Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждения

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР»

Студент гр. 586-2

\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Девяшин

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Руководитель

К.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

2020

**РЕФЕРАТ**

Пояснительная записка: 21 стр, 14 рисунков, 10 источников.

AutoCAD 2020, ПЛАГИН, СВАЙНЫЙ АНГАР, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР.

Целью данной работы является разработка плагина, автоматизирующего построение модели «Ангар» для системы автоматизированного проектирования AutoCAD 2020 [1] с помощью интегрированной среды разработки Visual Studio 2019 Сommunity [2].

В процессе работы изучены основные параметры формы для выпечки и основные функции библиотек DLL из ObjectARX SDK[3] системы автоматизированного проектирования AutoCAD 2020.

В результате работы был создан плагин, автоматизирующий построение модели «Ангар».

Оглавление

[1 Введение 4](#_Toc40543429)

[2 Постановка и анализ задачи 5](#_Toc40543430)

[2.1 Описание предмета проектирования 5](#_Toc40543431)

[2.2 Выбор инструментов и средств реализации 6](#_Toc40543432)

[2.3 Назначение плагина 6](#_Toc40543433)

[2.4 Описание аналогов разрабатываемого продукта 7](#_Toc40543434)

[2.4.1 Описание ПК Лира 7](#_Toc40543435)

[3 Описание реализации 9](#_Toc40543436)

[3.1 Описание технических и функциональных аспектов проекта 9](#_Toc40543437)

[3.2 Диаграмма USE CASE 9](#_Toc40543438)

[3.2 Диаграмма классов 9](#_Toc40543439)

[3.3 Макеты пользовательского интерфейса 10](#_Toc40543440)

[3.4 Сравнение готового проекта и с дополненной функциональностью 12](#_Toc40543441)

[4 Тестирование программы 13](#_Toc40543442)

[4.1 Функциональное тестирование 13](#_Toc40543443)

[4.2 Модульное тестирование 14](#_Toc40543444)

[4.2 Нагрузочное тестирование 18](#_Toc40543445)

[Заключение 20](#_Toc40543446)

[Список использованных источников 21](#_Toc40543447)

# 1 Введение

Система автоматизированного проектирования — система, реализующая информационную технологию выполнения функций проектирования. Представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности.

Цели создания САПР:

• сокращение трудоёмкости проектирования и планирования;

• сокращение сроков проектирования;

• повышение качества и технико-экономического уровня результатов проектирования;

• сокращение затрат на натурное моделирование и испытания.

Задачи САПР:

• автоматизация оформления документации;

• информационная поддержка и автоматизация процесса принятия решений;

• использование технологий параллельного проектирования;

• унификация проектных решений и процессов проектирования;

• повторное использование проектных решений, данных и наработок;

• стратегическое проектирование;

•замена натурных испытаний и макетирования математическим моделированием;

• повышение качества управления проектированием;

• применение методов вариантного проектирования и оптимизации.

# 2 Постановка и анализ задачи

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируясь с системой AutoCAD 2020, строит модель. Необходимо чтобы плагин являлся библиотекой , запускаемой из AutoCAD, позволял задавать и очищать параметры, а также изменять их. К изменяемым параметрам относятся длина, ширина, высота, ангара, высота стены ангара. К зависимым параметрам относятся высота и ширина ворот. А также снеговые нагрузки.

## 2.1 Описание предмета проектирования

Ангар ­­ – сооружение для хранения, ремонта и технического обслуживания техники[4].

Параметры проектируемой 3D модели:

•высота ангара ­– H (от 2 до 10 м);

•длина ангара – L (от 2 до 40 м);

•ширина ангара – W (от 2 до 14 м);

•высота стен – l (от 0.1 до 0.3 м);

•снеговые нагрузки – S (от 200 до 600 кг/);

•высота ворот – h;

•ширина ворот – w;

•первый слой грунта – Q1;

•второй слой грунта – Q2;

•третий слой грунта – Q3.

Зависимые параметры 3D модели:

Ширина ворот определяется по формуле (2.1):

=< w <= (2.1)

где W – ширина ангара,

l – ширина ворот.

Высота ворот определяется по формуле (2.2):

2 м =< h <=, (2.2)

где H – высота ангара,

h – высота ворот.

Пример проектируемого изделия приведен ниже, на рисунке 2.1.

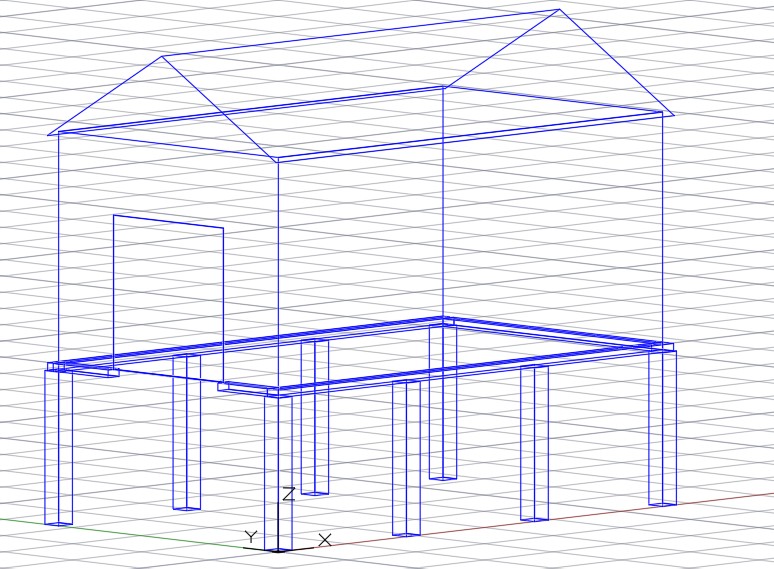


Рисунок 2.1 – Предмет изделия

## 2.2 Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2 [5].

Инструментом тестирования и создания модульных тестов был выбран фреймворк NUnit [6].

## 2.3 Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием ангара разных типов. Благодаря данному расширению, проектировщики могут наглядно увидеть спроектированную модель, при необходимости перестроить под необходимые им параметры.

## 2.4 Описание аналогов разрабатываемого продукта

### 2.4.1 Описание ПК Лира

ПК Лира — многофункциональный программный комплекс для проектирования и расчёта строительных и машиностроительных конструкций различного назначения. Реализация технологии информационного моделирования зданий.

ЛИРА-САПР реализует технологию информационного моделирования зданий (BIM) и ориентирована на проектирование и расчет строительных конструкций. Реализация технологии BIM обеспечивается нативной связью с другими архитектурными, расчетными, графическими и документирующими системами (САПФИР-3D, Revit, Tekla, AutoCAD, ArchiCAD, Advance Steel, BoCAD, Allplan, STARK ES, Gmsh и др.) на основе DXF, MDB, STP, SLI, MSH, STL, OBJ, IFC и др. файлов.

Развитая интуитивная графическая среда пользователя с возможностью 3D-визуализации расчетной схемы на всех этапах синтеза и анализа. Мощная система диагностики. Многочисленные виды представления результатов решения задачи — в графическом (изополя, эпюры, деформированные схемы, анимация форм колебаний) и табличном (перемещения, напряжения, усилия, РСУ, РСН, результаты подбора арматуры в железобетонных элементах и сечений стальных элементов) позволяет быстро провести необходимый анализ. Режим вариантного проектирования — в одной задаче пользователь может варьировать сечениями элементов, материалами, нормативами.

Интерфейс пользователя нового поколения. Синтез расчетной схемы здания или сооружения на основе управляемой процедуры преобразования 3D и 2D архитектурных моделей, созданных в различных графических программах: САПФИР-3D, Allplan, Revit, AutoCAD и др.

Развитая библиотека конечных элементов позволяет создавать компьютерные модели практически любых конструкций: плоских и пространственных рам, балок стенок, изгибаемых плит, оболочек, массивных тел, а также комбинированных систем - плит и оболочек подпертых ребрами, плит на грунтовом основании, каркасных конструкций зданий, системы "надземное строение - фундаментные конструкции - грунтовое основание" и мн. др.

Проверка и подбор сечений железобетонных и стальных элементов в соответствии с действующими в мире нормативами. Выполнение рабочих чертежей стадии КМ и КЖ.

Элементное моделирование с визуализацией на всех этапах расчета, позволяющее в ряде случаев ускорить решение задачи и снизить влияние плохой обусловленности большеразмерной матрицы.

Специализированный документатор позволяющий формировать отчет, состоящий из текстовой, табличной и графической информации . Режим интерактивных копий экрана позволяет осуществлять фиксацию и возврат к фрагменту расчетной схемы, а также выполнять автоматическое обновление изображений после ее изменения (перенумерация, перетриангуляция, смена конфигурации изображаемого объекта) [7].

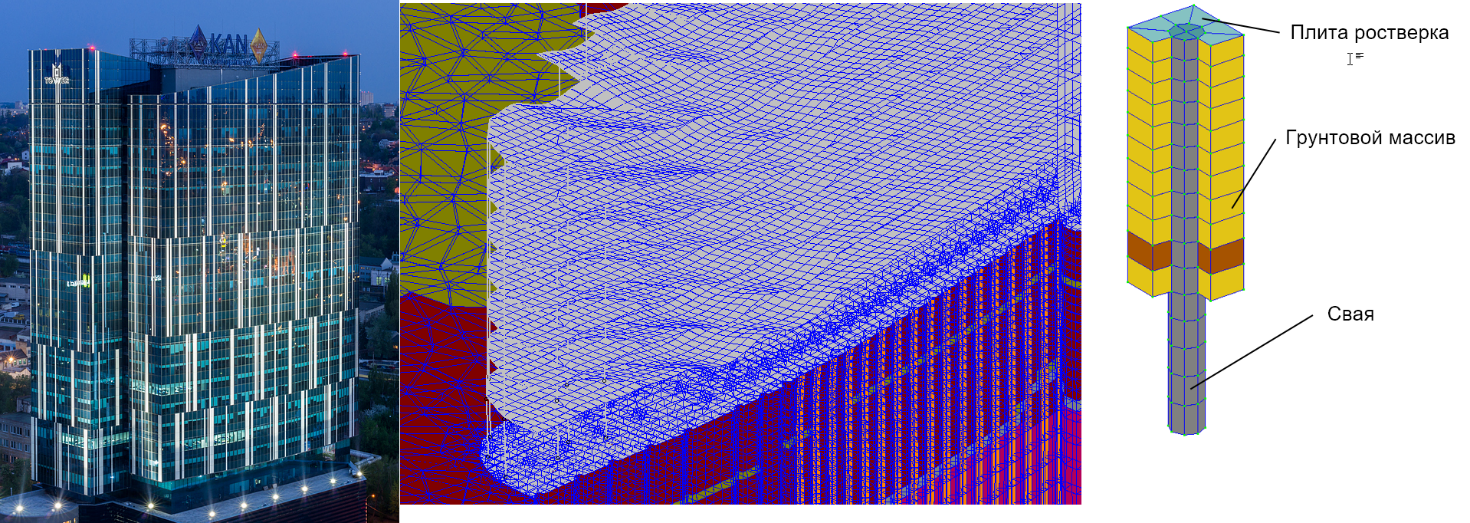
****

Рисунок 2.2 – Некоторые изображения из документации к ПК Лира.

# 3 Описание реализации

## 3.1 Описание технических и функциональных аспектов проекта

Язык UML (Unified Modeling Language) предназначен для описания, визуализации и документирования объектно-ориентированных систем и бизнес-процессов с ориентацией на их последующую реализацию в виде программного обеспечения.[8]

## 3.2 Диаграмма USE CASE

Диаграммы прецедентов составляют модель прецедентов (вариантов использования, use-cases). Прецедент — это функциональность системы, позволяющая пользователю получить некий значимый для него, ощутимый и измеримый результат[8]. Диаграмма представлена на рисунке 3.1.

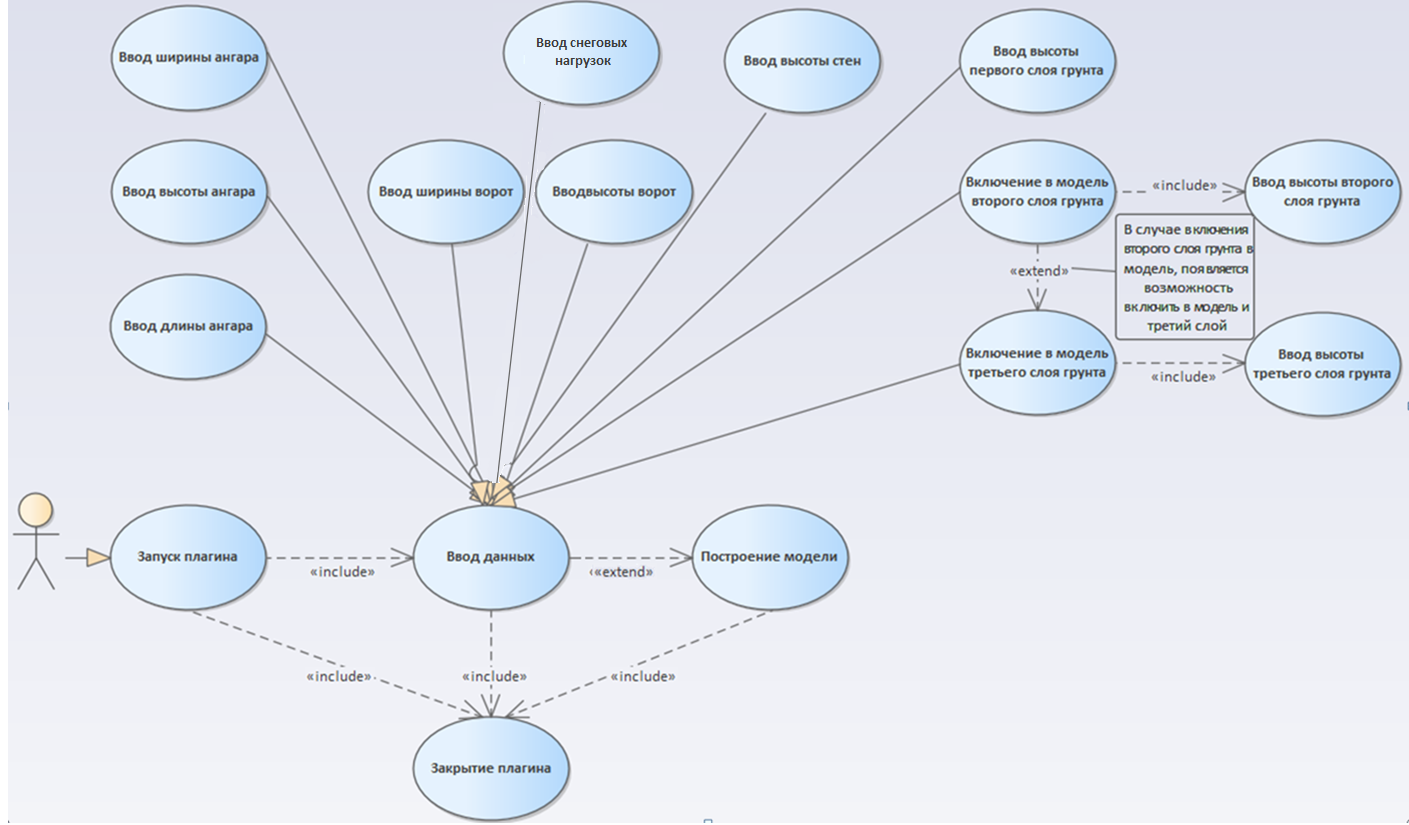


Рисунок 3.1 – Use Case диаграмма проекта.

## 3.2 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 [8]. Диаграмма классов представлена на рисунке 3.2.

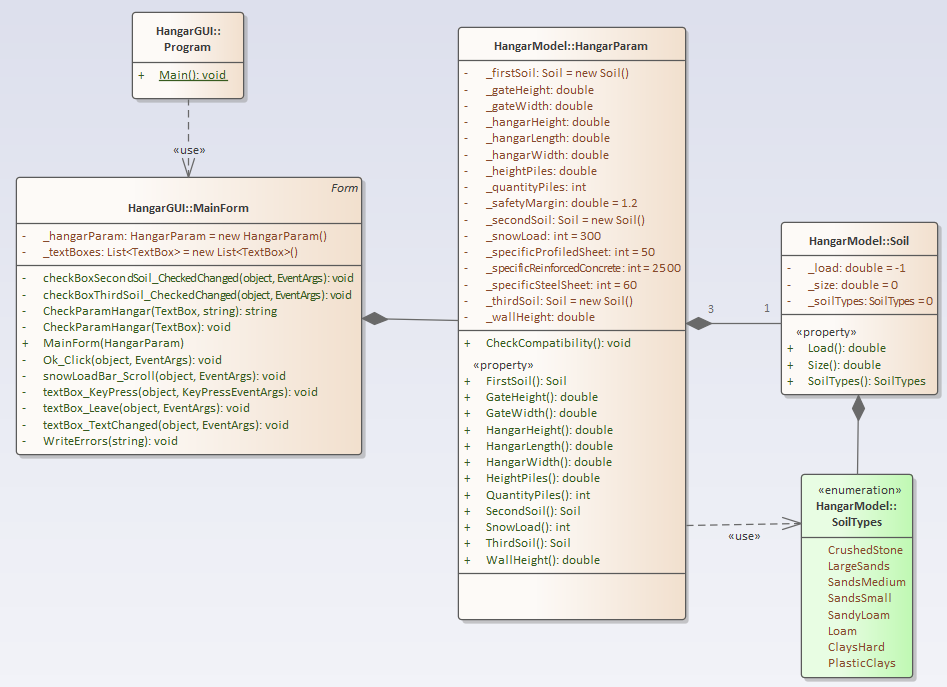


Рисунок 3.2 Диаграмма классов.

Класс Program запускает плагин из AutoCad и имеет функцию по построению модели.

Класс MainForm отображает на экран пользовательский интерфейс, передаёт введённые данные в класс HangarParam для проверки, а так же выводит сообщение об ошибках при вводе.

Класс HangarParam хранит и проверяет в себе введённые параметры ангара.

Класс Soil хранит и проверяет в себе параметры грунта.

Перечисление SoilTypes хранит в себе основные типы грунтов.

## 3.3 Макеты пользовательского интерфейса

Пользовательский интерфейс, или UI (User Interface) — это внешний вид продукта, способ общения между пользователем и программой.

Большую часть макета интерфейса составляет блок ввода. В блок ввода пользователю необходимо ввести желаемые параметры для 3D модели ангара. Кнопка ввода «Построить», при нажатии на которую, произойдет построение 3D модели по параметрам, введённым в поля блока ввода, в случае если данные оказались корректными. Шкала прогресса в свою очередь отображает прогресс заполненности блока ввода.

Все ошибки, допущенные при вводе, будут выведены в режиме реального времени. После нажатия кнопки «Ок», в случае ошибок, в диалоговом окне будут показаны допущенные ошибки. Так же изменится цвет фона поля с прозрачного на красный, в котором была допущена ошибка. Внешний вид пользовательского интерфейса изображен на рисунках 3.3 и 3.4

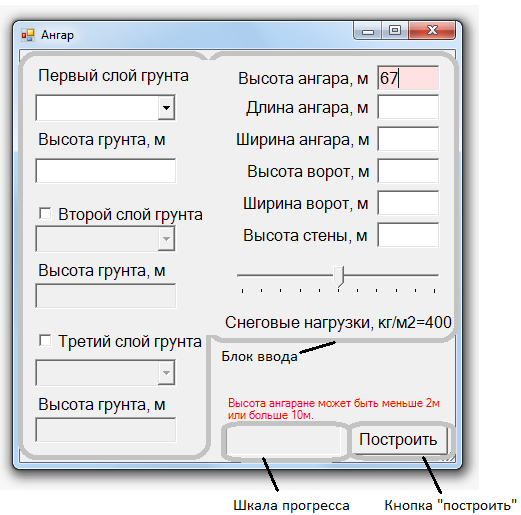


Рисунок 3.3 – Макет пользовательского интерфейса при вводе.

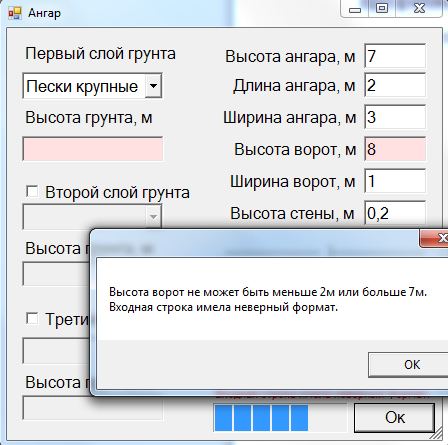


Рисунок 3.4 – Макет пользовательского интерфейса при нажатии кнопки.

## 3.4 Сравнение готового проекта и с дополненной функциональностью

Во время создания программы были внесены коррективы по введению дополнительной функциональности. Дополнительным параметром ввода были сезонные снеговые нагрузки, которые варьируются от 200 до 600 кг на метр квадратный. Сравнение интерфейсов до и после изменений представлены на рисунке 3.5.

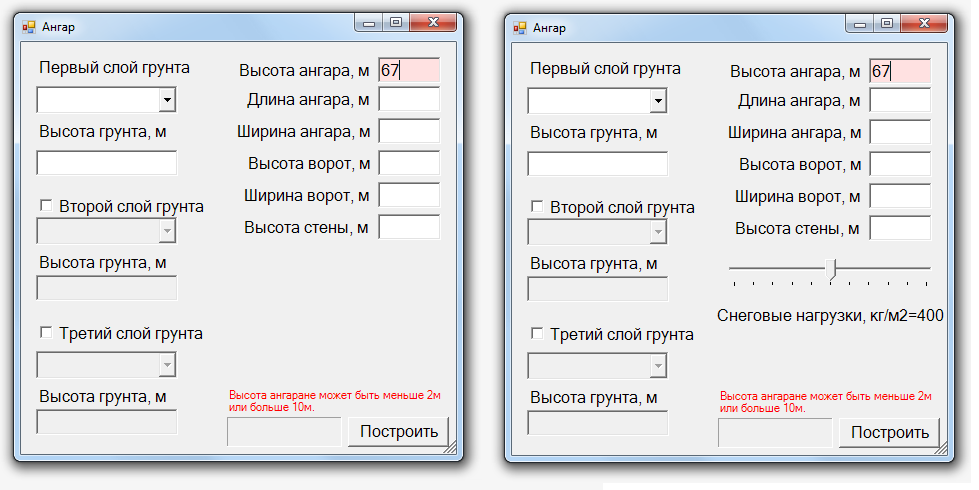


Рисунок 3.4 – Макет пользовательского интерфейса (слева без дополнительного параметра, справа с параметром).

Программно были добавлены поле \_snowLoad и свойство поля SnowLoad в класс HangarParam для хранения значения снеговой нагрузки, а также объект интерфейса TrackBar (snowLoadBar) и метод snowLoadBar\_Scroll, для изменения с помощью пользовательского интерфейса величины снеговой нагрузки.

# 4 Тестирование программы

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

## 4.1 Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Ангар», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведено тестирование максимальных и минимальных параметров модели.

На рисунке 4.1 представлена модель ангара с минимальным введенными параметрами.

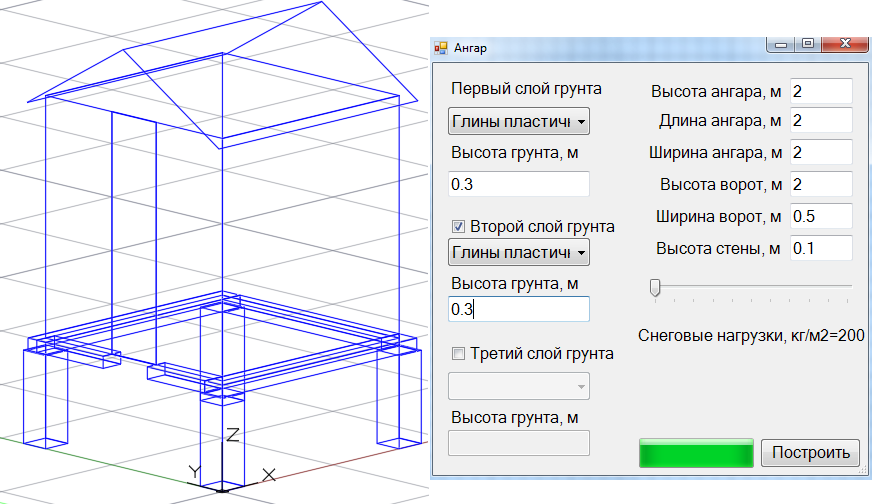


Рисунок 4.1 – Модель с минимальными веденными параметрами

На рисунке 4.2 представлена проверка размеров модели с максимальными введенными параметрами.

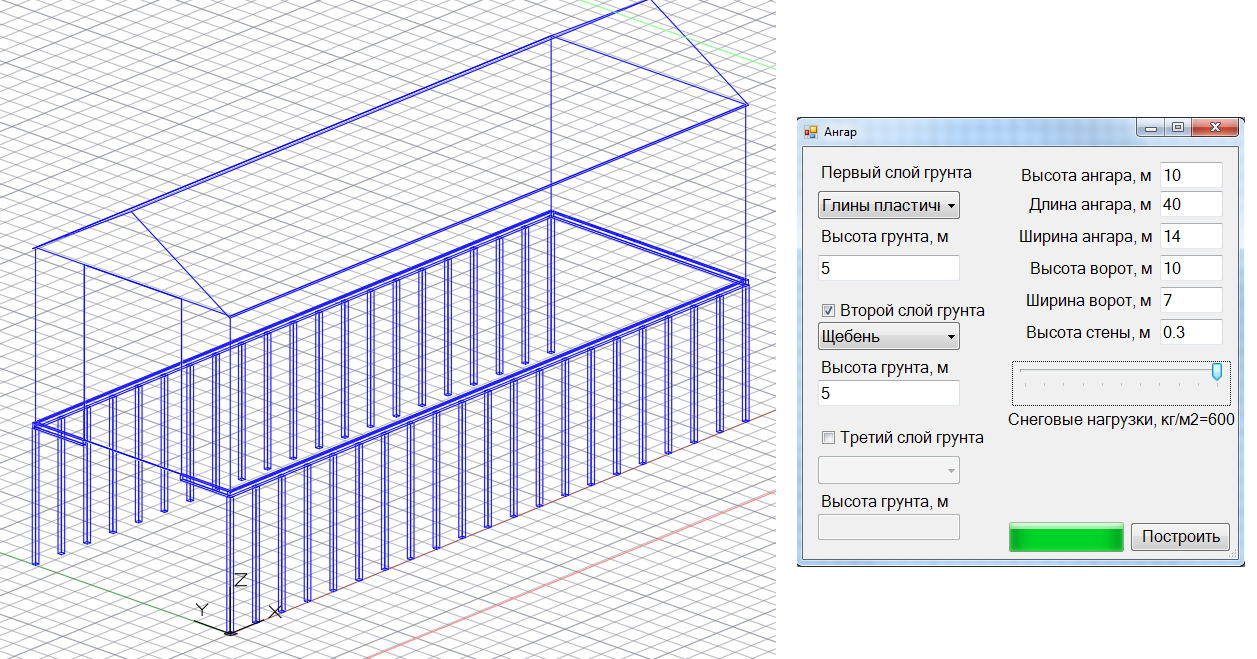


Рисунок 4.2 – Модель с максимальными веденными параметрами

## 4.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи обозревателя тестов Visual Studio было проведено модульное тестирование [9], проверялись открытые поля и методы, для этого были созданы тестовые классы:

1. HangarUnitTests – Класс для проверки открытых свойств класса HangarParam (Таблица 4.1)
2. SoilUnitTests – Класс для проверки открытых свойств класса Soil (Таблица 4.2)

Ниже представлено диалоговое окно состояний запущенных тестов класса HangarUnitTests (Рисунок 4.3)

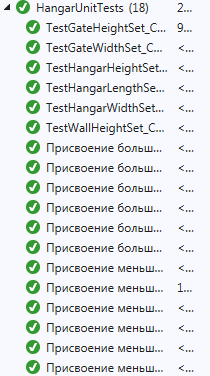


Рисунок 4.3 – Диалоговое окно состояний запущенных тестов класса HangarUnitTests

Таблица 4.1 – Описание тестов класса HangarUnitTests.

|  |  |  |
| --- | --- | --- |
| Название | Описание | Входной пар-тр |
| TestHangarHeightSet\_CorrectValue | Проверка на правильное значение высоты ангара | 8 |
| TestHangarWidthSet\_CorrectValue | Проверка на правильное значение ширины ангара | 4 |
| TestHangarLengthSet\_CorrectValue | Проверка на правильное значение длины ангара | 14 |
| TestWallHeightSet\_CorrectValue | Проверка на правильное значение высоты стены | 0.3 |
| TestGateHeightSet\_CorrectValue | Проверка на правильное значение высоты ворот | 3 |
| TestGateWidthSet\_CorrectValue | Проверка на правильное значение ширины ворот | 2 |
| TestHangarHeightSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Высота ангара | 0 |

Окончание таблицы 4.1.

|  |  |  |
| --- | --- | --- |
| TestHangarHeightSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Высота ангара | 1000 |
| TestHangarWidthSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Ширина ангара | 0 |
| TestHangarWidthSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Ширина ангара | 1000 |
| TestHangarLengthSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Длина ангара | 0 |
| TestHangarLengthSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Длина ангара | 1000 |
| TestWallHeighttSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Высота стены | 0 |
| TestWallHeighttSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Высота стены | 1000 |
| TestGateHeightSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Высота ворот | 0 |
| TestGateHeightSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Высота ворот | 1000 |
| TestGateWidthSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Ширина ворот | 0 |
| TestGateWidthSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Ширина ворот | 1000 |

Ниже представлено диалоговое окно состояний запущенных тестов класса SoilUnitTests (Рисунок 4.4)

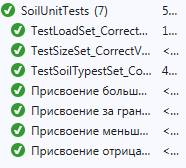


Рисунок 4.4 – Диалоговое окно состояний запущенных тестов класса SoilUnitTests

Таблица 4.2 – Описание тестов класса HangarUnitTests.

|  |  |  |
| --- | --- | --- |
| Название | Описание | Входной пар-тр |
| TestSoilTypestSet\_CorrectValue | Проверка на правильное значение типа слоя | 0 |
| TestSizeSet\_CorrectValue | Проверка на правильное значение высоты слоя | 1 |
| TestLoadSet\_CorrectValue | Проверка на правильное значение нагрузки грунта | 5 |
| TestSoilTypestSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - типа грунта | -1 |
| TestSoilTypestSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - типа грунта | 8 |
| TestSizetSet\_ArgumentException | Проверка на исключение, если число больше граничных значений - Высоты слоя | -1 |
| TestSizetSet\_ArgumentException | Проверка на исключение, если число меньше граничных значений - Высоты слоя | 8 |

## 4.2 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [10]. Для проведения нагрузочного тестирования был добавлен цикл, который просчитывал и строил один и тот же ангар. Измерялись потребляемые ресурсы физической памяти и время построения модели.

Тестирование проводилось на ПК следующей конфигурации:

Процессор: Intel(R) i5-4590;

ОЗУ: 8 Гб;

Тип системы: х64.

На рисунке 4.5 представлен график зависимости занимаемой памяти, от количества построенных моделей в одном оке AutoCAD.

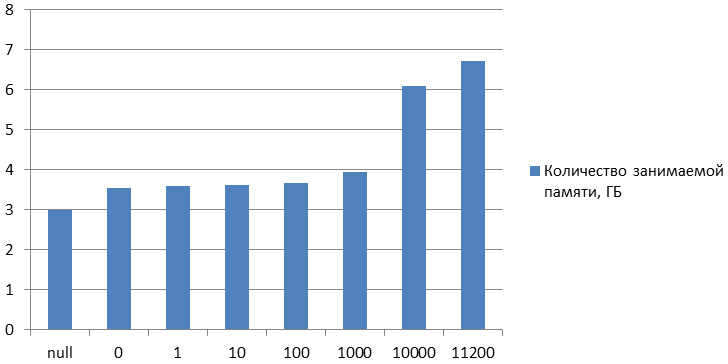
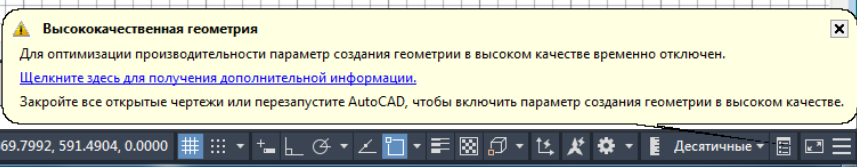


Рисунок 4.5 – График зависимости занимаемой памяти (вертикальная ось) от количества построенных моделей (горизонтальная ось)

Можно заметить, как память занимается не пропорционально количеству моделей. Это связано с тем, что с ростом числа блоков в пространстве, сокращается потребление памяти на графическую составляющую (Рисунок 4.6).



На рисунке 4.6 – Сообщение программы после пятидесятой итерации создания модели.

На рисунке 4.7 представлен график зависимости времени работы программы от количества построенных моделей в одном оке AutoCAD.

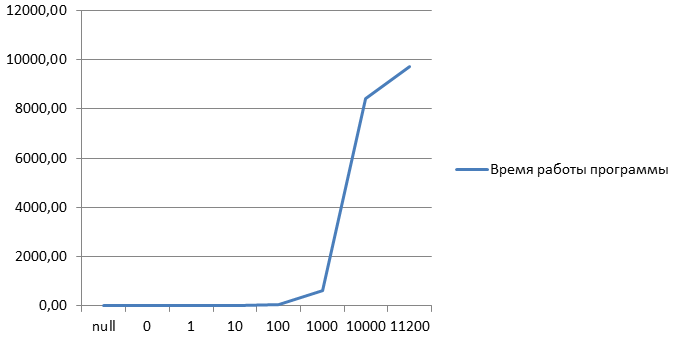


Рисунок 4.7 – График зависимости времени работы программы (вертикальная ось) от количества построенных моделей (горизонтальная ось)

Скорость построения одной модели заняло около 3 секунд, сотня моделей создалось в AutoCad за 7.3 минут, а построение 10 000 моделей заняло более двух часов.

На 11 200-о й модели AutoCad перестал отвечать на запросы и завершил работу.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D моделей «Ангар» в САПР AutoCAD 2020.

# Список использованных источников

1. Autodesk AutoCAD — краткий обзор возможностей программы, плюсы практической работы в ней. [Электронный ресурс]. — URL: <https://cgschool.pro/base/baza-1/> (дата обращения 28.01.2020)
2. Visual Studio [Электронный ресурс]. — URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 20.04.2020);
3. AutoCAD .NET Developer's Guide [Электронный ресурс]. — URL: <http://docs.autodesk.com/ACD/2010/ENU/AutoCAD%20.NET%20Developer%27s%20Guide/index.html> (дата обращения 25.01.2020)
4. Новая иллюстрированная энциклопедия. Кн. 1 Аа – Ар — М.: Большая Российская энциклопедия, 2004.
5. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 20.04.2020);
6. NUnit [Электронный ресурс]. – URL: <https://nunit.org/> (дата обращения: 20.04.2020);
7. ООО "Лира сервис" [Электронный ресурс]. — URL: <https://www.liraland.ru/lira> (дата обращения 25.01.2020)
8. Самоучитель UML, Александр Леоненков [Электронный ресурс]. – URL: <http://kaf401.rloc.ru/TRPO/LeonenkovUML.pdf> (дата обращения 9.04.2020)
9. Модульное тестирование [Электронный ресурс]. – URL: <http://espressocode.top/unit-testing-software-testing/> (дата обращения: 13.04.2020);
10. Нагрузочное тестирование [Электронный ресурс]. – URL: <https://daglab.ru/nagruzochnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 13.04.2020).