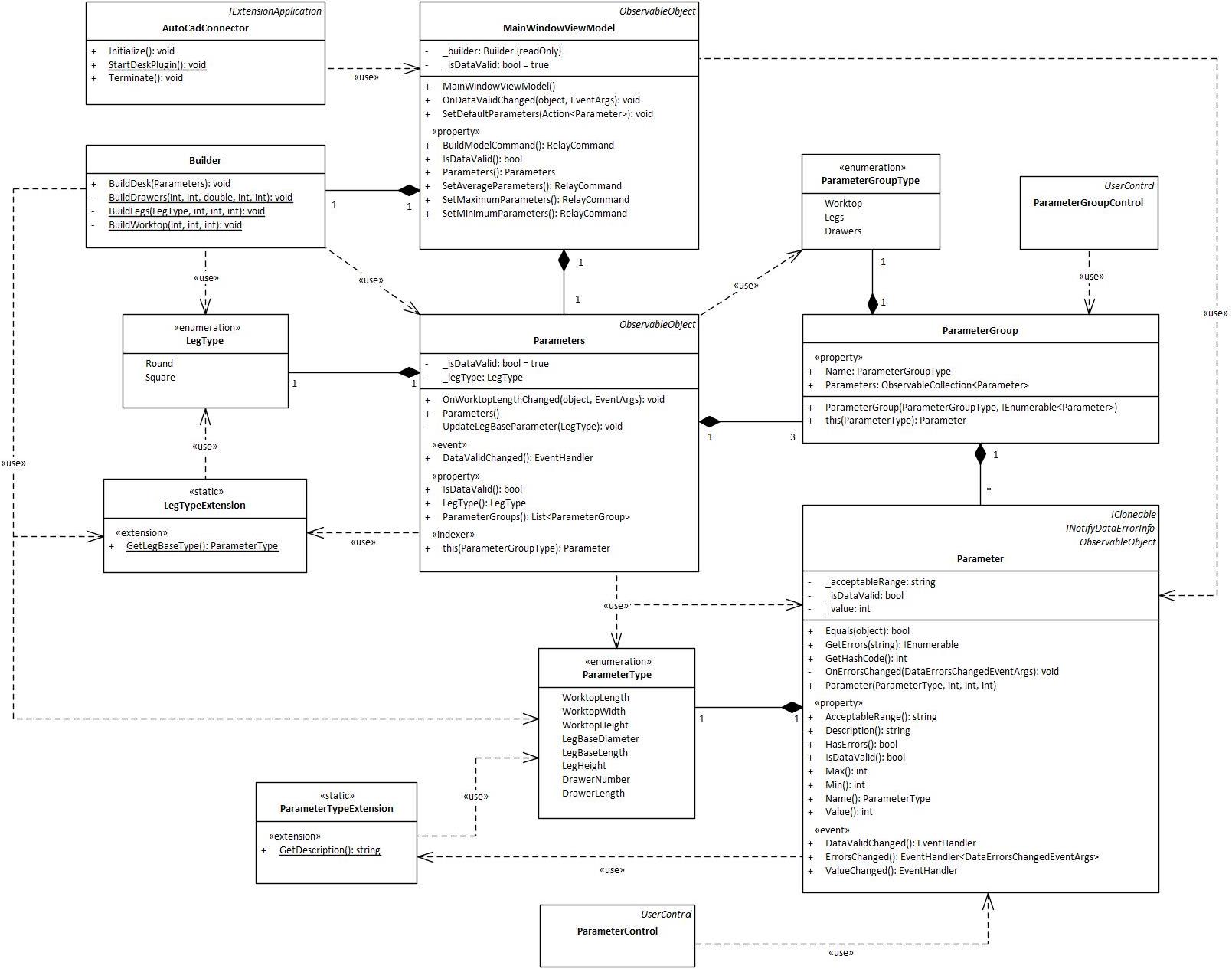


Рисунок 3.1 – UML-диаграмма классов плагина «Письменный стол»

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

Класс Parameter хранит данные о каждом параметре модели из перечисления ParameterType. Параметр хранит минимальное и максимальное значение параметра, а также текущее значение. Свойство AcceptableRange возвращает диапазон допустимых значений параметра в виде строки. Свойство Description возвращает текстовое описание параметра, которое получает с помощью метода расширения из класса ParameterTypeExtension, расширяющего перечисление ParameterType.

Класс ParameterGroup хранит данные об определенной группе параметров из перечисления ParameterGroupType в виде списка.

Все параметры, принадлежащие определенной группе, хранятся в классе Parameters в коллекции типа ObservableCollection. Данный класс также хранит типы ножек стола, описанные в перечислении LegType.

Класс LegTypeExtension расширяет перечисление LegType, позволяя получить тип параметра из перечисления ParameterType, соответствующий типу ножек стола.

Для построения модели письменного стола используется класс Builder. Метод BuildDesk строит 3D-модель на основании списка параметров, передаваемого в данный метод с объектом класса Parameters.

Класс MainWindowViewModel связывает модели (models) и представление (view) через механизм привязки данных (binding), а также содержит команды (RelayCommand) для выполнения построения модели письменного стола в AutoCAD и задания параметров 3D-модели по умолчанию.

ParameterControl и ParameterGroupControl являются пользовательскими элементами (UserControl) и введены с целью уменьшения дублирования кода и дальнейшего повторного использования в проекте. Они содержат разметку на языке XAML, соответственно, для представления параметра (Parameter) и группы параметров (ParameterGroup).

## Макеты пользовательского интерфейса

Макет пользовательского интерфейса представлен на рисунке 3.2.

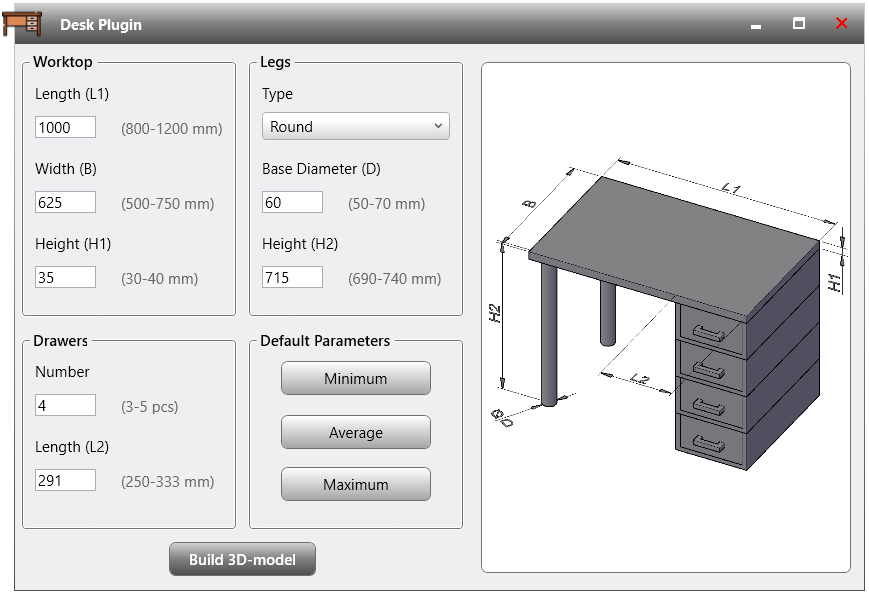


Рисунок 3.2 – Макет пользовательского интерфейса

Окно пользовательского интерфейса (рисунок 3.2) содержит:

* поля для ввода параметров модели письменного стола;
* надписи рядом с полями ввода, указывающие допустимый диапазон значений параметров;
* изображение модели письменного стола с буквенными обозначениями параметров;
* набор кнопок для задания параметров по умолчанию;
* кнопку построения модели, по нажатию на которую в AutoCAD будет производиться построение модели.

Все элементы пользовательского интерфейса сгруппированы по своему функциональному назначению для удобства взаимодействия с программой: так, длина, ширина и высота столешницы образуют группу параметров для построения столешницы (Worktop), кнопки задания параметров по умолчанию выделены в отдельную группу (Default Parameters) и т.д.

По умолчанию выбранный тип ножек 3D-модели письменного стола – «круглые» (Round). В зависимости от выбранного из выпадающего списка типа ножек стола изменяется надпись над полем ввода размера основания ножек, а также обновляется изображение модели письменного стола, как показано на рисунке 3.3.

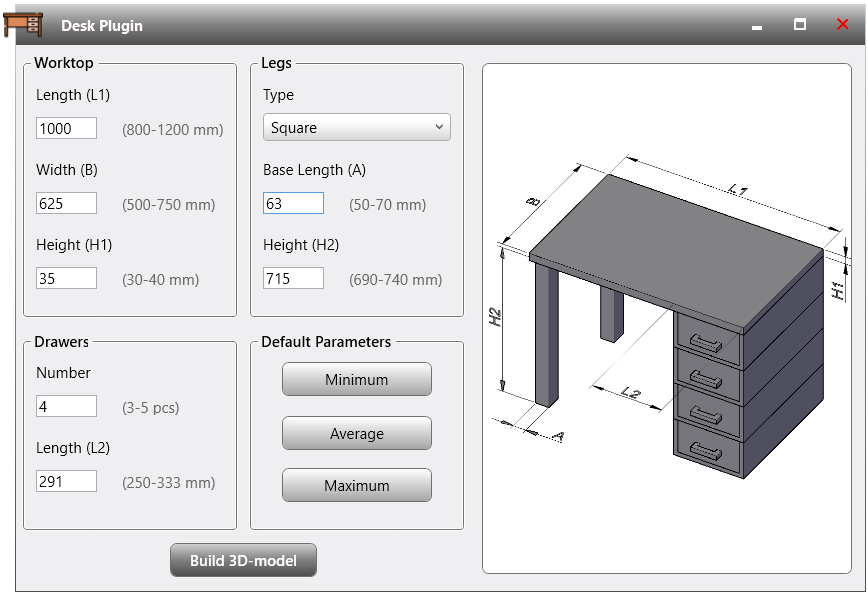


Рисунок 3.3 – Изменение надписи к текстовому полю для ввода размера основания ножек и изображения модели письменного стола при выборе квадратного типа основания

Кнопки в группе параметров по умолчанию (Default Parameters) позволяют выбрать минимально возможные, усредненные (задаются по умолчанию при запуске плагина) и максимально возможные параметры модели. Пример задания параметров по умолчанию приведен на рисунке 3.4.

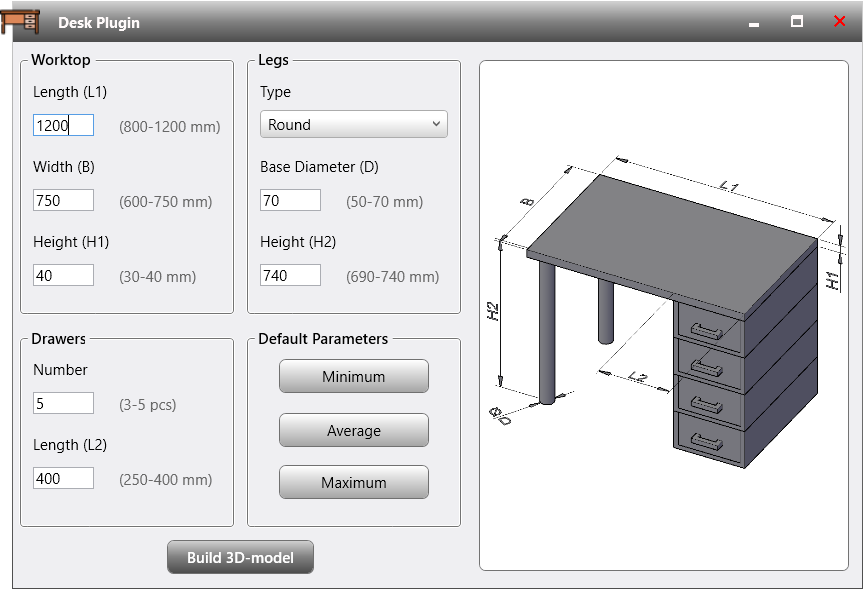


Рисунок 3.4 – Пример задания максимальных параметров модели письменного стола

Если пользователь вводит недопустимые значения параметров, кнопка построения модели становится недоступной, а поле для ввода, содержащее недопустимое значение параметра, выделяется красным цветом. Для большего привлечения внимания пользователя к допущенной ошибке справа от поля ввода отображается значок – красный круг с белым восклицательным знаком внутри. При наведении курсора мыши на поле, в котором произошла ошибка, появляется всплывающая подсказка с пояснением ошибки. Кроме того, поля для ввода параметров, зависимых от данного параметра, также подсвечиваются красным цветом, а справа от них на месте надписи с диапазоном допустимых значений появляется надпись Error (рисунок 3.5).

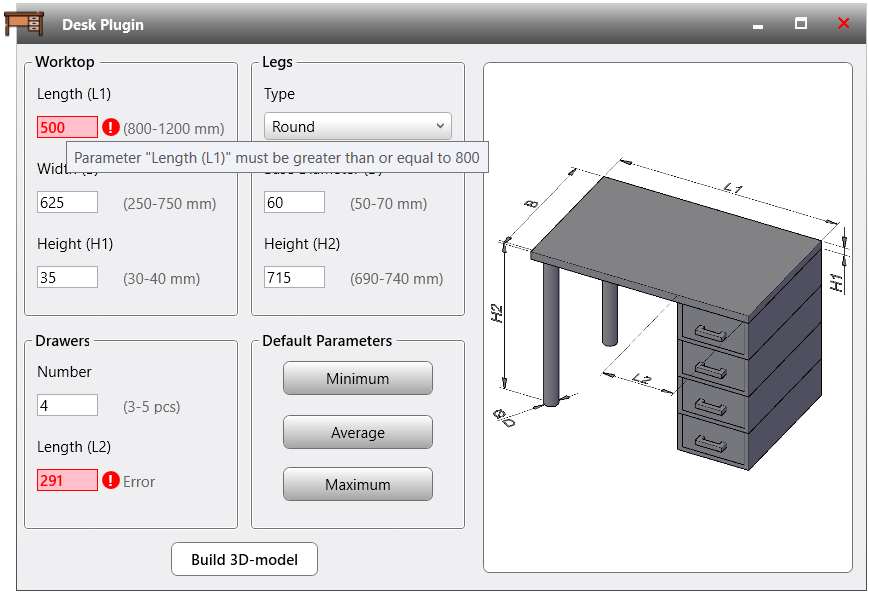


Рисунок 3.5 – Пример обработки ввода значения параметра, выходящего за пределы допустимого диапазона

# Заключение

В ходе разработки проекта системы изучены основные этапы проектирования программного продукта, исследована предметная область предмета проектирования, произведено знакомство с программным интерфейсом AutoCAD .NET API, а также рассмотрены некоторые аналоги плагина «Письменный стол» для САПР AutoCAD.

В результате разработки данного проекта системы возможна реализация плагина для автоматизации построения модели письменного стола в САПР AutoCAD 2022.

# Список использованных источников

Норенков И.П. Основы автоматизированного проектирования. М: МГТУ, 2002. 336 с.

Мазеин П.Г., Шаламов А.В. Сквозное автоматизированное проектирование в CAD/CAM системах: учеб. пособие. – Челябинск: ЮУрГУ, 2002. 83 с.

API [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 17.09.2022).

Плагин [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 14.09.2022).

AutoCAD [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/AutoCAD> (дата обращения: 15.09.2022).

AutoCAD Software [Электронный ресурс]: официальный сайт компании Autodesk. URL: <https://www.autodesk.com/products/autocad/overview> (дата обращения: 15.09.2022).

AutoCAD Platform Technologies [Электронный ресурс]: официальный сайт компании Autodesk. URL: https://www.autodesk.com/developer-network/platform-technologies/autocad (дата обращения: 17.09.2022).

AutoCAD 2022 Developer and ObjectARX Help [Электронный ресурс]: официальная справка Autodesk AutoCAD 2022 Help. URL: https://help.autodesk.com/view/OARX/2022/RUS/?guid=GUID-C3F3C736-40CF-44A0-9210-55F6A939B6F2 (дата обращения: 17.09.2022).

Создание плагинов для AutoCAD с помощью .NET API (часть 1 – первые шаги) [Электронный ресурс]: веб-сайт в формате системы тематических коллективных блогов Хабр. URL: https://habr.com/ru/post/235723/ (дата обращения 18.09.2022).

Введение в .NET. AutoCAD .NET API – презентация онлайн [Электронный ресурс]: просмотр PPT-файлов онлайн. URL: https://ppt-online.org/476500 (дата обращения: 19.09.2022).

Auto-Furniture [Электронный ресурс]: сайт компании FurnitureSoft. URL: <https://www.furnituresoft-software.com/auto> (дата обращения: 20.09.2022).

SymbolDesigner Beta [Электронный ресурс]: сайт компании CAD Division. URL: <http://www.caddivision.com/en/software/symboldesigner> (дата обращения: 20.09.2022).

ГОСТ 20400-2013. Продукция мебельного производства. Термины и определения. М.: Стандартинформ, 2014. 41 с.

Фаулер М. UML. Основы. 3-е издание. СПб.: Символ-Плюс, 2019. 192 с.