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 «Молоток»**

**ДЛЯ «КОМПАС-3D»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 582-1

\_\_\_\_\_\_\_\_\_\_Хорошкова Э.Г. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. Руководитель:

к.т.н., доцент каф. КСУП \_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А.А. «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

* 1. Описание программы

КОМПАС-3D – это российская импортонезависимая система трёхмерного проектирования, ставшая стандартом для тысяч предприятий и сотен тысяч профессиональных пользователей.

КОМПАС-3D широко используется для проектирования изделий основного и вспомогательного производств в таких отраслях промышленности, как машиностроение (транспортное, сельскохозяйственное, энергетическое, нефтегазовое, химическое и т.д.), приборостроение, авиастроение, судостроение, станкостроение, вагоностроение, металлургия, промышленное и гражданское строительство, товары народного потребления и т. д.[1]

Данная САПР позволяет проектировать модели и сборки разного уровня сложности, благодаря разнообразному функционалу, включающего в себя работу как с 2-мерными эскизами, так и с 3D-моделями. В САПР есть возможность работать со всеми основными примитивами необходимыми для создания эскизов и моделей, а также существует достаточное количество инструментов для работы с 3D-моделями (вытягивание, вращение, вырезание и др.).

Компас 3D имеет множество прямых аналогов на рынке, среди них встречаются AutodeskInventor, SOLIDWORKS и др.

В рамках дисциплины выбор данной САПР объясняется наличием описания API на русском языке, доступность учебной версии САПР без необходимости получать одобрения от компании, а также большим количеством информации на сторонних ресурсах на русском языке, позволяющим детальнее узнать о возможностях работы с САПР.

1.2 Описание API

SDK КОМПАС-3D - это ориентированные на прикладного программиста инструментальные средства разработки приложений (библиотек конструктивов, прикладных САПР) на базе системы КОМПАС-3D.)

Для подключения API к проекту плагина требуется:

1. скопировать в папку проекта файлы: KAPITypes.dll, Kompas6API5.dll, Kompas6Constants.dll, Kompas6Constants3D.dll, KompasAPI7.dll;
2. подключить скопированные .dll файлы к проекту путем добавления их в «Ссылки» проекта;
3. с помощью оператора using подключить .dll файлы к скриптам, где планируется использование API.

Для взаимодействия с Kompas 3d с помощью API требуется создать объект типа «kompasObject», у которому нужно использовать команду kompas.CreateInstance(L"KOMPAS.Application.5"). С помощью свойства kompas.visible можно открыть окно компаса.

Таблица 1.1 – используемые свойства KompasObject

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

Таблица 1.2 – 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 1.3 – используемые методы ksPart

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

Таблица 1.4 – 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 1.5 – используемые методы ksSketchDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| SetPlane | ksEntity plan- 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 1.6 – используемые методы ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| ksLineSeg | double x1,double y1 -Координаты первой точки double x2,double y2- Координаты второй точки int style – стиль линии | int(В случае неудачи возвращает 0, в случае успеха указатель на отрезок ) | Создать отрезок |
| ksCircle | double x1,double y1-координаты центра окружности  double rad – радиус  int style – стиль линии | int(В случае неудачи возвращает 0, в случае успеха указатель на окружность) | Создать отрезок |

Таблица 1.8 – используемые методы ksFilletDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип данных | Описание |
| Array | ksEntityCollection | Получить указатель на интерфейс массива скругляемых объектов (граней и ребер) |

Таблица 1.9 – используемые свойства ksBaseExtrusionDefinition

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

Таблица 1.10 – используемые методы ksBaseExtrusionDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| SetSideParam | bool forward – направление выдавливания направление выдавливания: (TRUE -прямое направление, FALSE – обратное направление.)  short type – тип выдавливания  double depth – глубина выдавливания  double draftValue –угол уклона  bool draftOutward – направление уклона (TRUE –уклон наружу, FALSE – уклон внутрь.) | Bool  (TRUE в случае успеха, FALSE в случае неудачи) | Установить параметры выдавливания в одном направлении |
| SetSketch | ksEntity sketch - указатель на интерфейс эскиза ksEntity | Bool  (TRUE в случае успеха, FALSE в случае неудачи) | Задать указатель на интерфейс эскиза элемента |

Таблица 1.11 – используемые свойства ksCutExtrusionDefinition

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

Таблица 1.12 – используемые методы ksBaseExtrusionDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| SetSideParam | bool forward – направление выдавливания направление выдавливания: (TRUE - прямое направление, FALSE – обратное направление.)  short type – тип выдавливания  double depth – глубина выдавливания  double draftValue –угол уклона  bool draftOutward – направление уклона (TRUE –уклон наружу, FALSE – уклон внутрь.) | bool  (TRUE в случае успеха) | Изменить параметры выдавливания в одном направлении |
| SetSketch | ksEntity sketch - указатель на интерфейс эскиза ksEntity | bool  (TRUE в случае успеха) | Изменить указатель на интерфейс эскиза элемента |

1.3 Обзор аналогов плагина

Первым аналогом является плагин для генерации модели топора в КОМПАС-3D.[3] Он демонстрирует, как можно расширить функциональность КОМПАС-3D для создания 3D-моделей через пользовательский интерфейс плагина. Плагин разработан на языке C# и интегрируется в среду КОМПАС-3D, предоставляя пользователю инструменты для параметрического проектирования топоров. Проект находится в открытом доступе на GitHub, что позволяет изучить его и проанализировать любому желающему. Данный аналог является прямым для разрабатываемого плагина «Молоток». Интерфейс взаимодействия представлен на рисунке 1.1.

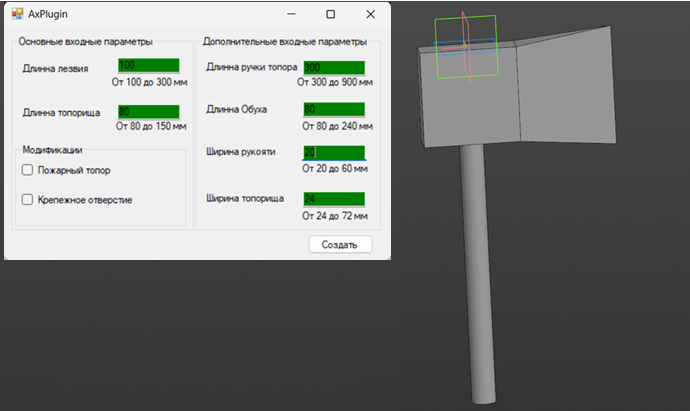


Рисунок 1.1 − Интерфейс приложения «Плагин для Компас 3D “Топор”»

Вторым аналогом плагина является «Плагин для КОМПАС 3D – Клинок»[4]. Данный плагин позволяет строить клинок ножа с рукоятью, на основе введенных параметров. Интерфейс плагина и результат его работы представлен на рисунке 1.2.

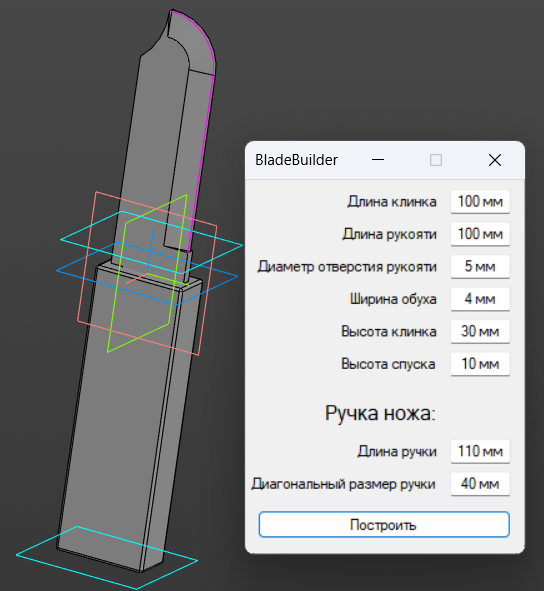


Рисунок 1.2 – Интерфейс и результат работы плагина

**2 Описание предмета проектирования**

Молоток — это ударный ручной инструмент, состоящий из головки (обычно металлической) и рукоятки, предназначенный для забивания гвоздей, разбивания предметов и выполнения других работ, где требуется силовое воздействие. Чертеж модели представлен на рисунке 2.1.

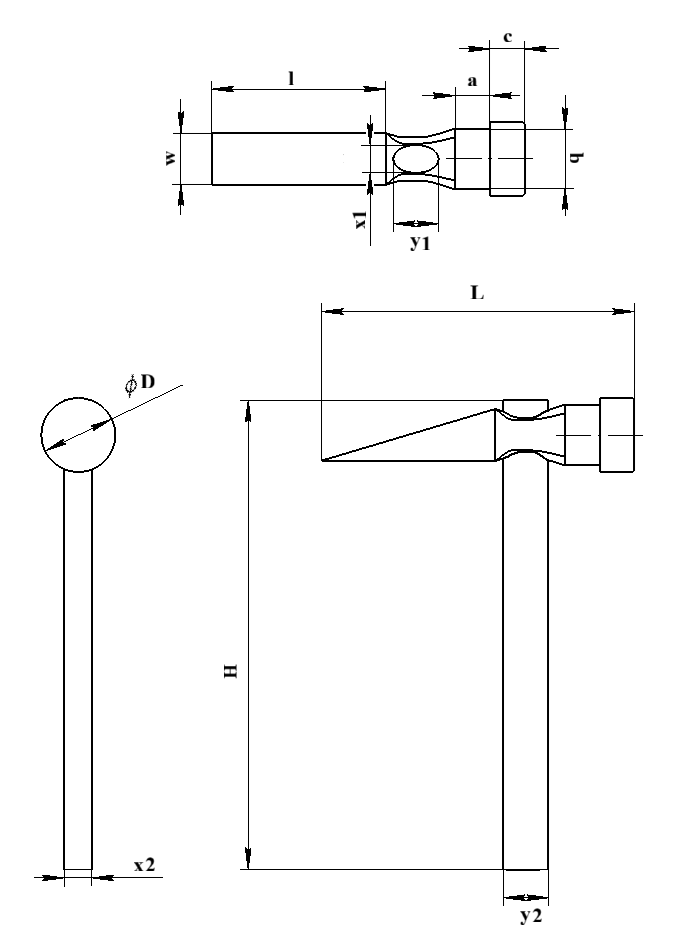


Рисунок 2.1 — Модель молотка с размерами

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

− общая высота молотка H (150-600мм);

− длина головки молотка L (2/3 высоты H);

− диаметр бойка D (20 – 100мм);

− ширина бойка с (10-100мм);

− ширина выступа перед бойком a (10-100мм);

− диаметр выступа перед бойком b (меньше диаметра бойка D);

− ширина отверстия под рукоятку(всад)x1(20 - 100);

− ширина отверстия под рукоятку(всад) y1(10 - 80);

− ширина рукояти x2 (зависит от отверстия в головкеx1);

− ширина рукояти y2 (зависит от отверстия в головкеy1);

− длина носка l (1/1,8 длины головки L);

− ширина носка w (не больше диаметра бойка D,10-100 мм).

**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.[6]

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

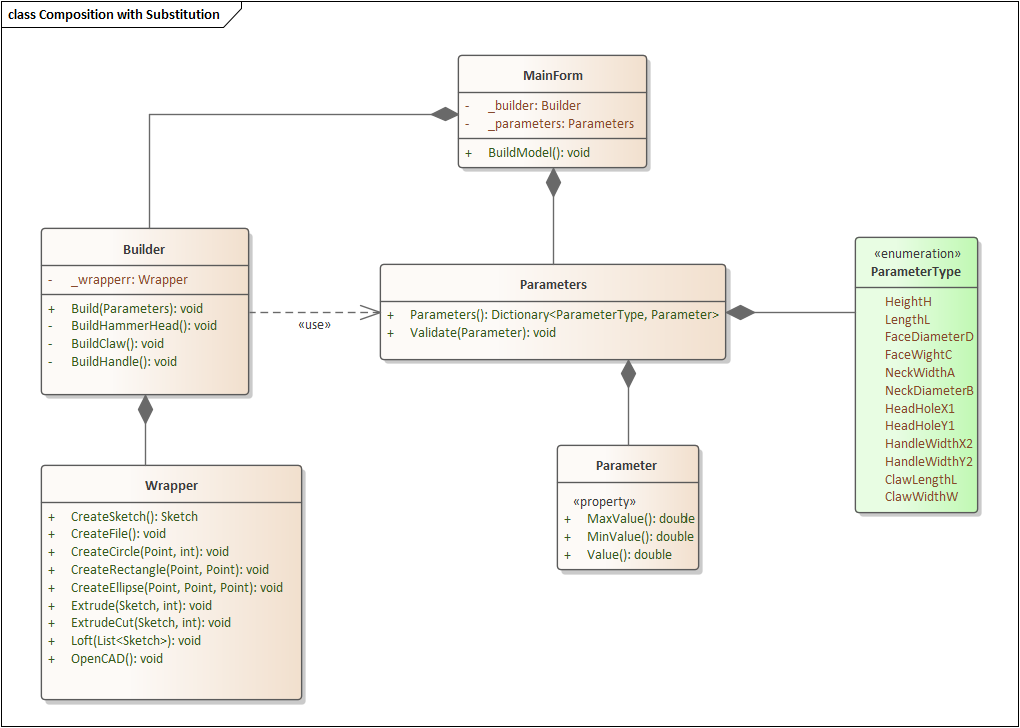


Рисунок 3.1 –UML диаграмма классов для плагина «Молоток»

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 | | Тип возвращаемых данных | Описание |
| BuildModel | | void | Построение модели по заданным параметрам |
|  |

Таблица 3.3 – Поля класса Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_wrapper | Wrapper | Хранит объект объект обертки API |

Таблица 3.4 – Методы класса Builder

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Build | Parameters | void | Последовательное построение всех частей модели |
| BuildHammerHead | - | void | Построение головки молотка |
| BuildClaw | - | void | Построение носка головки молотка |
| BuildHandle | - | void | Построение рукоятки |

Таблица 3.5 – Методы класса Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Тип возвращаемых данных | Описание |
| Parameters | - | Dictionary<ParameterType, Parameter> | Хранит словарь с параметрами объекта построения |
| Validate | Parameter | void | Выполняет валидацию параметра |

Таблица 3.6 –Методы класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| MaxValue | double | Максимальное значение параметра |
| MinValue | double | Минимальное значение параметра |
| Value | double | Значение параметра |

Таблица 3.7 – Методы класса Wrapper

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| CreateSketch | - | Sketch | Создание эскиза |
| CreateFile | - | void | Создание файла |
| CreateCircle | Point, int | void | Создание окружности |
| CreateRectangle | Point, Point | void | Создание прямоугольника |
| CreateEllipse | Point, Point, Point | void | Создание эллипса |
| Extrude | Sketch, int | void | Операция выдавливания |
| ExtrudeCut | Sketch, int | void | Операция вырезания выдавливанием |
| Loft | List<Sketch> | void | Создание элемента по сечениям |
| OpenCAD | - | void | Открытие CAD системы |

3.2 Макеты пользовательского интерфейса

Пример макета пользовательского интерфейса, с введенными по умолчанию параметрами представлен на рисунке 3.2.

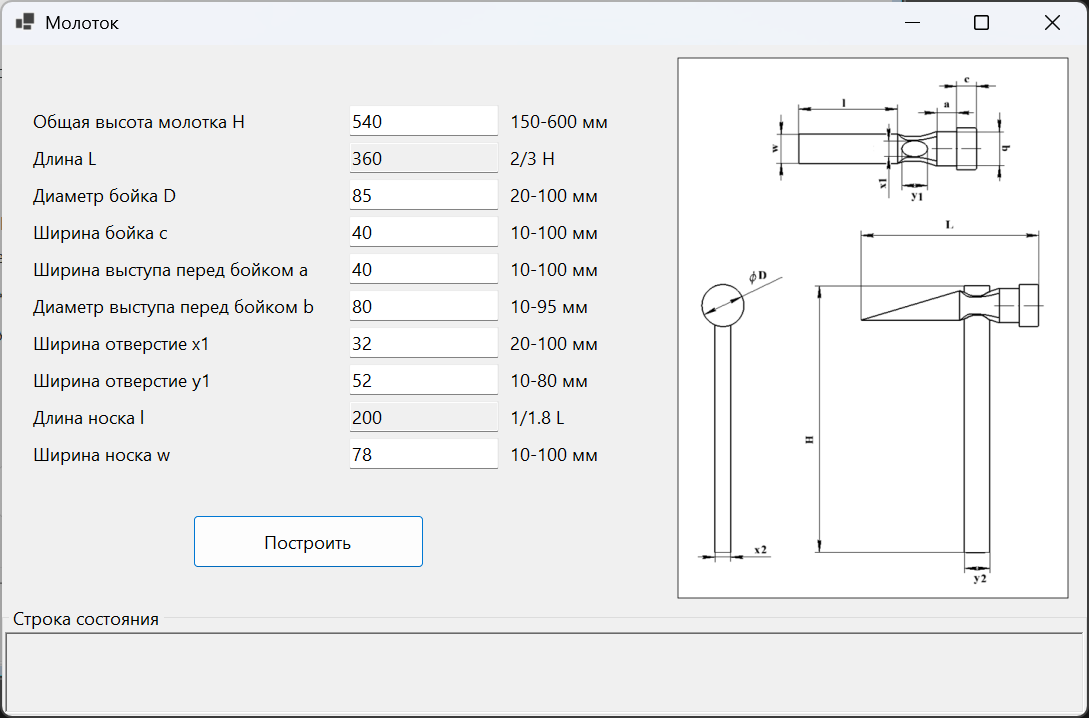


Рисунок 3.2 − Макет пользовательского интерфейса

Пользовательский интерфейс можно разделить на три части: область ввода параметров, окно вывода ошибок, чертеж молотка с обозначениями. В области ввода данных пользователь вводит параметры для построения модели молотка. При вводе некорректных данных, поле с некорректными данными подсвечивается красным и в окне вывода ошибок появляется пояснение.

На рисунке 3.3 представлена ошибки валидации, подсвечивание полей и пояснение ошибок в строке состояния.

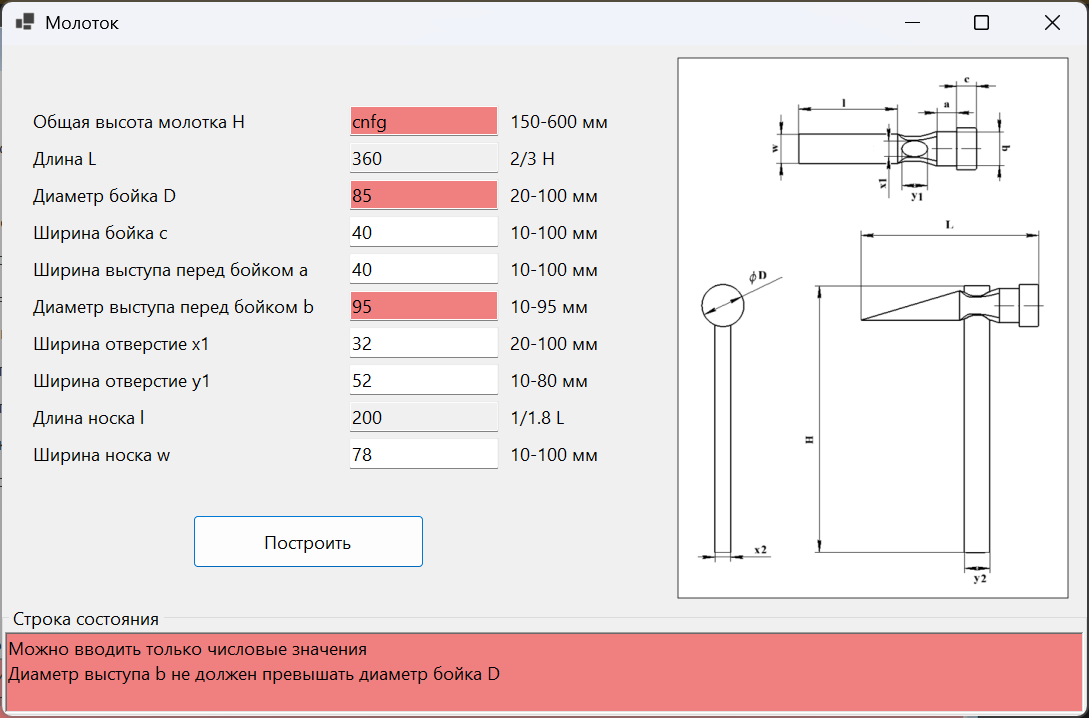


Рисунок 3.3 – Интерфейс с неправильно введенными значениями параметров

**4 Список источников**

1. КОМПАС-3D [Электронный ресурс]. − Режим доступа <https://kompas.ru/kompas-3d/about/> (дата обращения 15.10.2025)
2. API[Электронный ресурс]. − Режим доступа <https://habr.com/ru/articles/464261/>(дата обращения 15.10.2025)
3. Ax\_plugin[Электронный ресурс]. − Режим доступа [https://github.com/Vlad-Ust-123/Ax\_for\_orsapar](https://github.com/Vlad-Ust-123/Ax_for_orsapar%20) (дата обращения 27.10.2025)
4. Плагин для КОМПАС-3D “Клинок”. [Электронный ресурс]. − Режим доступа https://github.com/eskorpado/kompas-blade-plugin (дата обращения 05.10.2025)
5. UML [Электронный ресурс]. − Режим доступа https://www.uml-diagrams.org/ (дата обращения 15.10.2025)