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 «МАНГАЛ»**

**ДЛЯ «AutoCAD»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

По дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

Студент гр. 580-2

\_\_\_\_\_\_\_\_\_\_ Оксингерт В.К.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Оглавление

[1 Описание САПР 3](#_Toc152556829)

[1.1 Описание программы 3](#_Toc152556830)

[1.2 Описание API 3](#_Toc152556831)

[1.3 Обзор аналогов плагина 6](#_Toc152556832)

[2 Описание предмета проектирования 8](#_Toc152556833)

[3 Проект системы 10](#_Toc152556834)

[3.1 Диаграмма классов 10](#_Toc152556835)

[3.2 Макеты пользовательского интерфейса 14](#_Toc152556836)

[3.3 Пользовательский интерфейс 14](#_Toc152556837)

[4 Список источников 17](#_Toc152556838)

# Описание САПР

## Описание программы

Autodesk AutoCAD — система автоматизированного проектирования (САПР) для создания трёх- и двухмерных моделей. Позволяет выполнять построение 3D-моделей деталей, объединять их в сборки, а также выполнять чертежи и инженерные расчёты физических характеристик. AutoCAD и специализированные приложения на его основе применяются в области машиностроения, строительства, архитектуры и т.д. Программа имеет русскую локализацию [1].

Аналоги AutoCAD: Autodesk Inventor, Autodesk Fusion 360, SolidWorks, Компас 3D.

## Описание API

API (Application Program Interface) – программный интерфейс приложения, набор функций, позволяющий взаимодействовать с программой через другие программы. Для AutoCAD есть API на двух языка программирования: C#/.NET и Python. Для разработки плагина, рассматриваемого в данной работе, будет использоваться API для языка C#/.NET [2].

Основные библиотеки API:

• AcCoreMgd.dll Используется при построении графиков, а также при определении команд и функций, используемых в AutoLISP;

• AcDbMgd.dll Используется при работе с объектами, хранящимися в файле чертежа;

• AcMgd.dll Используется при работе с приложением и пользовательским интерфейсом;

• AcCui.dll Используется при работе с файлами настроек.

Некоторые используемые методы и классы API представлены в таблицах 1.1 – 1.6.

Таблица 1.1 – Некоторые используемые классы API

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| MdiActiveDocument() | Document | Метод для создания и получения документа чертежа |
| MdiActiveDocument.Editor() | Editor | Метод для получения редактора текущего чертежа |

Таблица 1.2 – Некоторые используемые методы API

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| TransactionManager | TransactionManager | Доступ к TransactionManager для базы данных. |

Таблица 1.3 – Основные методы интерфейса DocumentManager

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| StartTransaction | Transaction | Начинает новую транзакцию. |

Таблица 1.4 – Основные методы интерфейса TransactionManager

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Commit |  | void | Функция фиксирует изменения, внесенные во все объекты DBObject, открытые во время транзакции, а затем закрывает их. |
| GetObject | ObjectId, DatabaseServices.OpenMode | DBObject | Функция вызывает функцию Open() верхней транзакции, передавая все полученные аргументы. |
| AddNewlyCreatedDBObject | DBObject, [MarshalAs(UnmanagedType.U1)] bool | void | Если add == true, объект, на который указывает obj, добавляется в верхнюю транзакцию. Если add == false, то объект удаляется из любой транзакции, в которой он находится. |

Таблица 1.5 – Основные методы класса BlockTableRecord

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Типа возвращаемых данных | Описание |
| AppendEntity | [CallerMustClose] Entity | ObjectId | Добавляет объект в базу данных и запись таблицы блоков. |
| GetObject | ObjectId, DatabaseServices.OpenMode | DBObject | Функция вызывает функцию Open() верхней транзакции, передавая все полученные аргументы. |

Таблица 1.6 – Основные методы класса Solid3d

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Add | Point3d | int | Добавляет объект, представленный значением, в эту коллекцию. |

## Обзор аналогов плагина

Прямых аналогов плагина для построения мангала нет.

SolidWorks может работать с листовым металлом, который обычно используется в качестве вкладышей для компонентов или для создания опоры для других компонентов. Мангал состоит из листов металла определённой толщины, так что с помощью инструментов SolidWorks можно спроектировать мангал [3].

Пользовательский интерфейс SolidWorks представлен на рисунке 1.2.

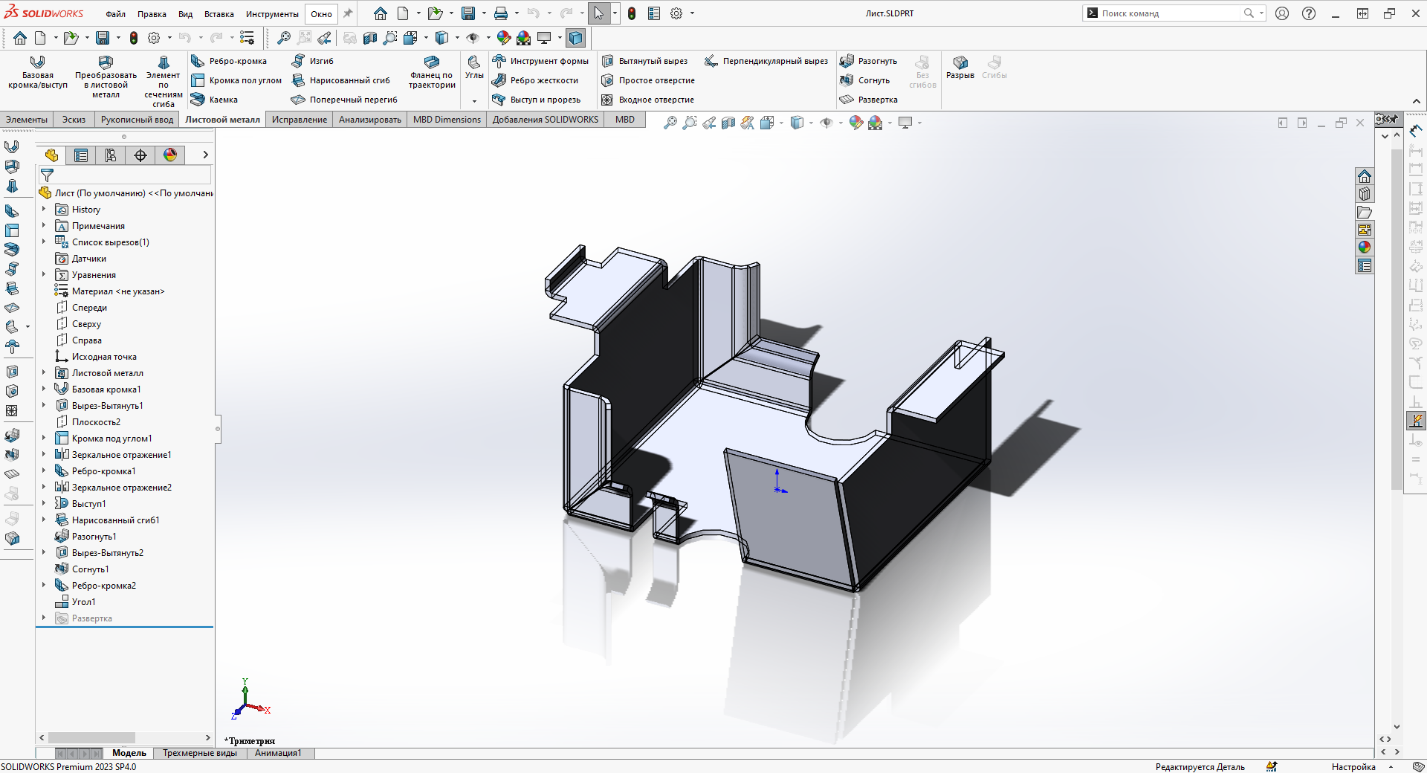


Рисунок 1.2 – Пользовательский интерфейс SolidWorks

Плагин SoftDraft Stainless Steel Shapes 2DS может создавать изделия из металла различных форм: W-образные и S-образные формы, стандартные и различные каналы, равные и неравные углы, T-образные и W-образные стержни.

Особенности плагина:

• Создавайте фигуры в плане, сечении и на высоте по мере необходимости;

• Фигуры в разрезе создаются как блок AutoCAD, что позволяет минимизировать размер файла;

• Все фигуры создаются параметрически по требованию;

• Фигуры привязываются к курсору на основе центральной точки вставки и поворачиваются после размещения [4].

# Описание предмета проектирования

Предметом проектирования является мангал.

Мангал служит для разведения огня с целью последующего приготовления мясных и овощных блюд. Чертёж мангала представлен на рисунке 2.1.



Рисунок 2.1 - Чертёж мангала

У пользователя есть возможность менять параметры мангала, приведённые ниже:

• длина мангала L (500 – 2000 мм);

• ширина мангала W (300 – 500 мм);

• толщина стенок мангала Ww (2 – 8 мм);

• высота короба мангала Hg (200 – 500 мм);

• высота ножки мангала Hl (500 – 1000 мм);

• диаметр ножки мангала Dl (не меньше мм, не больше мм);

• высота центра отверстия для прохода воздуха (не больше мм, не меньше мм);

• диаметр отверстия для прохода воздуха Dh (не больше мм, не меньше 10 мм);

• диаметр паза для шампура Dg (5 – 20 мм);

• расстояние между отверстиями для прохода воздуха dH (не больше мм, не меньше Dh мм. Количество отверстий определяется автоматически исходя из данного параметра и длины забора);

• расстояние между пазами для шампуров dg (не больше мм, не меньше Dg мм. Количество отверстий определяется автоматически исходя из данного параметра и длины забора).

# Проект системы

## Диаграмма классов

UML-диаграмма классов — тип статической структурной диаграммы, описывающей структуру системы посредством обозначения классов, их атрибутов, методов, связей на диаграмме [5].

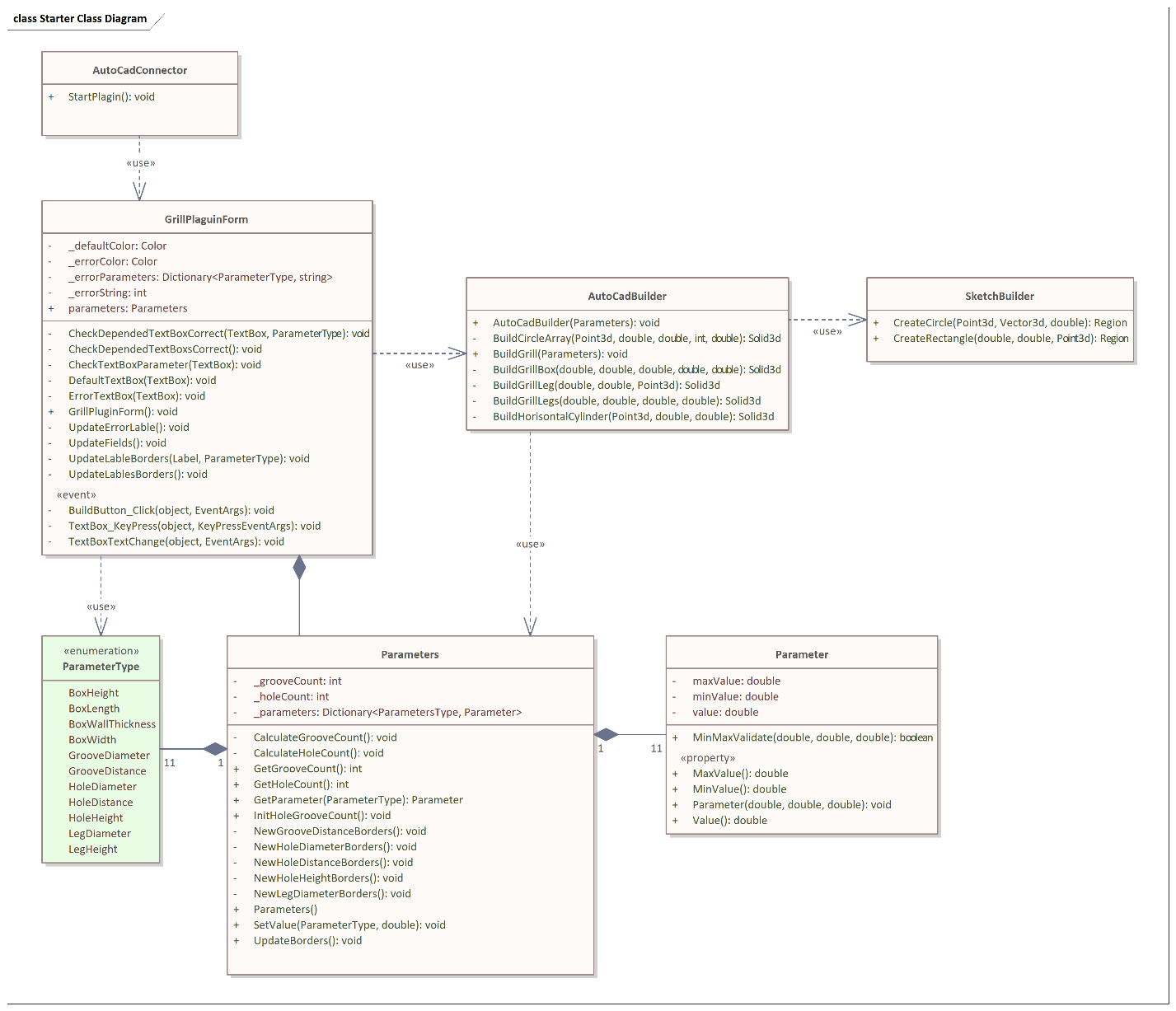


Рисунок 3.1 - Диаграмма классов приложения

AutoCadWrapper – класс обёртка API САПР AutoCAD. Вызывает форму GrillPlaguinForm и после вызывает AutoCadBuilder, отдавая ему Parameters из формы (таблица 3.1).

Таблица 3.1 – Описание класса AutoCadWrapper

|  |  |
| --- | --- |
| Название | Описание |
| StartPlaguin | Вызывает форму GrillPlaguinForm |
| BuildGrill | Принимает Parameters и вызывает класс AutoCadBuilder |

AutoCadBuilder – класс использующий методы API, для построения мангала в AutoCAD по параметрам Parameters (таблица 3.2).

Таблица 3.2 – Описание класса AutoCadBuilder

|  |  |
| --- | --- |
| Название | Описание |
| BuildGrill | Принимает Parameters и строит мангал методами API AutoCAD |
| BuildGrillBox | Принимает параметры и строит короб мангала |
| BuildGrillLegs | Принимает параметры ножек мангала и строит их |
| BuildGrillLeg | Строит 1 ножку по полученным параметрам |
| BuildCircleArray | Строит массив горизонтально направленных цилиндров |
| BuildHorisontalCylinder | Строит 1 горизонтально направленный цилиндр |

SketchBuilder– класс использующий методы API, для построения эскизов (таблица 3.3).

Таблица 3.3 – Описание класса SketchBuilder

|  |  |
| --- | --- |
| Название | Описание |
| CreateCircle | Строит эскиз окружности по параметрам и возвращает его |
| CreateRectangle | Строит эскиз прямоугольника по параметрам и возвращает его |

Форма GrillPlaguinForm – главная форма приложения. Содержит в себе Parameters.

Parameters – класс содержащий в себе параметры мангала. Состоит из словаря данных и двух параметров, высчитываемых в зависимости от параметров в словаре (таблица 3.4).

Таблица 3.4 – Описание класса Parameters

|  |  |
| --- | --- |
| Название | Описание |
| \_grooveCount | Количество пазов для шампуров мангала |
| \_holeCount | Количество отверстий для вентиляции мангала |
| \_parameters | Словарь параметров мангала |
| InitHoleGrooveCount | Использует методы CalculateGrooveCount и CalculateHoleCount для обновления параметров \_grooveCount и \_holeCount |
| CalculateGrooveCount | Высчитывает количество пазов для шампуров мангала |
| CalculateHoleCount | Высчитывает количество отверстий для вентиляции мангала |
| UpdateBorders | Использует все методы, определяющие новые границы |
| NewHoleDistanceBorders | Задаёт новые границы расстояния между отверстиями для вентиляции |
| NewHoleHeightBorders | Задаёт новые границы высоты центра отверстия для вентиляции мангала |
| NewHoleDiameterBorders | Задаёт новые границы диаметра отверстия для вентиляции |
| NewGrooveDistanceBorders | Задаёт новые границы расстояния между пазами для шампуров |
| NewLegDiameterBorders | Задаёт новые границы диаметра ножек |
| GetGrooveCount | Возвращает количество пазов для шампуров |
| GetHoleCount | Возвращает количество отверстий для вентиляции |
| SetValue | Задаёт значение определённого параметра |
| GetParameter | Возвращает значение определённого параметра |
| Parameters | Конструктор, задающий начальные значения параметров мангала |

ParameterType – перечисление параметров мангала (таблица 3.5).

Таблица 3.5 – Описание перечисления ParameterType

|  |  |
| --- | --- |
| Название | Описание |
| boxHeight | Высота короба мангала |
| boxLength | Длина короба мангала |
| boxWallThickness | Толщина стен мангала |
| boxWidth | Ширина короба мангала |
| grooveDiameter | Диаметр паза для шампура |
| grooveDistance | Расстояние между пазами для шампуров |
| holeDiameter | Диаметр отверстия для вентиляции |
| holeDistance | Расстояние между отверстиями для вентиляции |
| holeHeight | Высота центра отверстия для вентиляции |
| legDiameter | Диаметр ножки мангала |
| legHeight | Высота ножки мангала |

Parameter – класс параметр. Содержит в себе минимальное, максимальное и текущее значение (таблица 3.6).

Таблица 3.6 – Описание класса Parameter

|  |  |
| --- | --- |
| Название | Описание |
| \_maxValue | Максимальное значение параметра |
| \_minValue | Минимальное значение параметра |
| \_value | Текущее значение параметра |
| MaxValue | Автосвойство \_maxValue |
| MinValue | Автосвойство \_minValue |
| Value | Автосвойство \_value |
| Parameter | Конструктор требующий три значения: значение, минимальное значение, максимальное значение |
| MinMaxValidate | Метод, определяющий входит ли текущее значение параметра и границы. При выходе за границы выдаёт ArgumentException с сообщением за какую границу значение вышло. |

## Макеты пользовательского интерфейса

Пользовательский интерфейс представляет собой форму для ввода параметров. Пользователь вводит значения самостоятельно, опираясь на подсказки, отображенные около полей. При нажатии на кнопку «Построить» проводится проверка зависимых параметров и, если условия соблюдены, строится 3D-модель мангала в САПР. Пример макета пользовательского интерфейса представлен на рисунке 3.2.

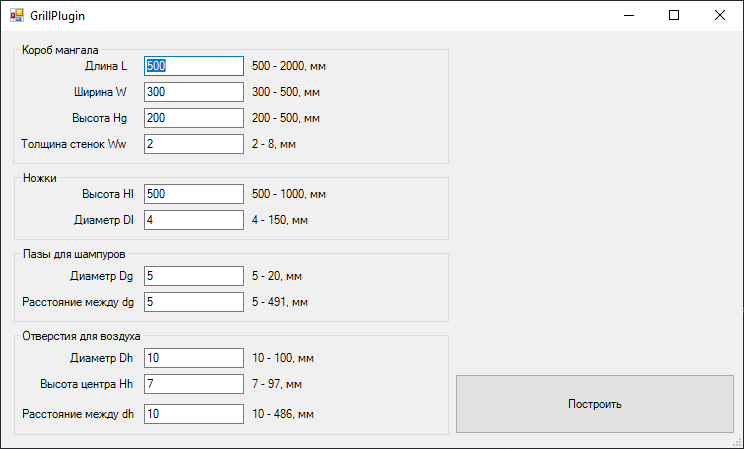


Рисунок 3.2 – Пример пользовательского интерфейса

Проверка правильности ввода значений проводится по ходу заполнения полей. Если поле заполнено неправильно, то есть пользователь ввел значение, выходящее за границы, то оно подсвечивается красным цветом, сигнализирующем об ошибке. Пример ввода некорректных данных представлен на рисунке 3.3.

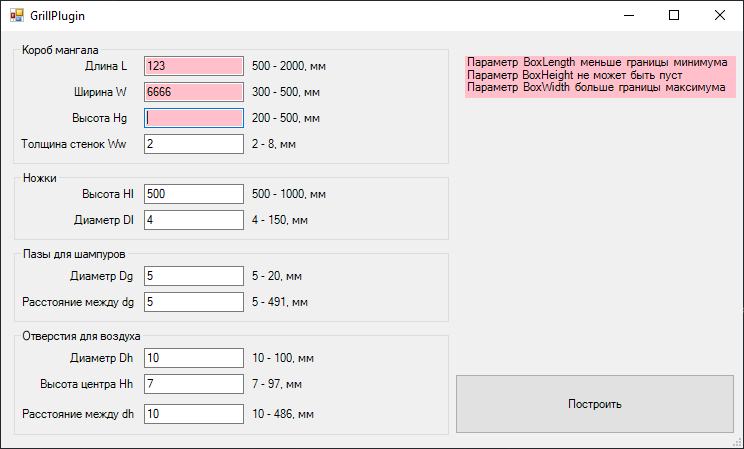


Рисунок 3.3 - Интерфейс с неправильно введённым значением параметра

Некоторые параметры мангала зависят от других. Вводя новые значения параметров, границы зависимых параметров автоматически пересчитываются и, если старое значение зависимого параметра выходит за пределы новых границ, поле зависимого параметра подсвечивается красным. Пример подобного ввода данных представлен на рисунке 3.4.

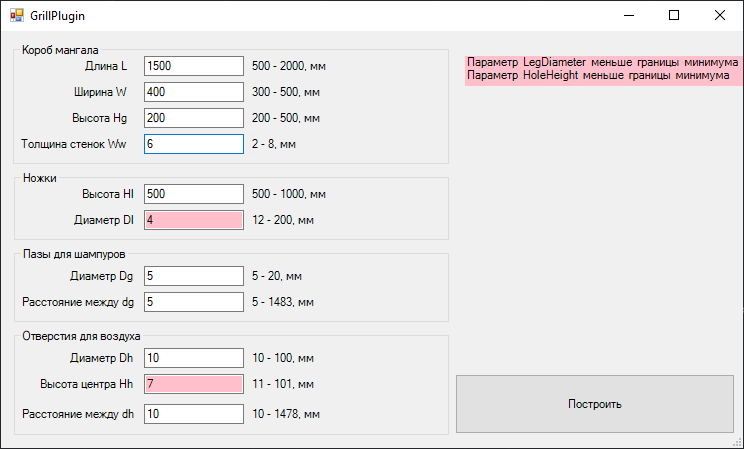


Рисунок 3.4 - Интерфейс с неправильно введённым значением зависимого параметра

# Список источников

1. Языковые пакеты для САПР Autodesk AutoCAD. [Электронный ресурс]. Режим доступа: свободный (дата образения 16.10.23). https://www.autodesk.com/support/technical/article/caas/tsarticles/ts/2edY5Oczsv93tDF1ugCKHU.html

2. Документация AutoCAD .NET API для разработчиков. [Электронный ресурс]. Режим доступа: свободный (дата обращения 16.10.23), <https://help.autodesk.com/view/OARX/2022/ENU/?guid=GUID-390A47DB-77AF-433A-994C-2AFBBE9996AE>

3. Листовой металл – 2021 – Справка по SOLIDWORKS [Электронный ресурс]. Режим доступа: свободный (дата обращения: 04.12.2023), <https://help.solidworks.com/2021/Russian/SolidWorks/sldworks/c_Sheet_Metal.htm>

4. SoftDraft Stainless Steel Shapes 2DS | AutoCAD | Autodesk App Store [Электронный ресурс]. Режим доступа: свободный (дата обращения: 04.12.2023), https://apps.autodesk.com/ACD/ru/Detail/Index?id=8957153203166034680&appLang=en&os=Win32\_64

5. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. Изд: Символ-Плюс, 2011, с.192 (3-е издание)