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)

**РАЗРАБОТКА ПЛАГИНА «ПОЖАРНАЯ ЛЕСТНИЦА» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
по дисциплине  
«Основы разработки САПР» (ОРСАПР)

Выполнил:  
студент гр. 580-2  
\_\_\_\_\_\_\_\_\_\_\_ Емельянов Р.А.  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Руководитель:  
к.т.н., доцент каф. КСУП  
\_\_\_\_\_\_\_\_\_\_\_ Калентьев А.А.  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**Содержание**

[Введение 3](#_Toc155940943)

[1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_Toc155940944)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 5](#_Toc155940945)

[3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 7](#_Toc155940946)

[4 НАЗНАЧЕНИЕ ПЛАГИНА 9](#_Toc155940947)

[5 ОБЗОР АНАЛОГОВ 10](#_Toc155940948)

[6 ОПИСАНИЕ РЕАЛИЗАЦИИ 13](#_Toc155940949)

[7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 20](#_Toc155940950)

[8 ТЕСТИРОВАНИЕ ПЛАГИНА 23](#_Toc155940951)

[8.1 Функциональное тестирование 23](#_Toc155940952)

[8.2 Модульное тестирование 26](#_Toc155940953)

[8.3 Нагрузочное тестирование 29](#_Toc155940954)

[Заключение 31](#_Toc155940955)

[Список использованных источников 32](#_Toc155940956)

# Введение

Автоматизация моделирования играет ключевую роль в современной инженерной и научной практике. Благодаря использованию специализированных программ и инструментов, инженеры, ученые и дизайнеры могут создавать сложные модели, проводить анализ данных и прогнозировать поведение систем в автоматическом режиме. Это не только ускоряет процесс моделирования, но и повышает точность результатов.

Создание пожарных лестниц – важный аспект безопасности зданий, требующий тщательного проектирования и соответствия строгим нормативам. С развитием технологий автоматизация этого процесса становится все более значимой. Автоматизированные инструменты позволяют ускорить проектирование, улучшить точность расчетов и обеспечить соответствие лестниц нормативам безопасности.

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение пожарной лестницы для системы автоматизированного проектирования КОМПАС-3D с помощью интегрированной среды разработки Visual Studio 2022.

# 1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

Главной целью выполнения работы является разработка плагина “Пожарная лестница” для системы автоматизированного проектирования КОМПАС-3D.

Пожарная лестница имеет следующие изменяемые параметры: высота, ширина и толщина лестницы, ширина балки, длина и высота ступени, расстояние между ступенями. Ширина лестницы и балки и длина ступени являются зависимыми. Например, если изменить длину ступени на определенное значение, то и ширина лестница так же должна измениться на это же значение.

Разрабатываемый плагин должен обеспечивать обработку некорректных значений, вводимых пользователем. При вводе некорректного значения в поле, это поле должно поменять свой цвет на ярко-розовый. При попытке пользователя нажать на кнопку “Построить”, должно появиться окно с ошибкой, в котором будет указано, какие поля формы заполнены неверно и какие значения допустимы для них.

В рамках проекта были поставлены следующие задачи:

1. Разработка технического задания (07.10.2023г.);
2. Разработка проекта системы (21.10.2023г.);
3. Реализация плагина (15.11.2023г.);
4. Доработка плагина, написание пояснительной записки (29.12.2023г.).

В процессе анализа задач были определены некоторые возможные проблемы, которые могут возникнуть при разработке плагина:

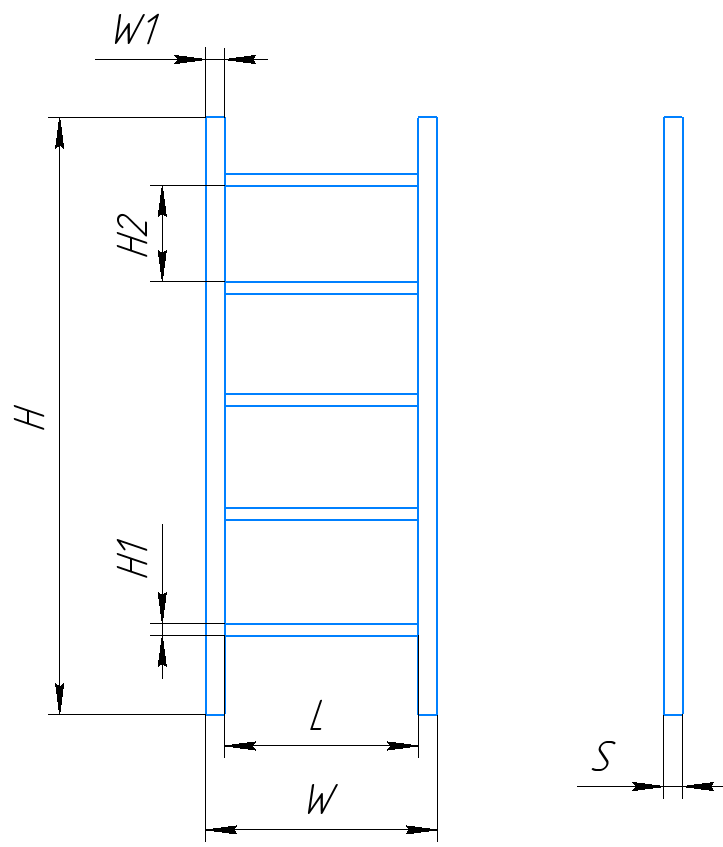
* Определение количества ступеней лестницы при изменении ее высоты;
* Возникновение непредвиденных ошибок и исключений связанных с API КОМПАС-3D;
* Одновременная разработка нескольких исключений.

# 2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Пожарная лестница – стационарно закрепленная к стене здания лестница, предназначенная для:

* эвакуации людей при пожаре;
* подъема личного состава и пожарно-технического вооружения на этажи, кровлю здания и сооружения.

Для данного плагина была выбрана пожарная лестница вертикального типа, конструктивно состоящая их двух параллельных вертикальных направляющих, жестко соединенных поперечными опорными ступенями. На рисунке 2.1 представлен чертеж пожарной лестницы.



а) б)

Рисунок 2.1 – Модель пожарной лестницы вид спереди (а) и сбоку (б)

Изменяемые параметры для плагина (также все обозначения показаны на рис. 2.1):

* высота лестницы H (1000мм – 20000мм);
* ширина лестницы W (190мм – 1000мм);
* толщина лестницы S (20мм – 50мм);
* ширина бруска W1 (20мм – 50мм);
* длина ступени L (150мм – 960мм);
* высота ступени H1 (20мм – 40мм);
* расстояние между ступенями H2 (250мм – 350мм);
* при изменении ширины лестницы W изменяется длина ступени L (L = W – 2 ∙ W1).

В качестве дополнительной функциональности была поставлена задача добавления возможности изменения высоты ступени в пределах от 20мм до 40мм и изменения расстояния между ступенями от 250мм до 350мм.

# 3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ

В качестве платформы для разработки плагина был выбран WinForms с использованием рабочей среды .Net Framework 4.8.

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений [[1](https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0)].

Также в процессе реализации плагина были использованы следующие библиотеки и инструменты:

* ReSharper 2023.3;
* StyleCop 1.1.188;
* SpellChecker 2023.12.29.0;
* Editor Guidelines 2.2.10;
* Fine Code Coverage 1.1.197;
* NUnit 4.0.1;
* NUnit3TestAdapter 4.5.0.

ReSharper – расширение для Visual Studio, анализирующее качество кода, помогающее устранять ошибки и проблемы в структуре кода, осуществляющее мгновенный поиск и навигацию по всему решению и многое другое [[2](https://www.jetbrains.com/ru-ru/resharper/)].

StyleCop – утилита с открытым исходным кодом, выполняющая статический анализ файлов исходного кода на C# [[3](https://learn.microsoft.com/ru-ru/archive/msdn-magazine/2013/october/alm-rangers-implementing-static-code-analysis-with-stylecop)].

SpellChecker – утилита, которая проверяет правописание строк кода, комментариев и другой текстовой информации. Данной утилите необходимы словари, с помощью которых она будет определять слова, в которых была допущена ошибка.

Editor Guidelines – расширение, которое добавляет вертикальную линию помогающую ограничить ширину строк.

Fine Code Coverage – расширение, которое визуализирует покрытие кода модульными тестами.

NUnit – библиотека, предназначенная для написания юнит-тестов приложений для платформы .NET.

NUnit3TestAdapter – дополнительный пакет, позволяющий запускать юнит-тесты в среде Visual Studio. Без данного пакета запуск юнит-тестов придется выполнять в отдельном консольном приложении.

# 4 НАЗНАЧЕНИЕ ПЛАГИНА

Назначение разрабатываемого плагина обусловлено быстрым моделированием пожарных лестниц с разными параметрами. С помощью данного плагина специалисты по пожарной безопасности могут наглядно рассмотреть проектируемую модель пожарной лестницы и при необходимости перестроить ее под необходимые параметры. На рисунке 2.1 представлена модель пожарной лестницы.

# 5 ОБЗОР АНАЛОГОВ

Прямым аналогом является пакет плагинов по “серии 1.450.3. Лестницы и ограждения Tekla” [[4](https://sapr-art.by/s1450)], в который входит плагин, предназначенный для моделирования стремянки по серии “1.450.3-7.94 выпуск 2. Конструкции из горячекатаных профилей”. Основными данного плагина являются четыре вкладки “Стремянка”, “Ограждение”, “Узлы”, “Детали”. Дальнейшее описание плагина основывалось руководстве для него [[5](https://docs.google.com/document/d/1DXjKSFF1_5HtommXsSB5qXpN2p4rtVMhjUq8-GVRBEM/edit#heading=h.s78n08hb4lp7)].

На вкладке “Стремянка” можно выбрать марку стремянки, наличие ступеней, шаг ступеней, болты для нижней крепежной пластины, общую высоту стремянки и др. Интерфейс вкладки “Стремянка” плагина “Стремянка” представлен на рисунке 5.1.

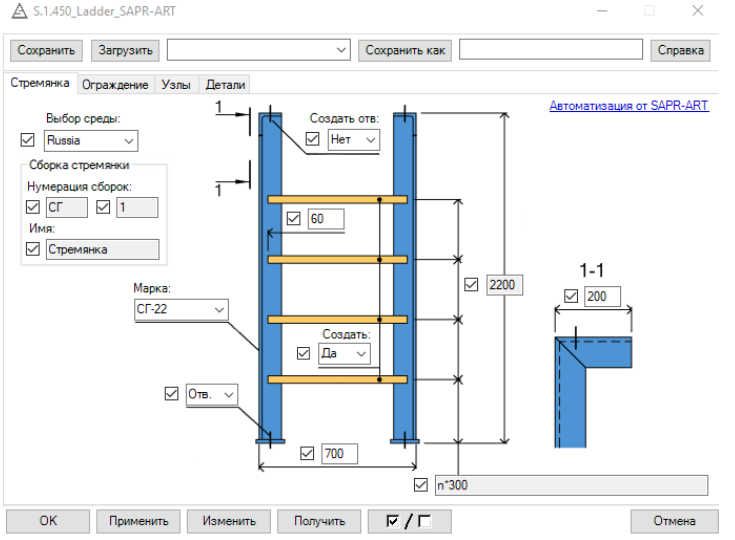


Рисунок 5.1 – Интерфейс вкладки “Стремянка” плагина “Стремянка”

На вкладке “Ограждение” можно выбрать марку ограждения, длину ограждения, шаг секций, конфигурацию и др. Интерфейс вкладки “Ограждение” плагина “Стремянка” представлен на рисунке 5.2.

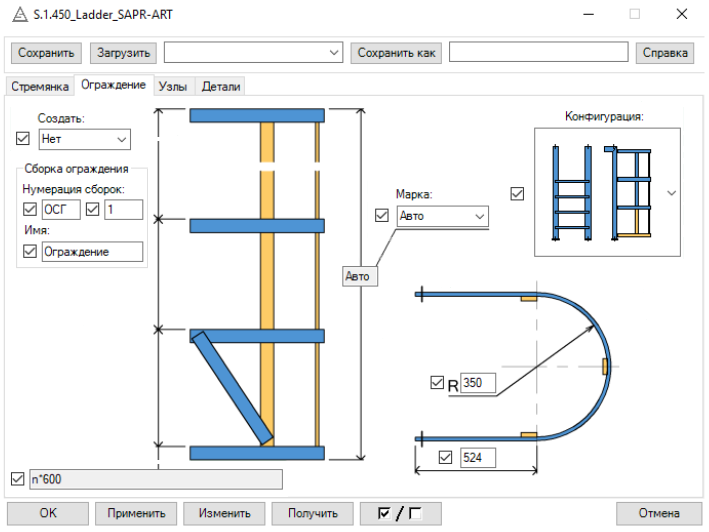


Рисунок 5.2 – Интерфейс вкладки “Ограждение” плагина “Стремянка”

На вкладке “Узлы” можно выбрать размеры отверстий или болтов с допусками. Интерфейс вкладки “Узлы” плагина “Стремянка” представлен на рисунке 5.3.

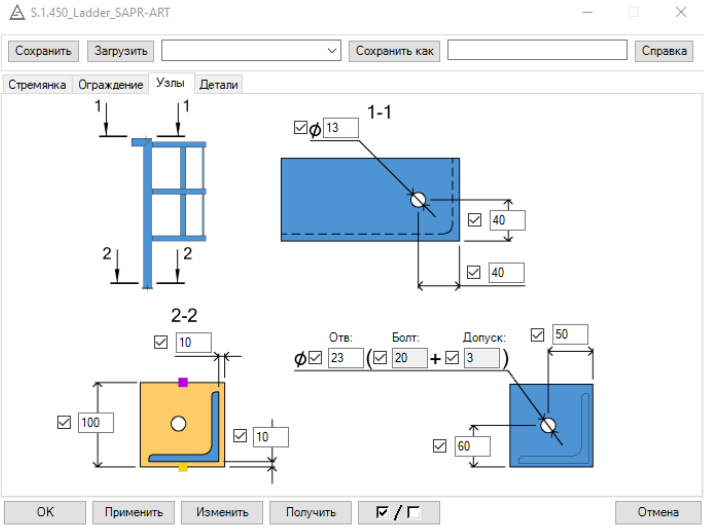


Рисунок 5.3 – Интерфейс вкладки “Узлы” плагина “Стремянка”

Вкладка “Детали” предназначена для задания детальных настроек элементов плагина. В этой вкладке доступна детальная настройка болтов как для стремянки, так и для ее ограждения. Интерфейс вкладки “Детали” плагина “Стремянка” представлен на рисунке 5.4.

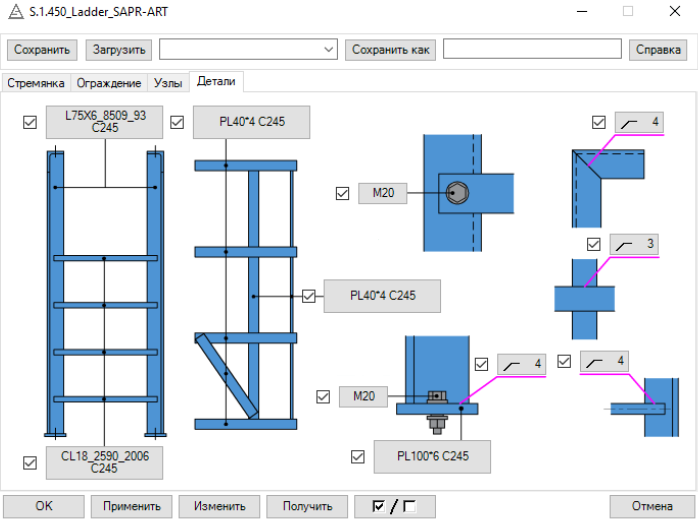


Рисунок 5.4 – Интерфейс вкладки “Детали” плагина “Стремянка”

# 6 ОПИСАНИЕ РЕАЛИЗАЦИИ

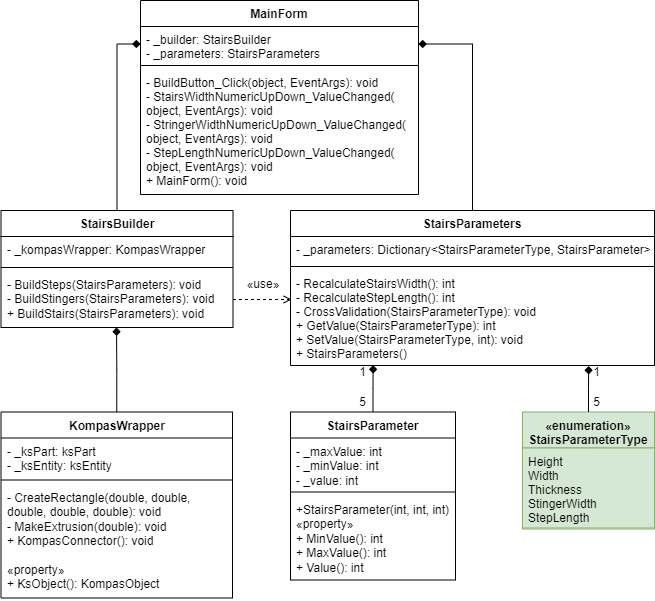


Рисунок 6.1 – UML диаграмма классов до реализации программы

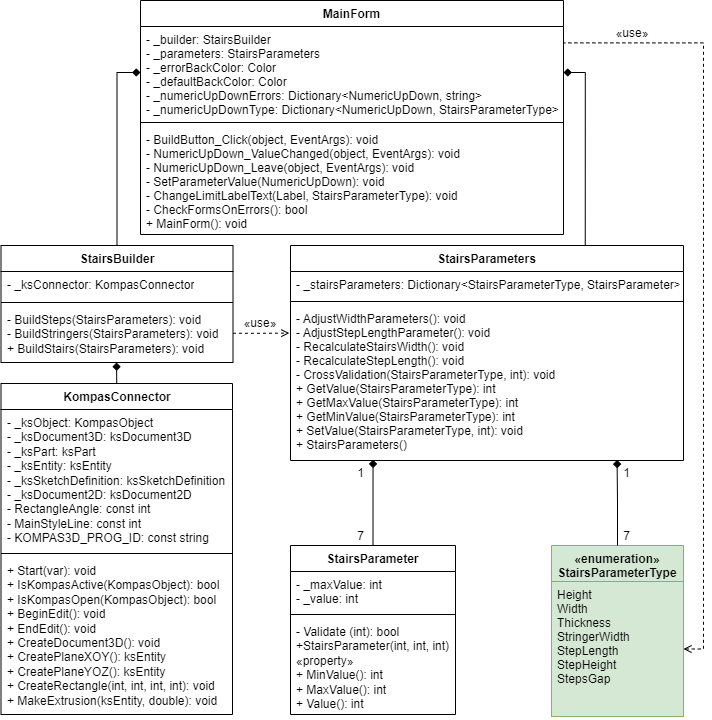


Рисунок 6.2 – UML диаграмма классов после реализации программы

Изменения, которые были сделаны по ходу реализации плагина:

* В классе “StairsParameter” было убрано поле “\_minValue” и оставлено только авто свойство “MinValue”. Данное изменение было сделано в связи с тем, что для минимального значения не проводится никаких проверок.
* В классе “StairsParameters” были добавлены методы “GetMaxValue” и “GetMinValue”, чтобы можно было обращаться к минимальным и максимальным значениям параметров лестницы из класса “MainForm”. Из метода “SetValue” было выделено два метода “AdjsutWidthParameters” и “AdjustStepLengthParameter” для лучшей читаемости кода.
* В классе “KompasConnector” были добавлены следующие поля: “\_ksDocument3D”, “\_ksSketchDefinition”, “\_ksDocument2D”, “RectangleArea”, “MainStyleLine”, “KOMPAS3D\_PROG\_ID”, а авто свойство “KsObject” было заменено на поле “\_ksObject”. Метод “KomapsConnector” был разделен на три метода: “Start”, “IsKompasActive” и “IsKompasOpen”. Также были добавлены методы “BeginEdit”, “EndEdit”, “CreateDocument3D”, “CreatePlaneXOY”, “CreatePlaneYOZ”. Для метода “CreateRectangle” был изменен тип данных входных параметров и убран один входной параметр, отвечающий за угол (теперь значение угла перемещено в константу). Такие большие изменения в данном классе могут быть вызваны тем, что работа с API КОМАС-3D производилась в первый раз и не все удалось продумать на этапе проекта системы.
* В классе “MainForm” были добавлены поля с цветом ошибки и обычным, словари ошибок и элементов управления “NumericUpDown”. Также был оставлен один обработчик при изменении значения в элементе управления “NumericUpDown”, добавлен обработчик “NumericUpDown\_Leave”, метод “SetParameterValue”, “ChangeLimitLabelText”, которые были добавлены для сокращения дублирования кода. Метод “CheckFormOnErrors” был добавлен для компиляции текста ошибок в одно сообщение.

Далее приведем описание полей и методов класса в виде таблиц.

Таблица 6.1 – Поля класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | StarisBuilder | Экземпляр класса StairsBuilder |
| \_parameters | StairsParameters | Экземпляр класса StairParametrs |
| \_errorBackColor | Color | Цвет ошибки |
| \_defaultBackColor | Color | Стандартный цвет |
| \_numeric UpDownErrors | Dictionary<Numeric UpDown, string> | Словарь ошибок |
| \_numeric UpDownType | Dictionary<Numeric UpDown, StarisParameter Type> | Словарь элементов NumericUpDown по типу параметра лестницы |

Таблица 6.2 – Методы и обработчики класса MainForm

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходной параметр | Описание |
| BuildButton\_Click | object, EventArgs | void | Обработчик нажатия на кнопку BuildButton |
| NumericUpDown\_ ValueChanged |  | void | Обработчик изменения значения в элементе NumericUpDown |
| NumericUpDown\_ Leave |  | void | Обработчик покидания элемента NumericUpDown |
| SetParameterValue |  | void | Задает значение зависимого параметра лестницы. |
| ChangeLimitLabel Text |  | void | Изменяет надписи лимитов на форме. |
| CheckFormsOn Errors |  | bool | Проверяет NumericUpDown элементы на ошибки заполнения. |
| MainForm |  | – | Конструктор формы |

Таблица 6.3 – Поля класса StairsBuilder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_ksConnector | KompasConnector | Экземпляр класса KompasConnector |

Таблица 6.4 – Методы класса StairsBuilder

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходной параметр | Описание |
| BuildSteps | StairsParameters | void | Строит ступени |
| BuildStringers | StairsParameters | void | Строит балки |
| BuildStairs | StairsParameters | void | Строит лестницу |

Таблица 6.5 – Поля класса StairsParameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_stairsParameters | Dictionary<StarisParameterType, StairsParameter> | Словарь, содержащий параметры лестницы. |

Таблица 6.6 – Методы класса StairsParameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходной параметр | Описание |
| AdjustWidth Parameter | – | void | Устанавливает длину ступени и ширину балки |
| AdjustStepLength Parameter | – | void | Устанавливает ширину лестницы |
| RecalculateStairs Width | – | void | Пересчитывает ширину лестницы |
| RecalculateStep Length | – | void | Пересчитывает длину ступени |
| CrossValidation | StairsParameterType, int | void | Проверяет зависимые параметры |
| GetValue | StairsParameterType | int | Возвращает значение параметра из словаря |
| GetMaxValue | StairsParameterType | int | Возвращает максимальное значение параметра из словаря |
| GetMinValue | StairsParameterType | int | Возвращает минимальное значение параметра из словаря |
| SetValue | StairsParameterType | void | Задает значение для параметра в словаре |
| StairsParameters | – | Stairs Parameters | Создает экземпляр класса StairsParameters |

Таблица 6.7 – Поля класса KompasConnector

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_ksObject | KompasObject | Объект интерфейса API КОМПАС-3D |
| \_ksDocument3D | ksDocument3D | Указатель на интерфейс документа |
| \_ksPart | ksPart | Указатель на интерфейс компонента |
| \_ksEntity | ksEntity | Указатель на интерфейс сущности |
| \_ksSketchDefinition | ksSketchDefinition | Указатель на интерфейс параметров эскиза |

Продолжение таблицы 6.7.

|  |  |  |
| --- | --- | --- |
| \_ksDocument2D | ksDocument2D | Указатель на эскиз |
| RectangleAngle | const int | Угол прямоугольника |
| MainStyleLine | const int | Стиль линии |
| KOMPAS3D\_PROG\_ID | const string | Строковое наименование идентификатора COM-объекта |

Таблица 6.8 – Методы класса KompasConnector

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходной параметр | Описание |
| Start |  |  | Запускает КОМПАС-3D |
| IsKompasActive | KompasObject | bool | Проверяет состояние КОМПАС-3D |
| IsKompasOpen | KompasObject | bool | Проверяет, является ли КОМПАС-3D открытым |
| BeginEdit | – | void | Начать редактирование |
| EndEdit | – | void | Закончить редактирование |
| CreateDocument3D | – | void | Запускает окно создания 3D-модели |
| CreatePlaneXOY | – | ksEntity | Выбирает плоскость XOY для эскиза |
| CreatePlaneYOZ | – | ksEntity | Выбирает плоскость YOZ для эскиза |
| CreateRectangle | int, int, int, int | void | Создает прямоугольник |
| MakeExtrusion | ksEntity, double | void | Делает выдавливание эскиза |

Таблица 6.9 – Поля класса StairsParameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_maxValue | int | Максимальное значение параметра |
| \_value | int | Текущее значение параметра |

Таблица 6.10 – Методы и свойства класса StairsParameter

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Выходной параметр | Описание |
| Validate | int | bool | Проверка текущего значения параметра на принадлежность к диапазону допустимых значений |
| StairsParameter | int, int, int | – | Создает экземпляр класса StairsParameter |
| MinValue | – | int | Задает или возвращает минимальное значение параметра |
| MaxValue | – | int | Задает или возвращает максимальное значение параметра |
| Value | – | int | Задает или возвращает текущее значение параметра |

Таблица 6.11 – Перечисления StairsParameterType

|  |  |
| --- | --- |
| Название | Описание |
| Height | Высота лестницы |
| Width | Ширина лестницы |
| Thickness | Толщина лестницы |
| StringerWidth | Ширина балки |
| StepLength | Длина ступени |
| StepHeight | Высота ступени |
| StepsGap | Расстояние между ступенями |

# 7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ

При запуске плагина перед пользователем появляется форма, с которой он может взаимодействовать посредством ввода значений в элементы управления и нажатия на кнопку “Построить”. Пользовательский интерфейс плагина представлен на рисунке 7.1. На рисунке 7.2 представлен пользовательский интерфейс с обозначенными областями.

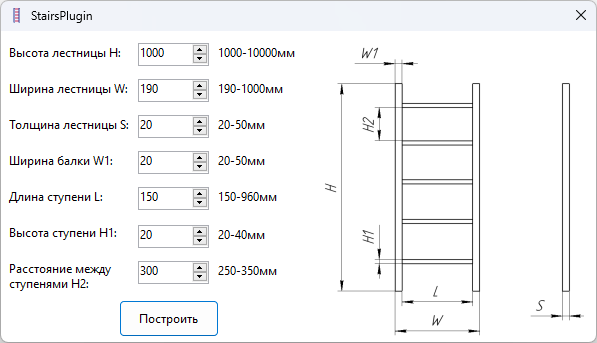
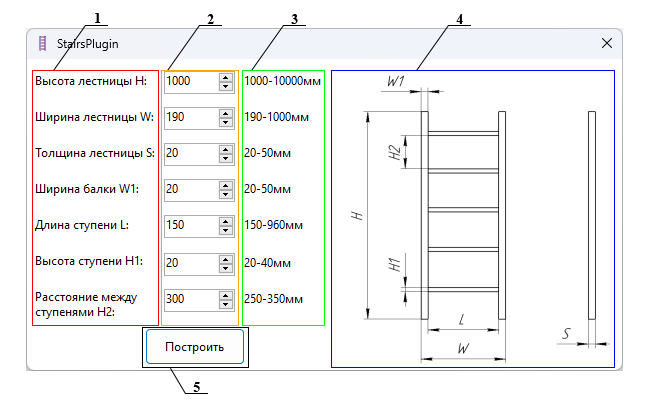


Рисунок 7.1 – Пользовательский интерфейс



1 – название параметров; 2 – поля для ввода значений; 3 – ограничения параметров; 4 – схематическое изображение лестницы; 5 – кнопка для построения модели в САПР   
Рисунок 7.2 – Пользовательский интерфейс с обозначенными областями

При попытке ввести пользователем значения, которые не входят в область допустимых значений, в элементы управления, это поле поменяет свой цвет на красный (рис. 7.3). Если пользователь проигнорирует цветовую индикацию ошибки, то при нажатии на кнопку построить появится окно с ошибкой, в котором будет указано, в каком поле (полях) была допущена ошибка и укажет корректные значения для поля (полей) (рис. 7.4).

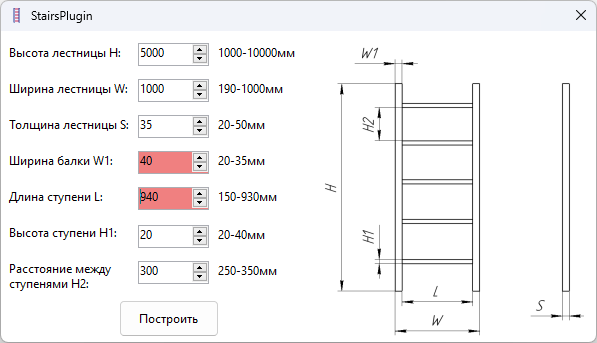


Рисунок 7.3 – Пользовательский интерфейс с неверно введенными значениями

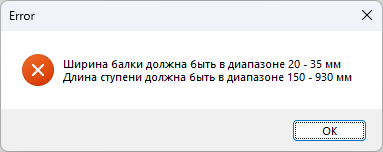


Рисунок 7.4 – Сообщение об ошибке

На рисунках 7.5 и 7.6 представлен результат работы плагина при корректно введенных значениях параметров.

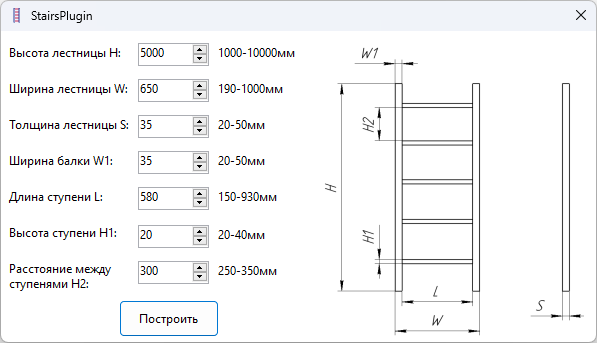


Рисунок 7.5 – Пользовательский интерфейс с верными значениями



Рисунок 7.6 – Результат работы плагина

# 8 ТЕСТИРОВАНИЕ ПЛАГИНА

## 8.1 Функциональное тестирование

На рисунках 8.1 – 8.4 продемонстрирована работа плагина при некорректных значениях.

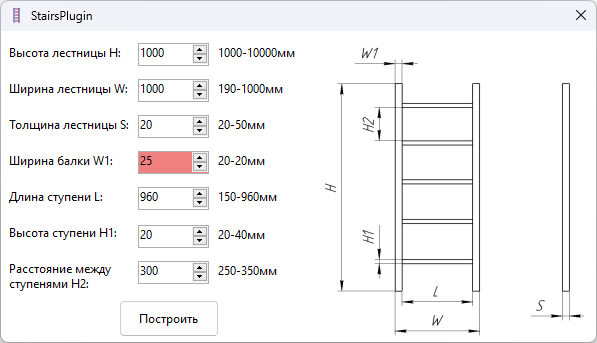


Рисунок 8.1 – Вид пользовательского интерфейса при одной ошибке

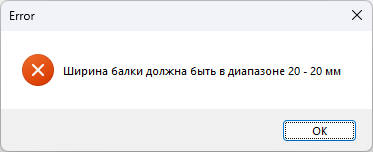


Рисунок 8.2 – Содержание сообщения при одной ошибке

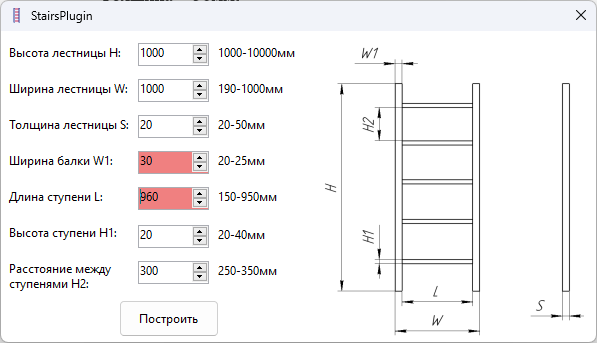


Рисунок 8.3 – Вид пользовательского интерфейса при нескольких ошибках

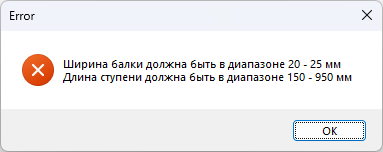


Рисунок 8.4 – Содержание сообщения при нескольких ошибках

Теперь протестируем работу плагина при стандартных, минимальных и максимальных значениях параметров. Так как минимальные значения параметров являются стандартными для плагина, то объединим эти два теста.

Результат работы плагина с минимальными и максимальными значениями продемонстрированы на рисунках 8.5 и 8.6.

Минимальные значения:

* Высота = 1000мм;
* Ширина = 190мм;
* Толщина = 20мм;
* Ширина балки = 20мм;
* Длина ступени = 190мм;
* Высота ступени = 20мм;
* Расстояние между ступенями = 250мм.



Рисунок 8.5 – Результат работы плагина с минимальными параметрами

Максимальные значения:

* Высота = 10000мм;
* Ширина = 1000мм;
* Толщина = 50мм;
* Ширина балки = 50мм;
* Длина ступени = 900мм;
* Высота ступени = 40мм;
* Расстояние между ступенями = 350мм.



Рисунок 8.6 – Результат работы программы с максимальными параметрами

## 8.2 Модульное тестирование

Для тестирования бизнес-логики плагина были написаны юнит-тесты при помощи библиотек NUnit и NUnit3TestAdapter. Юнит-тесты были написаны для классов StairsParameter и “StairsParameters.

В таблицах 8.1 и 8.2 приведены описания юнит-тестов для этих классов.

Таблица 8.1 – Описание юнит-тестов класса StairsParameter

|  |  |
| --- | --- |
| Название теста | Описание |
| MaxValue\_SetIncorrectValue\_ ThrowsArgumentException | Негативный тест сеттера максимального значения параметра, при котором присваивается некорректное значение |
| MaxValue\_SetCorrectValue\_ ReturnsSameValue | Позитивный тест сеттера максимального значения, при котором сравниваются ожидаемое и актуальное значение параметра |
| MinValue\_GetCorrectValue\_ ReturnsSameValue | Позитивный тест сеттера минимального значения, при котором сравниваются ожидаемое и актуальное значение параметра |
| Value\_SetIncorrectValue\_ ThrowsArgumentException | Негативный тест сеттера значения параметра, при котором присваивается некорректное значение |
| Value\_SetCorrectValue\_ ReturnsSameValue | Позитивный тест сеттера значения, при котором сравниваются ожидаемое и актуальное значение параметра |
| Constructor\_CreationWithCorrect Parameters\_ReturnsSameValue | Позитивный тест конструктора класса, при котором сравниваются ожидаемые значения экземпляра класса и актуальный значения |

Таблица 8.2 – Описание юнит-тестов класса StairsParameters

|  |  |  |
| --- | --- | --- |
| Название теста | Входные параметры | Описание |
| Value\_GetCorrectValue\_ ReturnsSameValue | – | Позитивный тест метода GetValue, при котором сравниваются ожидаемое и актуальное значение параметра |
| MaxValue\_GetCorrectValue\_ ReturnsSameValue | – | Позитивный тест метода GetMaxValue, при котором сравниваются ожидаемое и актуальное значение параметра |

Продолжение таблицы 8.2.

|  |  |  |
| --- | --- | --- |
| MinValue\_GetCorrectValue\_ ReturnsSameValue | – | Позитивный тест метода GetMinValue, при котором сравниваются ожидаемое и актуальное значение параметра |
| Value\_SetCorrectValue\_ ReturnsSameValue | StairsParameter Type – тип параметра лестницы, int – значение параметра | Позитивный тест метода SetValue, при котором сравниваются ожидаемое и актуальное значение параметра |
| Value\_SetStairsWidthValue\_ ReturnsCorrectRecalculated StringerWidthMaxValue | int – значение ширины лестницы, int – ожидаемое максимальное значение ширины балки | Позитивный тест, при котором ожидается, что ожидаемое максимальное значение ширины лестницы совпадет с актуальным. Рассматриваются два тестовых случая с ожидаемыми значениями 45 и 50 |
| Value\_SetStepLengthValue\_ ReturnsCorrect StringerWidthMaxValue | int – ширина балки, int – ширина лестницы, int – длина ступени | Позитивный тест, при котором ожидается, что ожидаемое максимальное значение ширины балки совпадет с актуальным. Рассматриваются два тестовых случая с входными параметрами (50, 1000, 890) и (50, 950, 850) |
| Value\_SetDependentValue\_ ThrowsArgumentException | StairsParameter Type – тип параметра лестницы, int – неправильное значение | Негативный тест зависимых параметров, при котором присваивается некорректное значение. Рассматриваются четыре тестовых случая по два для длины ступени и ширины балки со значениями 1000 и 100 для ступени и 0 и 55 для ширины балки |
| Value\_SetStairsWidthValue\_ ReturnsCorrectStepLength | – | Позитивный тест метода SetValue, при котором при котором сравниваются ожидаемое и актуальное значение параметра |

Покрытие бизнес-логики юнит-тествми плагина составило сто процентов, что показано на рисунке 8.7.

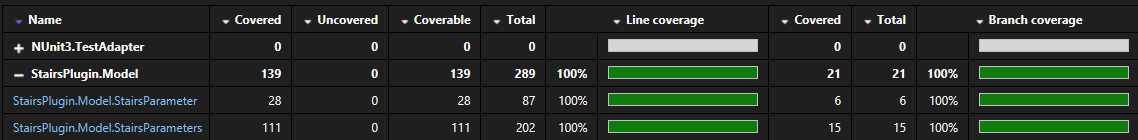


Рисунок 8.7 – Процент покрытия бизнес-логики плагина юнит-тестами

## 8.3 Нагрузочное тестирование

Нагрузочное тестирование плагина проводилось на ПК со следующей конфигурацией:

* Процессор AMD Ryzen 5 5600 6-Core Processor 3.50 GHz;
* Оперативная память объемом 16 Гб;
* Видеокарта AMD Radeon RX 6750 XT;
* Операционная система Windows 11 Pro x64.

Параметры для лестницы были заданы следующие:

* Высота = 5000мм;
* Ширина =650мм;
* Толщина = 35мм;
* Ширина балки = 35мм;
* Длина ступени = 580мм;
* Высота ступени = 20мм;
* Расстояние между ступенями = 300мм.

Тестирование проводилось около двух часов. За это время построилось 910 моделей пожарной лестницы. По графику 8.8 можно заметить, что с каждой новой моделью, время построения увеличивается линейно. Редкие резкие увеличения времени построения модели могут быть связаны с тем, что в этот момент происходит высвобождение оперативной памяти и выделение виртуальной памяти. Это приводит к кратному увеличению времени построения модели.

По графику 8.9 можно сделать вывод о том, что после построения 600 модели количество используемой оперативной памяти практически сравнялось с количеством общей оперативной памяти компьютера. Для построения последующих моделей система сама высвобождала необходимое количество оперативной и виртуальной памяти для дальнейшей работы.

Рисунок 8.8 – График зависимости времени построения от количества построенных деталей

Рисунок 8.9 – График зависимости объема оперативной памяти от количества построенных деталей

# Заключение

В процессе разработки приложения был создан плагин ‘Пожарная лестница” для системы автоматизированного проектирования КОМПАС-3D. Данный плагин является внешним приложением, с помощью которого можно строить лестницы с изменяемыми параметрами. Для лестницы предусмотрено семь изменяемых параметров: Высота, ширина и толщина лестницы, ширина балки, длина и ширина ступени, расстояние между ступенями.

Был получен первый опыт работы с API САПР и ее документацией на примере САПР КОМПАС-3D.

Также для плагина были написаны юнит-тесты для проверки бизнес-логики приложения. Помимо юнит-тестирования был проведен стресс-тест плагина, в ходе которого было построено 910 моделей лестницы примерно за два часа, что может являться неплохим результатом.

В работе также были задействованы сторонние библиотеки и расширения, которые помогали в реализации плагина. Были освоены следующе инструменты: Resharper, StyleCop, SpellCheker, GuideLines и др.

# Список использованных источников

1. Microsoft Learn. Руководство по классическим приложениям (Windows Forms .NET) [электронный ресурс]. – режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-8.0> (Дата обращения: 09.01. 2024)
2. Resharper. Расширение Visual Studio для .NET-разработчиков [электронный ресурс]. – режим доступа: <https://www.jetbrains.com/ru-ru/resharper/> (Дата обращения: 09.01. 2024)
3. Microsoft Learn. Реализация статического анализа кода с помощью StyleCop [электронный ресурс]. – режим доступа: <https://learn.microsoft.com/ru-ru/archive/msdn-magazine/2013/october/alm-rangers-implementing-static-code-analysis-with-stylecop> (Дата обращения: 09.01. 2024)
4. Плагины по серии 1.450.3. Лестницы и ограждения Tekla [электронный ресурс]. – режим доступа: <https://sapr-art.by/s1450> (Дата обращения: 10.01. 2024)
5. Руководство. Стремянка СГ [электронный ресурс]. – режим доступа: <https://docs.google.com/document/d/1DXjKSFF1_5HtommXsSB5qXpN2p4rtVMhjUq8-GVRBEM/edit#heading=h.s78n08hb4lp7> (Дата обращения: 10.01.2024)