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

« 03 » октября 2022

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ А.А. Калентьев  
« 03 » октября 2022

Томск 2022

Оглавление

[Введение 3](#_Toc115697786)

[1 ОПИСАНИЕ САПР 5](#_Toc115697787)

[1.1 Описание программы 5](#_Toc115697788)

[1.2 Описание API 6](#_Toc115697789)

[1.3 Обзор аналогов 18](#_Toc115697790)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 21](#_Toc115697791)

[3 ПРОЕКТ ПРОГРАММЫ 24](#_Toc115697792)

[3.1 Диаграмма классов 24](#_Toc115697793)

[3.2 Макеты пользовательского интерфейса 28](#_Toc115697794)

[Заключение 33](#_Toc115697795)

[Список использованных источников 34](#_Toc115697796)

# 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 – 1.23 представлены необходимые свойства и методы классов и структур 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 |

Таблица 1.2 – Методы класса Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращаемый тип | Описание |
| Аргумент | Тип | Описание |
| Show  Modeless  Window | formTo  Show | Form | Окно WPF, которое нужно отобразить | void | Отображает окно WPF в немодальном режиме. Применение данного метода вместо стандартного метода Show() предотвращает неожиданное поведение программы |

Таблица 1.3 – Свойства класса DocumentCollection

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| MdiActiveDocument | Document | Предоставляет доступ к активному документу с расширением MDI |

Таблица 1.4 – Свойства класса Document

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Database | Database | Возвращает объект базы данных, который используется данным документом |

Таблица 1.5 – Методы класса Document

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращаемый тип | Описание |
| Аргумент | Тип | Описание |
| Lock  Document | – | – | – | DocumentLock | Блокирует текущий документ. Используется для предотвращения конфликтов с другими запросами в базе данных при взаимодействии с AutoCAD через немодальное диалоговое окно |

Таблица 1.6 – Свойства класса Database

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| BlockTableId | ObjectId | Возвращает идентификатор объекта в таблице BlockTable базы данных |
| TransactionManager | TransactionManager | Предоставляет доступ к TransactionManager для текущей базы данных |

Таблица 1.7 – Методы класса TransactionManager

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращаемый тип | Описание |
| Аргумент | Тип | Описание |
| Start  Transaction | – | – | – | Transaction | Начинает новую транзакцию |

Таблица 1.8 – Методы класса Transaction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Add  Newly  Created  Object | obj | DB  Object | Объект для добавления в транзакцию или удаления из транзакции | void | Добавляет объект в транзакцию или удаляет его из транзакции |
| add | bool | Показывает, что нужно сделать с объектом (добавить или удалить) |
| GetObject | id | ObjectId | Уникальный идентификатор объекта в транзакции | DBObject | Получает объект, на данный момент открытый в транзакции, по его id |
| mode | Open  Mode | Объявляет права доступа к объекту (чтение, запись, уведомление) |
| Commit | – | – | – | void | Фиксирует изменения, внесенные во все объекты DBObject, открытые в течение транзакции, после чего закрывает их |

Таблица 1.9 – Методы класса Entity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| TransformBy | transform | Matrix  3d | Входная матрица преобра-зования | void | Применяет матрицу преобразования к указанному объекту |

Таблица 1.10 – Методы класса BlockTableRecord

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвра-щаемый тип | Описание |
| Аргумент | Тип | Описание |
| Append  Entity | entity | Entity | Добавляемая сущность | ObjectId | Добавляет сущность в базу данных и в таблицу блоков |
| GetBlockReferenceIds | directOnly | bool | Показывает, включать ли в список возвращаемых ссылок только те ссылки на блоки, которые имеют прямые ссылки на данную BlockTableRecord. Если данное значение равно true и блок является вложенным, родительские ссылки на данный блок не будут учитываться | ObjectId  Collection | Возвращает список ссылок на блоки, которые прямо или косвенно (через вложенность блоков) ссылаются на данный блок |
| force  Validity | bool | Показывает, должны ли старые чертежи, которые были загружены по требованию, быть загружены полностью, чтобы найти их идентификаторы BlockReferences. Данный параметр применим, только если directOnly равно false |

Таблица 1.11 – Методы класса Solid3D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвра-щаемый тип | Описание |
| Аргумент | Тип | Описание |
| Create  Box | length  AlongX | double | Длина примитива | void | Создает твердотельный примитив прямоугольной формы с центром тяжести в начале мировой системы координат (WCS) |
| length  AlongY | double | Ширина примитива |
| length  AlongZ | double | Высота примитива |
| Extrude | region | Region | Область выдавливания | void | Создает твердое тело посредством выдавливания области region на заданное расстояние по высоте height с заданным углом конусности taperAngle |
| height | double | Высота выдавливания |
| taper  Angle | double | Угол конусности |
| Boolean  Operation | type | Boolean  Operation  Type | Тип булевой операции (объединение, пересечение, вычитание) | void | Выполняет булеву операцию между двумя 3D-объектами |
| solid | Solid3d | Второй 3D-объект |

Таблица 1.12 – Свойства класса Circle

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Center | Point3d | Предоставляет доступ к центру окружности |
| Diameter | void | Предоставляет доступ к диаметру окружности |

Таблица 1.13 – Методы класса DBObject

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Erase | – | – | – | void | Устанавливает бит стирания объекта |
| Upgrade  Open | – | – | – | void | Если в данный момент объект открыт только для чтения (OpenMode.ForRead), данный метод обновляет объект, чтобы открыть его для записи (OpenMode.ForWrite). В процессе изменения доступа объект закрывается, тем самым вызывая любое соответствую-щее уведомление |

Таблица 1.14 – Методы класса SymbolTable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Add | value | SymbolTable  Record | Значение идентифика-тора записи | ObjectId | Добавляет значение идентификато-ра записи как в базу данных, содержащую таблицу, так и в саму таблицу |

Окончание таблицы 1.14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Has | key | string | Ключ – имя записи в виде строки | bool | Возвращает значение true, если таблица содержит запись с именем, соответствую-щим ключу. Если совпадений не найдено, возвращается значение false |

Таблица 1.15 – Свойства класса ObjectIdCollection

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Count | int | Возвращает общее количество объектов в коллекции |

Таблица 1.16 – Методы класса DBObjectCollection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращаемый тип | Описание |
| Аргумент | Тип | Описание |
| Add | value | Db  Object | Объект для добавления в коллекцию | int | Добавляет объект в данную коллекцию |

Таблица 1.17 – Методы класса Region

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Create  From  Curves | curve  Sergments | DB  Object | Коллекция кривых для создания замкнутых циклов | DB  Object  Collection | Создает набор объектов Region из замкнутых циклов, которые представлены кривыми |

Таблица 1.18 – Методы класса Polyline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Add  Vertex  At | index | int | Индекс вершины | void | Добавляет новую вершину к полилинии |
| pt | Point2d | Точка расположения вершины |
| bulge | double | Выпуклость вершины |
| startWidth | double | Начальная ширина вершины |
| endWidth | double | Конечная ширина вершины |

Таблица 1.19 – Свойства класса Drawable

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Bounds | Extents3d? | Возвращает границы объекта рисования, если таковые имеются. Возвращает значение null, если у объекта рисования нет границ или если границы не могут быть вычислены |

Таблица 1.20 – Свойства структуры Extents3d

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| MaxPoint | Point3d | Возвращает максимальную точку |

Таблица 1.21 – Свойства структуры Point3d

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Origin | Point3d | Возвращает начало координат 3D-пространства |
| X | double | Возвращает координату по оси X |
| Y | double | Возвращает координату по оси Y |
| Z | double | Возвращает координату по оси Z |

Таблица 1.22 – Методы структуры Point3d

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| GetVectorTo | point | Point3d | Заданная точка | Vector3d | Вычисляет и возвращает вектор относительно указанной точки |

Таблица 1.23 – Методы структуры Matrix3d

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Входные аргументы | | | Возвращае-мый тип | Описание |
| Аргумент | Тип | Описание |
| Displacement | vector | Vector  3d | Входной вектор для вычисления смещения |  | Возвращает вектор смещения |

## 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.

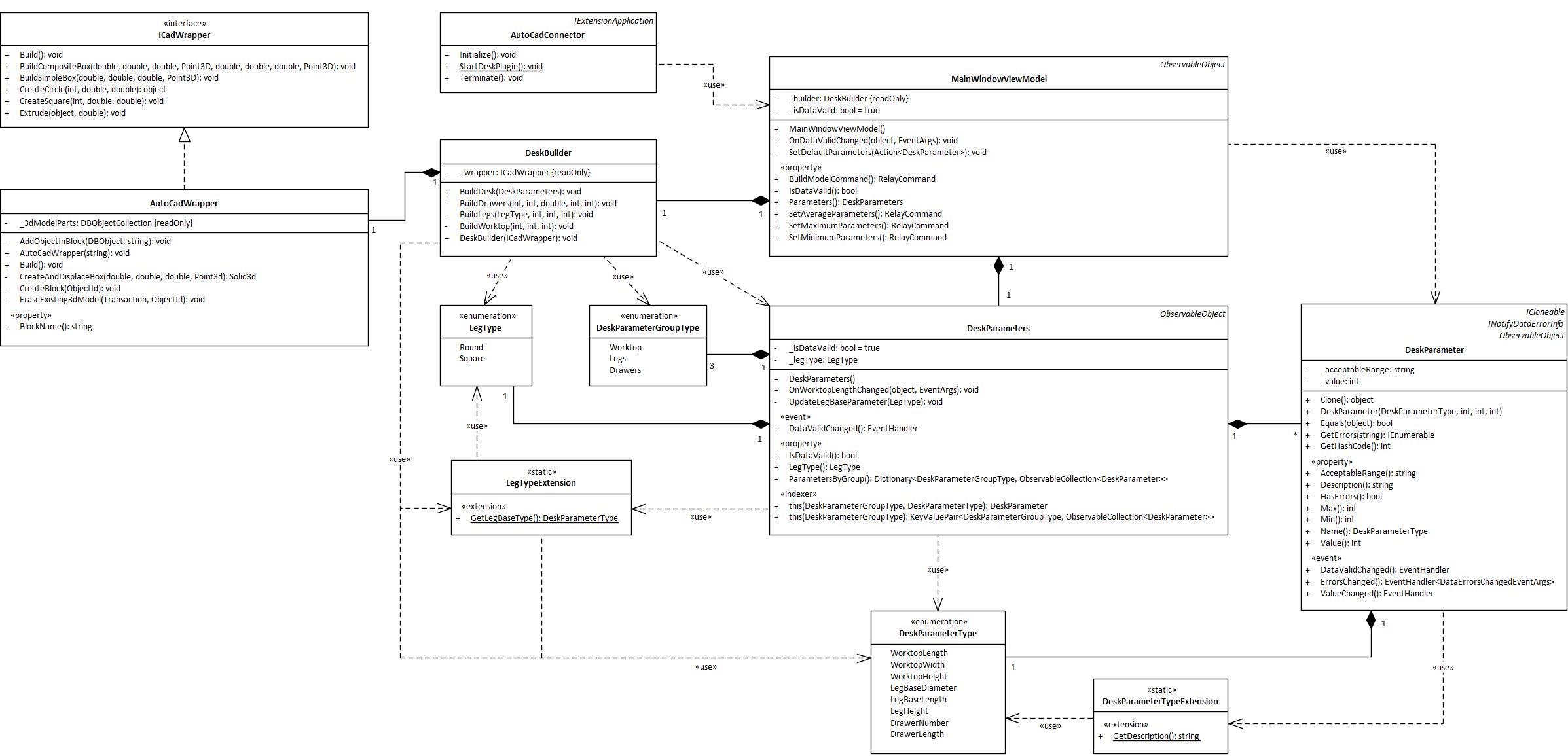


Рисунок 3.1 – UML-диаграмма классов плагина «Письменный стол»

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

Класс DeskParameter хранит данные о каждом параметре модели из перечисления DeskParameterType. 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 из класса DeskParameterTypeExtension, расширяющего перечисление ParameterType.

Все параметры, принадлежащие определенной группе, хранятся в классе DeskParameters в свойстве ParametersByGroup, возвращающем словарь, в котором ключом является группа параметров (значение перечисления DeskParameterGroupType), а значением – наблюдаемая коллекция параметров ObservableCollection<Parameter>. Данный класс также хранит типы ножек стола, описанные в перечислении LegType.

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

Также планируется обеспечить возможность работы плагина «Письменный стол» не только с AutoCAD, но с другими САПР. Для этого будет использован механизм изолирования обращения к методам API некоторой САПР посредством отдельного класса-обертки (от англ. wrapper). В случае с использованием AutoCAD такой класс носит название AutoCadWrapper. Данный класс работает непосредственно со свойствами и методами различных классов и структур AutoCAD .NET API.

Любой класс-обертка должен обеспечить реализацию интерфейса ICadWrapper, в котором содержатся определения сигнатур методов, отвечающих за основные операции для построения 3D-модели в САПР (построение параллелепипеда, создание окружности, выполнение операции выдавливания и т.д.). Реализация этих методов, а также каких-либо дополнительных методов для работы с API используемой САПР определяется уже на уровне соответствующего класса-обертки.

Непосредственно для построения 3D-модели письменного стола используется класс DeskBuilder. DeskBuilder хранит в себе объект класса-обертки используемой САПР через интерфейс ICadWrapper и при вызове метода BuildDesk() обращается к методам класса-обертки, чтобы построить 3D-модель на основании списка параметров, передаваемого в данный метод с объектом класса DeskParameters. Таким образом, для построения 3D-модели DeskBuilder не использует класс-обертку напрямую, а обращается только к тем его методам, которые являются реализациями методов интерфейса ICadWrapper.

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.

## Макеты пользовательского интерфейса

Макет пользовательского интерфейса представлен на рисунке 3.2.

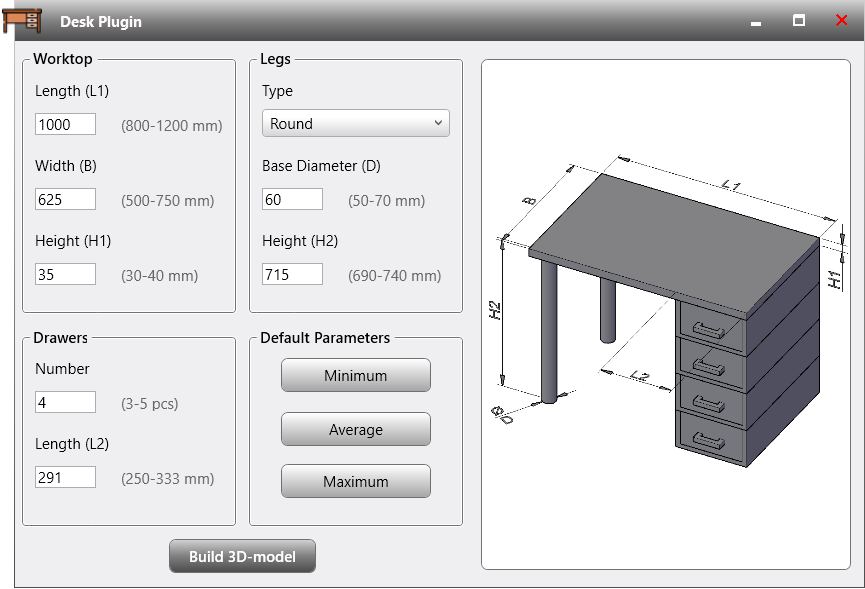


Рисунок 3.2 – Макет пользовательского интерфейса

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.

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 3.3.

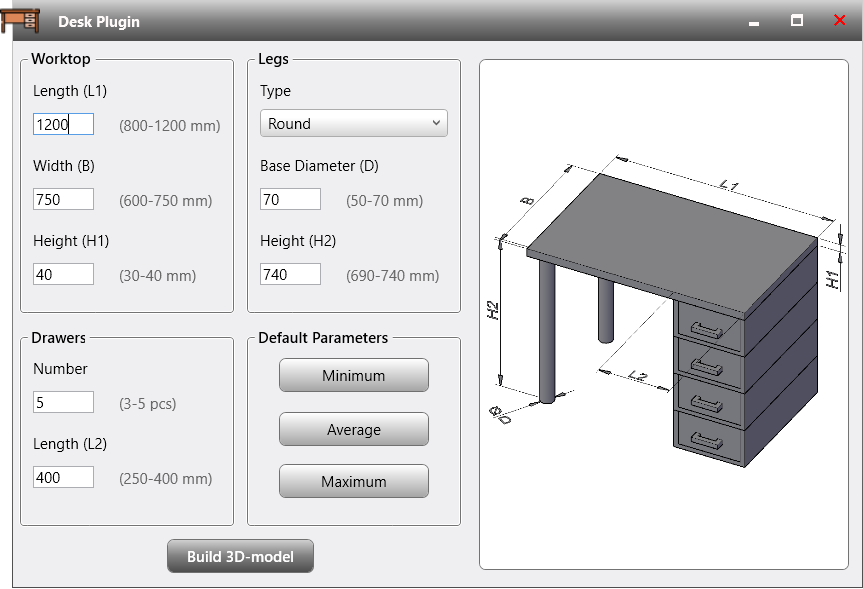


Рисунок 3.3 – Пример задания максимальных параметров модели письменного стола

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 3.4).

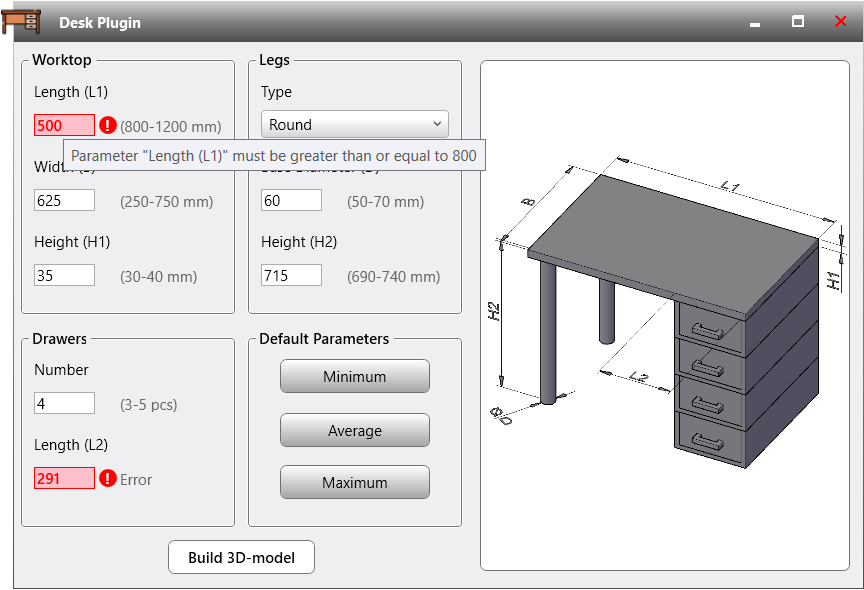


Рисунок 3.4 – Пример обработки ввода значения параметра, выходящего за пределы допустимого диапазона

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 рисунках 3.5 – 3.6.

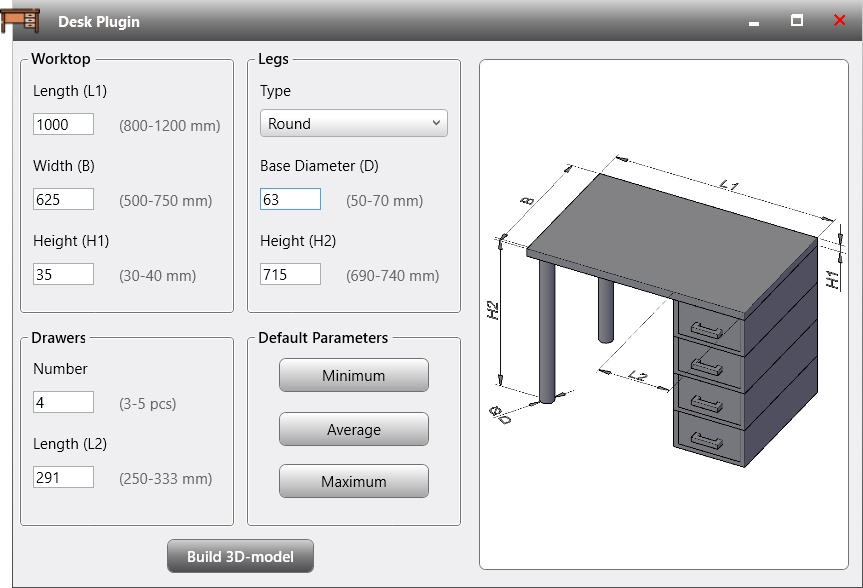


Рисунок 3.5 – Изменение надписи к текстовому полю для ввода размера основания ножек и изображения модели письменного стола при выборе круглого типа основания

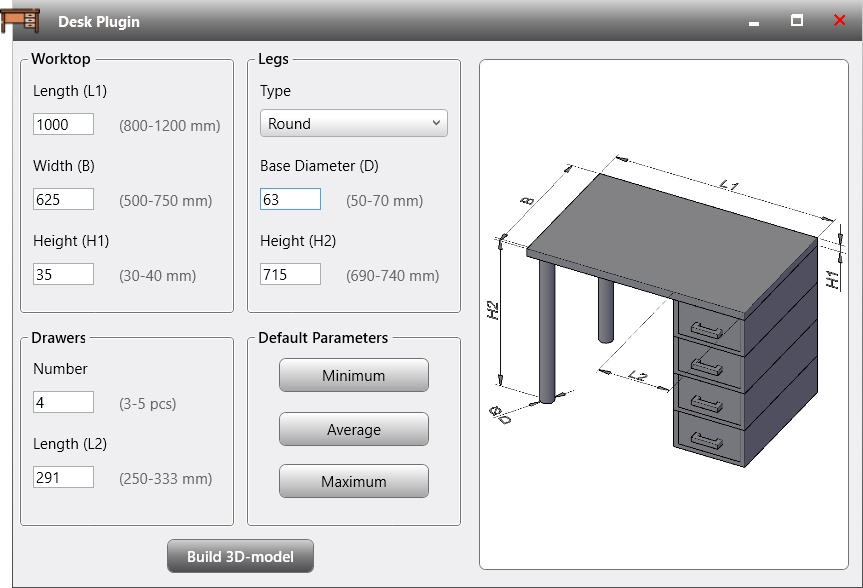


Рисунок 3.6 – Изменение надписи к текстовому полю для ввода размера основания ножек и изображения модели письменного стола при выборе квадратного типа основания

Кроме того, при возникновении ошибки в текстовом поле для ввода размера основания ножек письменного стола содержание всплывающей подсказки об ошибке изменяется в зависимости от того, какой тип ножек выбран пользователем в данный момент (рисунки 3.7 – 3.8).

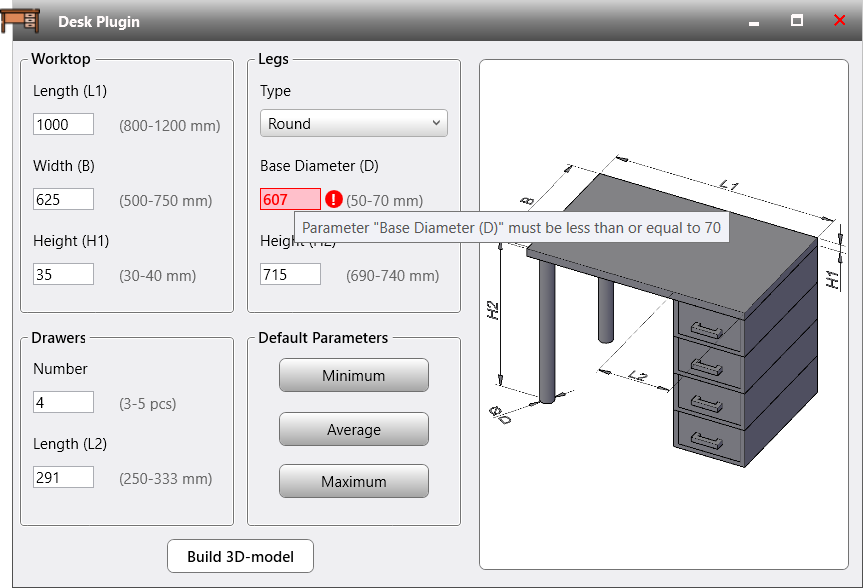


Рисунок 3.7 – Содержание всплывающей подсказки в момент, когда пользователем выбрано круглое основание ножек

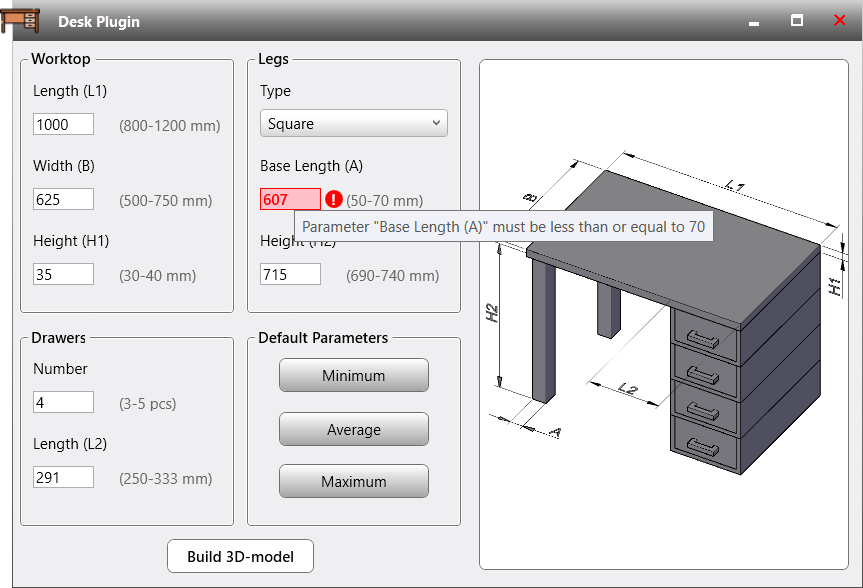


Рисунок 3.8 – Содержание всплывающей подсказки в момент, когда пользователем выбрано квадратное основание ножек

# 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: 22.09.2022).

Плагин [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 22.09.2022).

AutoCAD [Электронный ресурс]: Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/AutoCAD> (дата обращения: 22.09.2022).

AutoCAD Software [Электронный ресурс]: официальный сайт компании Autodesk. URL: <https://www.autodesk.com/products/autocad/overview> (дата обращения: 23.09.2022).

AutoCAD Platform Technologies [Электронный ресурс]: официальный сайт компании Autodesk. URL: https://www.autodesk.com/developer-network/platform-technologies/autocad (дата обращения: 25.09.2022).

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: 25.09.2022).

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 27.09.2022).

Введение в .NET. AutoCAD .NET API – презентация онлайн [Электронный ресурс]: просмотр PPT-файлов онлайн. URL: https://ppt-online.org/476500 (дата обращения: 28.09.2022).

Auto-Furniture [Электронный ресурс]: сайт компании FurnitureSoft. URL: <https://www.furnituresoft-software.com/auto> (дата обращения: 29.09.2022).

SymbolDesigner Beta [Электронный ресурс]: сайт компании CAD Division. URL: <http://www.caddivision.com/en/software/symboldesigner> (дата обращения: 29.09.2022).

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.