Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «Лестничный марш» ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 582-1

\_\_\_\_\_\_\_\_\_ Баранов А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Проверил:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025

**1 ОПИСАНИЕ САПР**

* 1. **Описание программы**

КОМПАС-3D — это отечественная система трёхмерного проектирования, обеспечивающая полную импортонезависимость и являющаяся стандартом для множества предприятий и специалистов.

Система широко применяется для проектирования изделий различного назначения в таких отраслях, как машиностроение (включая транспортное, сельскохозяйственное, энергетическое, нефтегазовое и химическое), приборостроение, авиастроение, судостроение, станкостроение, вагоностроение, металлургия, промышленное и гражданское строительство, а также при создании товаров народного потребления [1].

КОМПАС-3D предоставляет пользователю широкие возможности для разработки моделей и сборок любой сложности. В её функционал входят инструменты для работы как с двумерными эскизами, так и с трёхмерными объектами. Программа поддерживает создание всех основных геометрических примитивов и предлагает большой набор операций с 3D-моделями, таких как вытягивание, вращение, вырезание и другие.

На рынке САПР КОМПАС-3D имеет ряд аналогов, среди которых можно отметить Autodesk Inventor и SOLIDWORKS.

В рамках дисциплины, выбор данной САПР объясняется наличием описания API и информации на русском языке, доступность учебной версии, а также опытом работы с данным САПР.

* 1. **Описание API**

API (от англ. Application Programming Interface — интерфейс программирования приложений) — это программный интерфейс, определяющий правила и способы взаимодействия одной программы с другой. Иными словами, API описывает, каким образом программные компоненты могут обмениваться данными и использовать функционал друг друга [2]. Для подключения и работы с API на C# потребуется выполнить ряд следующих действий:

1. Включить в свойствах проекта функцию Register for COM Interop;
2. Создать DLL-обёртку для TLB Компас API с помощью Tlblmp.exe;
3. Подключить созданный DLL к проекту;
4. Зарегистрировать библиотеку в системе КОМПАС (а именно реализовать статический метод типа .htmSample с рядом настроек)
5. Зарегистрировать библиотеку на компьютере пользователя, воспользовавшись утилитой RegAsm.exe

Таблица 1.1 − Используемые свойства класса (интерфейса) Application

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| ActiveDocument | IKompasDocument | Свойство, содержащее текущий активный документ |
| Documents | IDocuments | Коллекция всех открытых документов в приложении |
| Math2D | IMath2D | Интерфейс 2D математики |

Таблица 1.2 − Методы класса (интерфейса) Application

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| ExecuteCompasCommand | commandId: long (константа из перечисления ProcessTypeEnum или ksKompasCommandEnum),  post: bool | bool | Выполнение команды системы КОМПАС |
| MessageBoxEx | Text: BSTR (текст сообщения), Caption: BSTR (заголовок сообщения), Flags: long (флаги) | long | Выдача всплывающего сообщения |

Таблица 1.3 − Используемые свойства KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Visible | bool | Свойство, отражающее видимость окна Kompas 3d |

Таблица 1.4 − Используемые методы kompasObject

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип данных | Описание |
| Document3DI | ksDocument3D | Получить указатель на интерфейс документа трехмерной модели |
| ActiveDocument3D | ksDocument3D | Получить указатель на интерфейс текущего документа трехмерной модели |

Таблица 1.5 − Используемые методы ksPart

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| GetDefaultEntity | ObjType: Type (тип необходимого объекта) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| NewEntity | ­­­­ ObjType: Type (тип создаваемого объекта) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

Таблица 1.6 − Используемые методы ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип данных | Описание |
| GetDefinition | IDispatch | Получить указатель на интерфейс параметров объектов и элементов |
| Create | Bool (При успешном выполнении возвращает TRUE) | Создать объект в модели |

Таблица 1.7 − Используемые методы ksSketchDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| SetPlane | Plan: ksEntity (указатель на интерфейс базовой плоскости эскиза) | bool  (При успешном выполнении возвращает TRUE) | Изменить базовую плоскость эскиза |
| BeginEdit | ­­­­ – | bool(При успешном выполнении возвращает TRUE) | Войти в режим редактирования эскиза |
| EndEdit | ­­­­ – | bool(При успешном выполнении возвращает TRUE) | Выйти из режима редактирования эскиза |

Таблица 1.8 − Поля класса (интерфейса) IDocuments

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| Item | IKompasDocument | Документ, заданный по имени, ссылке или индексу |

Таблица 1.9 − Используемые методы класса (интерфейса) IDocuments

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Add | Type: long (тип документа из перечисления DocumentTypeEnum), Visible: bool (создавать документ в видимом режиме). | IKompasDocument | Создаёт новый документ |
| Open | PathName: LPCTSTR (путь, имя и расширение файла),  Visible: bool (создавать документ в видимом режиме),  ReadOnly: bool (TRUE: открыть документ только для чтения). | IKompasDocument | Открывает документ (существующий) |

Таблица 1.10 − Используемые методы ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| ksLineSeg | x1, y1: double  (Координаты первой точки)  x2, y2: double  (Координаты второй точки)  Style:int (стиль линии) | double | Угол отклонения в градусах |

Таблица 1.11 − Используемые свойства ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| directionType | short | Направление выдавливания |

Таблица 1.12 − Используемые свойства ksBaseExtrusionDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| SetSideParam | forward: bool (TRUE - прямое направление)  type: short (тип выдавливания)  depth: double (глубина выдавливания)  draftValue: double (угол уклона)  draftOutward: bool (TRUE –уклон наружу) | bool | Установить параметры выдавливания в одном направлении |
| SetSketch | Sketch: ksEntity (указатель на интерфейс эскиза) | bool | Задать указатель на интерфейс эскиза элемента |

* 1. **Обзор аналогов плагина**

Первым аналогом является приложения ПЛАГИН «ПЕПЕЛЬНИЦА» [3] для Компас-3D. Проект написан на C# и предназначен для создания различных форм пепельниц. Он демонстрирует, как можно использовать функциональность api КОМПАС-3D для создания 3D-моделей через пользовательский интерфейс плагина. Плагин является примером того, как организовать создание объектов с помощью параметров, передаваемых из пользовательского интерфейса в процесс моделирования. Интерфейс взаимодействия представлен на рисунке 1.1.

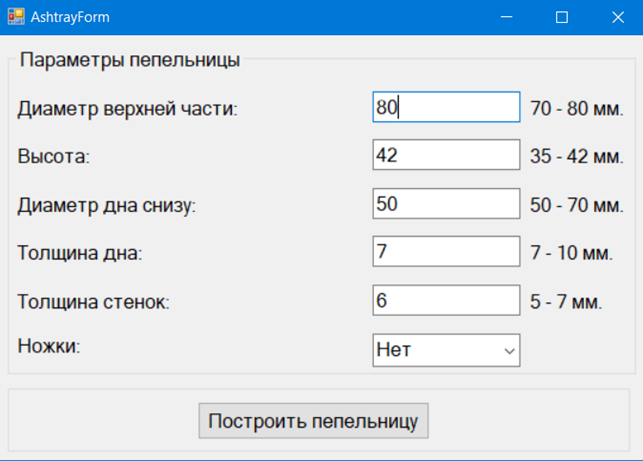


Рисунок 1.1 − Интерфейс приложения «ПЕПЕЛЬНИЦА»

Вторым аналогом является плагин «Построение втулки амортизаторной» для КОМПАС-3D. Проект написан на C# и ориентирован на моделирование втулки амортизаторной. [4] Он подключается к КОМПАС-3D, предоставляя пользователю инструменты для параметрического проектирования амортизаторной втулки. Проект находится в открытом доступе на GitHub, что позволяет изучить его и проанализировать любому желающему.

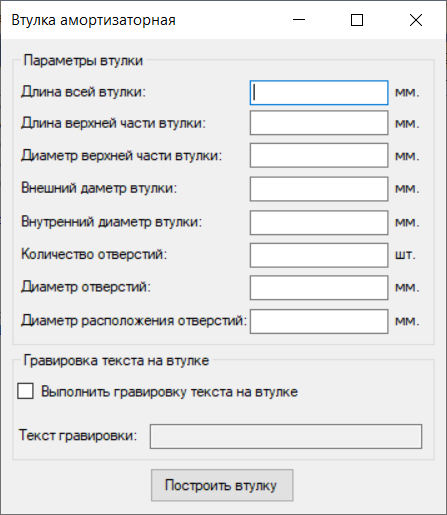


Рисунок 1.2 − Интерфейс приложения «Построение втулки амортизаторной»

**2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ**

Лестничный марш — это конструктивный элемент лестницы, представляющий собой непрерывный ряд ступеней, соединяющий два промежуточных или крайних уровня. Он используется для безопасного и удобного перемещения людей между этажами здания.

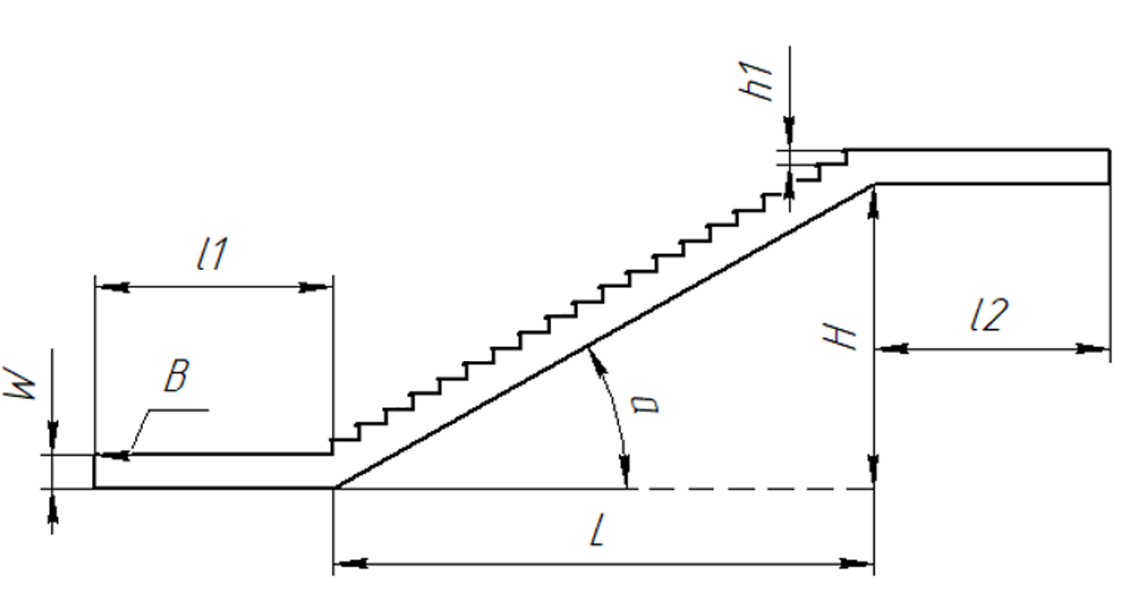


Рисунок 2.1 − Модель лестничного марша.

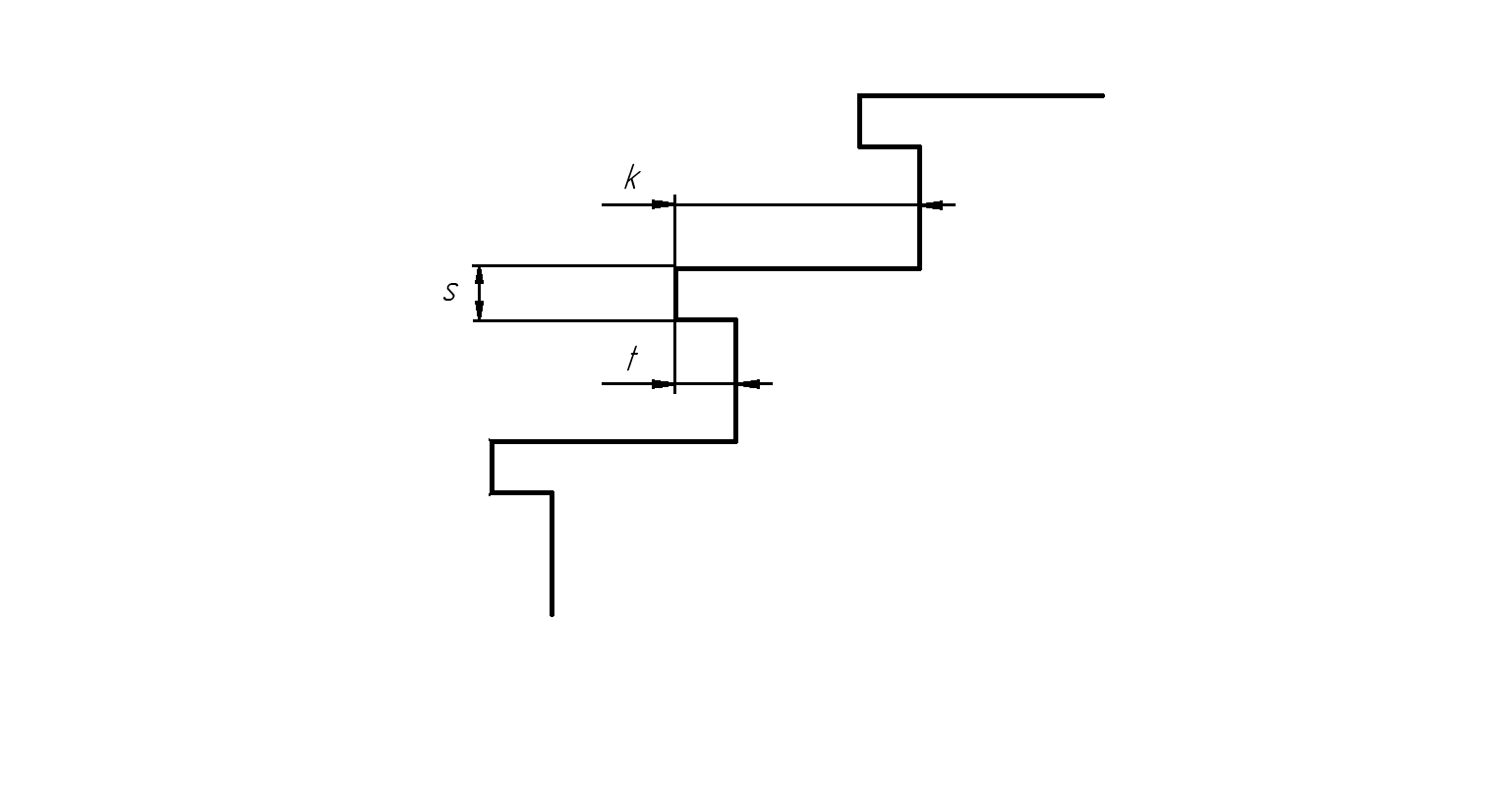


Рисунок 2.2 — Модель ступени марша с размерами

***Изменяемые параметры для предмета проектирования*** (также все обозначения показаны на рисунке 2.1 и 2.2):

− высота марша H (500мм – 8000мм);

− длина пролёта L (500мм – 8000мм);

− число ступеней N (1 - 60);

− высота ступени h1 (120мм – 200мм);

− ширина марша B (800мм – 2500мм);

− глубина выступа t (≤ 0.5 × h1.);

− толщина выступа s (≤ 0.5 × h1.);

− длина нижней платформы l1 (1000мм – 5000мм);

− длина верхней платформы l2 (1000мм – 5000мм);

− толщина марша W (100мм – 500мм).

**3 ПРОЕКТ СИСТЕМЫ**

**3.1 UML диаграмма классов**

UML — это стандартный язык визуального моделирования, предназначенный для следующего использования:

− моделирование бизнеса и подобных процессов;

− анализ, проектирование и внедрения программных систем.

UML — это общий язык для бизнес-аналитиков, архитекторов и разработчиков программного обеспечения, используемый для описания, спецификации, проектирования и документирования существующих или новых бизнес-процессов, структуры и поведения артефактов программных систем.[5].

UML диаграмма классов для плагина «Лестничный марш» представлена на рисунке 3.1.

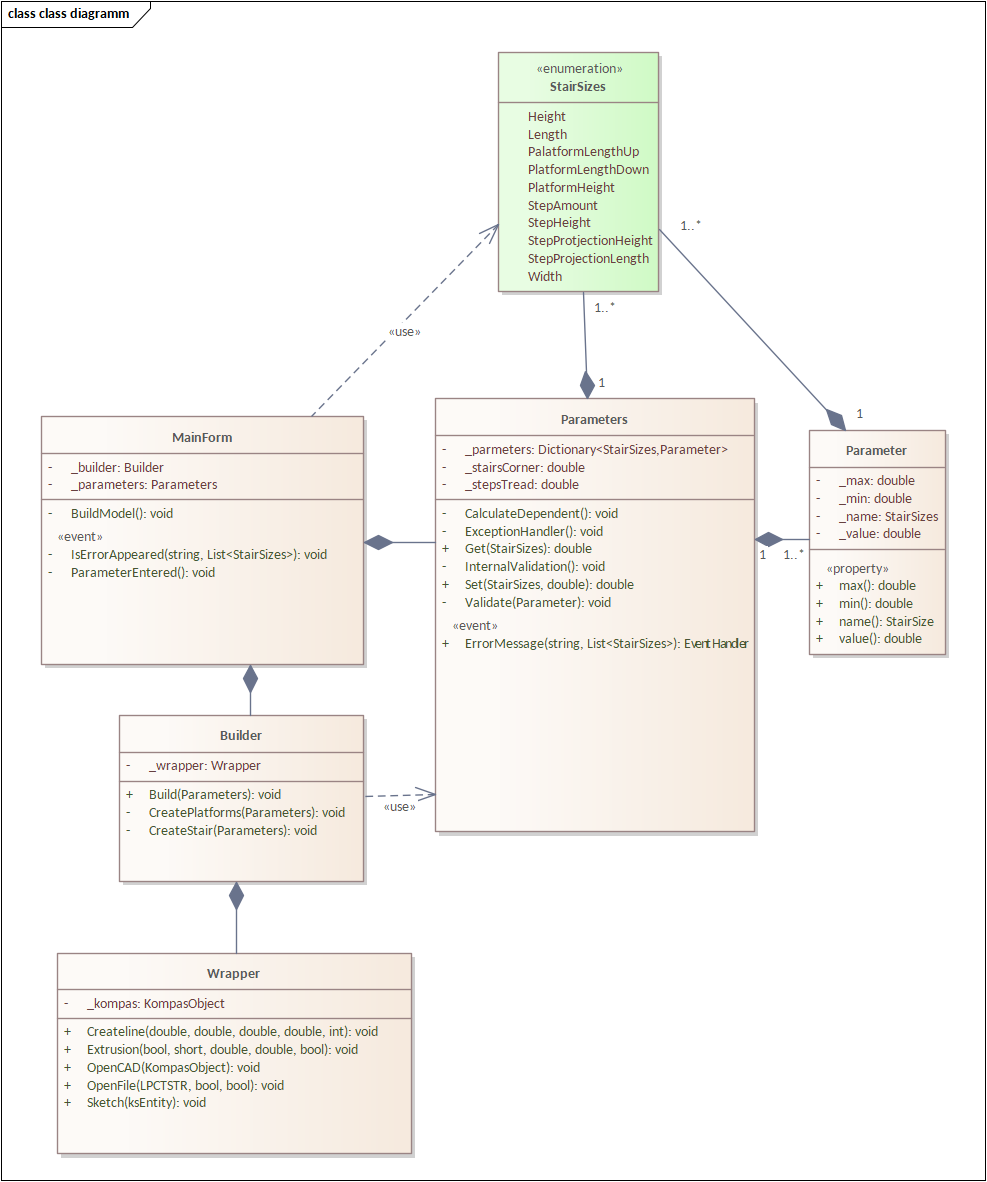


Рисунок 3.1 – UML диаграмма классов для плагина «Лестничный марш»

В таблицах ниже представлена информация о свойствах и методах каждого из классов.

Таблица 3.1 – Поля класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранит в себе объект построения |
| \_parameters | Parameters | Хранит в себе параметры для объекта построения |

Таблица 3.2 – Методы класса MainForm

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| BuildModel | − | Построение модели по заданным параметрам |
| IsErrorAppeared | message: string (сообщение об ошибке)  wrongs<enum> (список неправильных переменных) | Событие ошибки валидации, выводит сообщение об ошибке и подсвечивает соответствующие ему переменные |
| ParameterЕntered | − | Событие ввода переменной, отправляет переменную на валидацию, запускает внутреннюю валидацию или расчёт вычисляемых переменных |

Таблица 3.3 − Поля класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_stairsCorner | double | Хранит угол марша, необходим для валидации |
| \_stepsTread | double | Хранит длину проступи, необходим для валидации |
| \_parmeters | dictionary<StairSizes,Parameter> | Словарь, содержащий параметры, описанные в разделе 2 |

Таблица 3.4 − Методы класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CalculateDependent | − | Просчитывает недостающие переменные среди h1 и H а также необходимые для валидации tread\_length и alpha |
| ExeptionHandler | − | Обработчик исключений, формирует ошибку для вывода пользователю |
| InternalValidation | − | Проверка зависимых параметров |
| Validate | parameter: Parameter (Вводимый параметр) | Валидация входа в рамки от минимального к максимальному отдельному параметру |

Окончание таблицы 3.4

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип данных | Описание |
| Parameters | name: StairSizes (Название параметра)  value: double (Величина параметра) | double | Свойство принимает параметры и валидирует входимость в границы |
| ErrorMessage | − | ErrorEvent | Событие отправки ошибки в MainForm, в него добавляется обработчик «IsErrorAppeared» из MainForm |

Таблица 3.5 − Поля класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_wrapper | Wrapper | Хранит в себе объект обёртки API |

Таблица 3.6 − Методы класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| Build | parameters: Parameters | Построение модели по заданным параметрам |
| Create\_platforms | parameters: Parameters | Построение платформ |
| Create\_stair | parameters: Parameters | Построение ступеней |

Таблица 3.7 − Поля класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_kompas | KompasObject | Хранит в себе ссылку на компас |

Таблица 3.8 − Методы класса Wrapper

|  |  |  |
| --- | --- | --- |
| Название | Входные параметры | Описание |
| CreateLine | x1: double, y1: double, x2: double, y2: double, style: int | Создание линии |
| Sketch | Plan: ksEntity | Создание эскиза |
| Extrusion | forward: bool, type: short, depth: double, draftValue: double, draftOutward: bool | Выдавливание эскиза |
| CreateFile | PathName: LPCTSTR, Visible: bool, ReadOnly: bool | Создание файла |
| OpenFile | PathName: LPCTSTR, Visible: bool, ReadOnly: bool | Открытие файла |
| ModifyFile | PathName: LPCTSTR, Visible: bool, ReadOnly: bool | Редактирование файла |
| OpenCAD | Kompas: KompasObject | Открытие Компас3D |

**3.2 Пользовательский нтерфейс**

На рисунке 3.2 представлен макет пользовательского интерфейса.

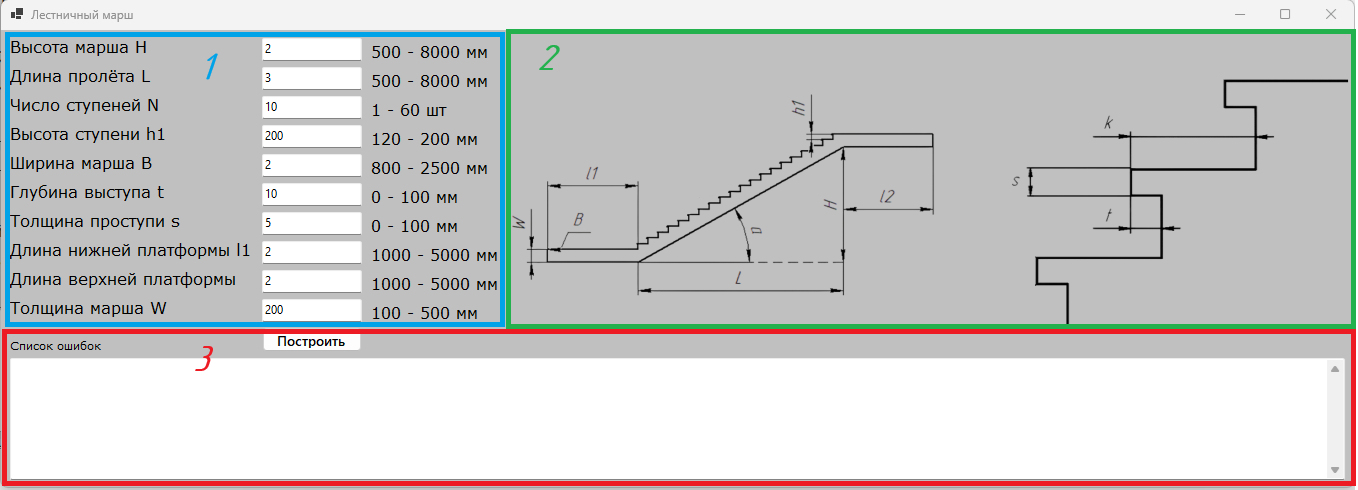


Рисунок 3.2 − Макет пользовательского интерфейса

На макете изображён планируемый интерфейс приложения. Слева сверху (отмечено 1, синим) находится поля ввода значений подписи к ним и их границы, слева снизу (отмечено 2, красным) поле вывода для ошибок и предупреждений с возможностью пролистывания, а справа (отмечено 3, зелёным) модель лестницы.

Если в ходе ввода или построения произошла ошибка (рисунок 3.3) в поле вывода будет размещён текст ошибки, а соответствующие поля ввода загорятся красным.

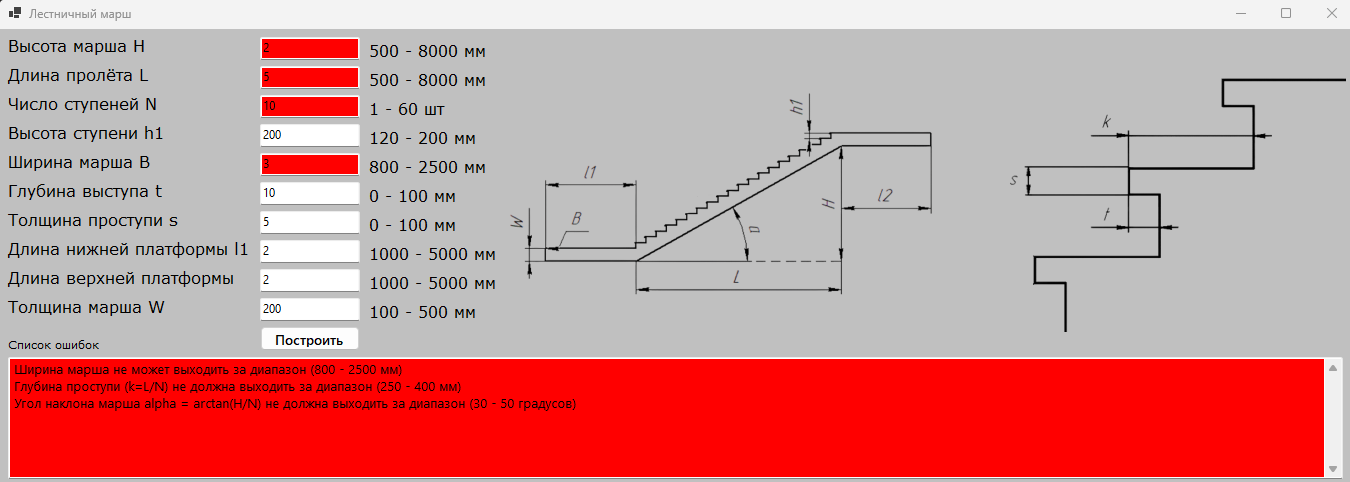


Рисунок 3.3 − Ошибка при неправильном вводе

**4 СПИСОК ИСТОЧНИКОВ**

1. КОМПАС-3D [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/about/> (дата обращения 25.09.2025)
2. API [Электронный ресурс]. − Режим доступа <https://habr.com/ru/articles/464261/> (дата обращения 28.09.2025)
3. «ПЕПЕЛЬНИЦА» [Электронный ресурс]. − Режим доступа <https://github.com/maxvishnyakov/ORSAPR> (дата обращения 14.10.2025)
4. «Построение втулки амортизаторной» [Электронный ресурс]. − Режим доступа <https://github.com/sergeydubinin/PluginKompas-3D> (дата обращения 14.10.2024)
5. UML [Электронный ресурс]. − Режим доступа https://www.uml-diagrams.org/ (дата обращения 28.09.2025)