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 «КЛИНОК ДЛЯ МЕЧА/НОЖА»**

**ДЛЯ «КОМПАС-3D»**

По дисциплине

«Основы разработки САПР» (ОРСАПР)

Выполнил:

студент гр. 582-1

\_\_\_\_\_\_\_\_\_ Бальжинов Д.Ж

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Руководитель:

к.т.н, доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ Калентьев А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023г.

Томск 2025

**1. ОПИСАНИЕ САПР**

1.1. Информация о выбранной САПР

2.1.1. Краткое описание САПР

Kompas-3d V23- семейство систем автоматизированного проектирования, универсальная система автоматизированного проектирования, позволяющая в оперативном режиме выпускать чертежи изделий, схемы, спецификации, таблицы, инструкции, расчётно-пояснительные записки, технические условия, текстовые и прочие документы.

Изначально система ориентирована на оформления документации в соответствии с ЕСКД, ЕСТД, СПДС и международными стандартами, но этим возможности системы не ограничиваются. [1]

Kompas-3d V23 был выбран по следующим причинам:

1.Доступность. Компания разработчик «Аскон» предоставляет бесплатную учебную версию ПО, что содержит все необходимые функции для проектирования, благодаря чему разработка плагина для данной версии ПО возможна.

2.Локаризированность. В виду того, что разработчик ПО «Аскон» является отечественным разработчиком, интерфейс ПО полностью переведен на русский язык, что облегчает освоение ПО.

2.1.2. Аналоги САПР

1. SOLID WORKS - программный комплекс САПР для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку изделий любой степени сложности и назначения.

2.Autodesk Inventor - система трёхмерного твердотельного и поверхностного параметрического проектирования (САПР) компании Autodesk, предназначенная для создания цифровых прототипов промышленных изделий.

1.2. Описание API

КОМПАС-3D API - это ориентированные на прикладного программиста инструментальные средства разработки приложений (библиотек конструктивов, прикладных САПР) на базе системы КОМПАС-3D.)[2]

Для подключение 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("KOMPAS.Application5"). С помощью свойства kompas.visible можно открыть окно компаса .

Таблица 1.1 – используемые свойства и поля KompasObject

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

Таблица 1.2 – используемые методы kompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| Document3D | – | 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 | Входные  параметры | 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 окружность |

Таблица 1.7 – используемые свойства и поля ksFilletDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| radius | double | Радиус скругления |
| tangent | bool | Признак продолжения скругления по касательным ребрам |

Таблица 1.8 – используемые методы ksFilletDefinition

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

Таблица 1.9 – используемые свойства и поля ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| directionType | short | 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 3.10 – используемые свойства и поля ksCutExtrusionDefinition

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

Таблица 3.11 – 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

Первым прямым аналогом плагина является «Плагин для КОМПАС 3D – Клинок»[1]. Данный плагин позволяет строить клинок ножа с рукоятью, на основе введенных параметров. Интерфейс плагина и результат его работы представлен на рисунке 2.1.

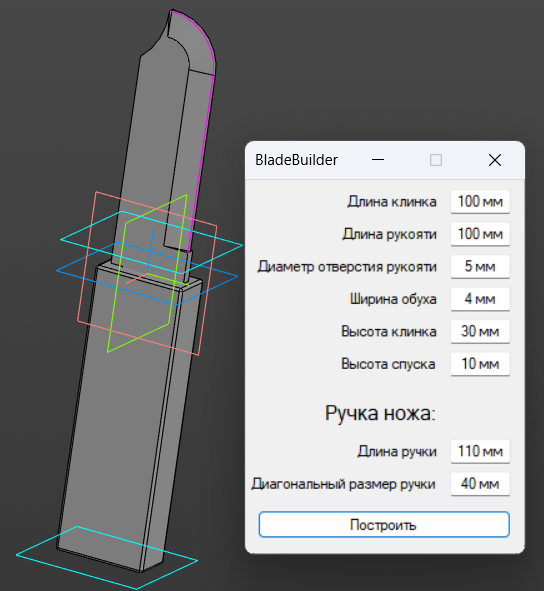


Рисунок 2.1. – Интерфейс и результат работы плагина

Вторым прямым аналогом плагина является веб-сервис для построения ножей «Knifeprint»[2]. Данный веб сервис предоставляет набор инструментов для построения ножа, путем черчения и дальнейшего преобразования в 3д модель. На рисунках 2.2-2.3 представлен интерфейс веб-сервиса.

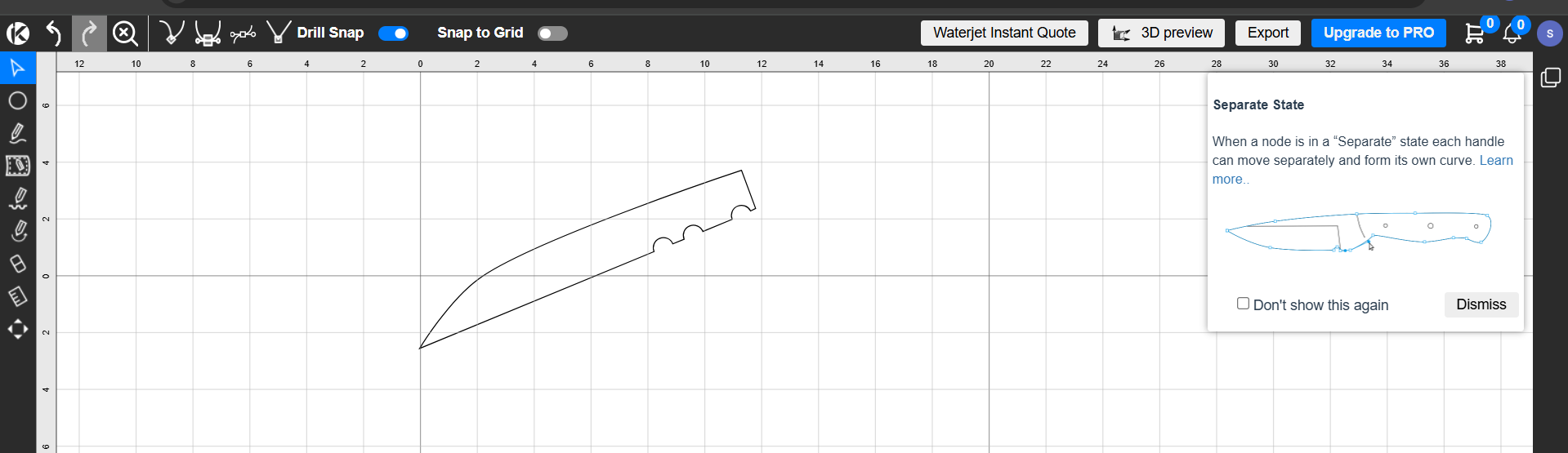


Рисунок 2.2 – Интерфейс создания 2д эскиза

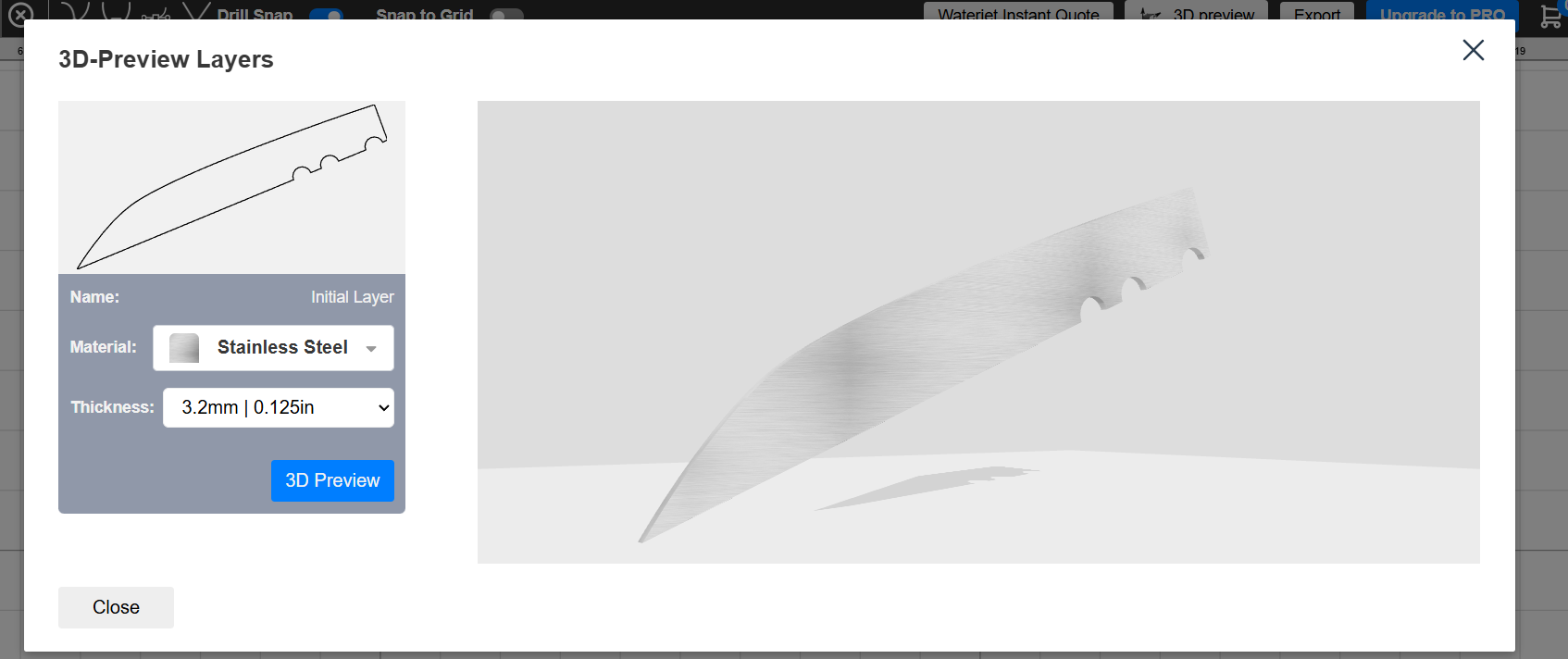


Рисунок 2.3 – Интерфейс создания 3д модели

**2. Описание предмета проектирования**

Клинок ножа - протяженная металлическая боевая часть холодного оружия с острием и одним или двумя лезвиями, являющаяся частью полосых, (согласно ГОСТ’У Р 51500-99 [3]).

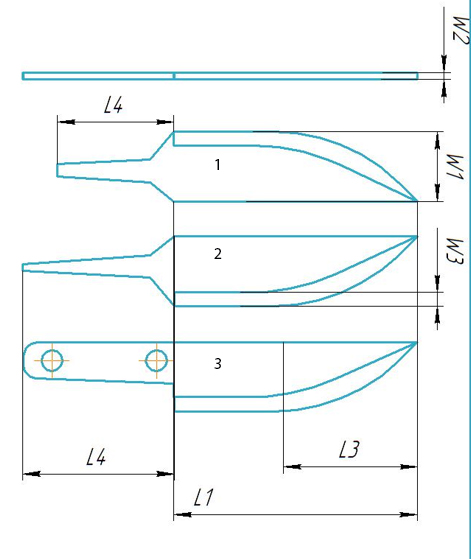


Рисунок 2.1 – Пример чертежа клинка.

Параметры клинка:

-Длина клинка L1(30-1200мм);

-Ширина клинка W1 (9-60мм);

-Толщина клинка W2(1-3мм);

-Тип клинка (двусторонний/ односторонний);

-Наличия острия(Да/нет);

-Тип крепления(всадной(1), сквозной(2), накладной(3), отсутствует);

-Ширина лезвия W3 (от 1/6 до 3/6 ширины клинка);

-Длина острия L3 (1/6 длины клинка);

-Длина крепления L4( Если сквозное, то равно длине клинка; если всадное то 3/4 клинка; если накладное, равно длине клинка).

**3. Проект системы**

3.1 Uml-диаграмма

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 «Клинок для меча/ножа» представлена на рисунке 3.1.

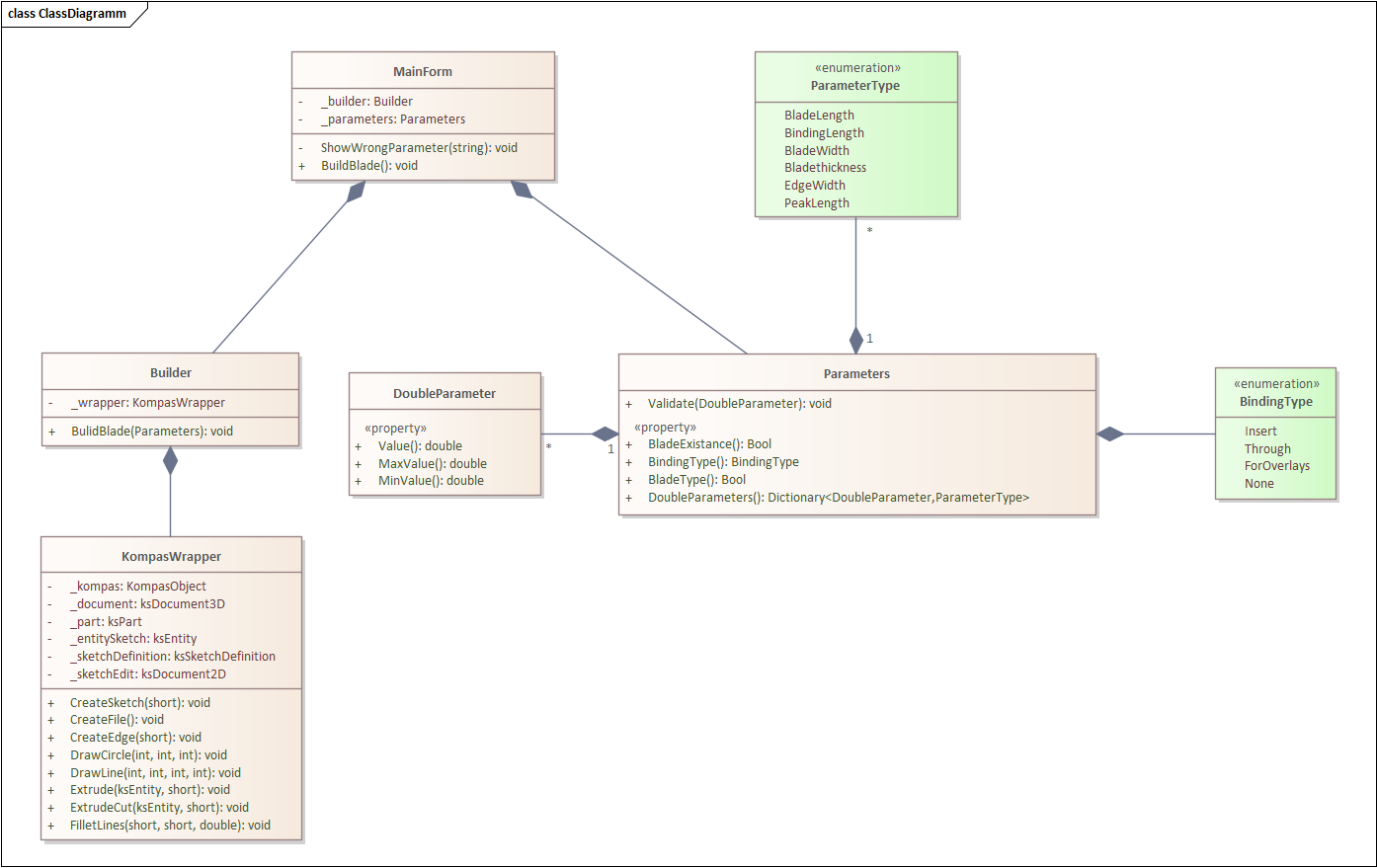


Рисунок 3.1 – Uml диаграмма плагина

В таблицах ниже представлена информация о поля, свойствах и методах классов.

Таблица 3.1 – используемые свойства и поля MainForm

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранить в себе объект строителя |
| \_parameters | Parameters | Хранит в себе объект параметров |

Таблица 3.2 – используемые методы MainForm

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| BuildBlade | – | void | Начать постройку клинка |
| ShowWrongParameter | string property – строка несущая в себе информацию об допущенной ошибке | void | Указывает на параметр, в котором была допущена ошибка |

Таблица 3.3 – используемые свойства и поля Builder

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_builder | Builder | Хранить в себе объект строителя |
| \_parameters | Parameters | Хранит в себе объект параметров |

Таблица 3.4 – используемые методы Builder

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| BuildBlade | – | void | Начать постройку клинка |
| ShowWrongParameter | string property – строка несущая в себе информацию о допущенной ошибке | void | Указывает на параметр, в котором была допущена ошибка |

Таблица 3.5 – используемые свойства и поля и поля Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| BladeType | bool | Тип клинка(TRUE -двусторонний, FALSE -односторонний) |
| BladeExistance | bool | Наличие острия(TRUE -есть, FALSE -нету) |
| BindingType | BindingType | Тип крепления |
| DoubleParameters | Dictionary<DoubleParameter,ParameterType> | Числовые параметры |

Таблица 3.6 – используемые методы Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| Validate | double DoubleParameter – Параметр для валидации | bool | Выполняет валидацию параметра |

Таблица 3.7 – используемые свойства и поля и поля Parameters

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

Таблица 3.8 – используемые свойства и поля KompasWrapper

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| \_kompas | KompasObject | Хранить в себе объект kompas-3d |
| \_document | ksDocument3D | Указатель на интерфейс документа |
| \_part | ksPart | Указатель на интерфейс компонента |
| \_entitySketch | ksEntity | Указатель на интерфейс сущности |
| \_sketchDefinition | ksSketchDefinition | Указатель на интерфейс параметров эскиза |
| \_sketchEdit | ksDocument2D | Указатель на эскиз |

Таблица 3.9 – используемые методы KompasWrapper

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные  параметры | Возвращаемый тип данных | Описание |
| CreateSketch | short plane – плоскость построения | void | Создать скетч |
| CreateFile | – | void | Создание файла в Kompas-3d |
| CreateEdge | short plane – плоскость построения | void | Создание скетча для лезвия |
| DrawCircle | int X – x координата центра окружности  int Y – y координата центра окружности  int radius – радиус окружность | void | Нарисовать окружность |

4

Конец таблицы 3.9

|  |  |  |  |
| --- | --- | --- | --- |
| DrawLine | int X1 – x координата первой точки отрезка  int Y1 – y координата первой точки отрезка  int X2 – x координата второй точки отрезка  int Y2 – y координата второй точки отрезка | void | Нарисовать отрезок |
| Extrude | ksEntity sketch – выдавливаемый скетч  short direction – направление выдавливания | void | Операция выдавливания |
| ExtrudeCut | ksEntity sketch – вырезаемый скетч  short direction – направление вырезания | void | Операция вырезания выдавливанием |
| FilletLines | short line1- первый отрезок  short line2- второй отрезок  double radius- радиус скругления | void | Создать скругление между двумя линиями |

Таблица 3.10 – Перечисление BindingType

|  |  |
| --- | --- |
| Название | Описание |
| Insert | Тип крепления- «всадное» |
| Through | Тип крепления- «сквозное» |
| ForOverLays | Тип крепления- «накладное» |
| None | Тип крепления «отсутсвует» |

Таблица 3.11 – Перечисление ParameterType

|  |  |
| --- | --- |
| Название | Описание |
| BladeLength | Тип параметра – «Длина клинка» |
| BladeWidth | Тип параметра – «Ширина клинка» |
| Bladethickness | Тип параметра – «Толщина клинка» |
| EdgeWidth | Тип параметра – «Ширина лезвия» |
| PeakLengthe | Тип параметра – «Длина острия» |

3.2 Макеты пользовательского интерфейса

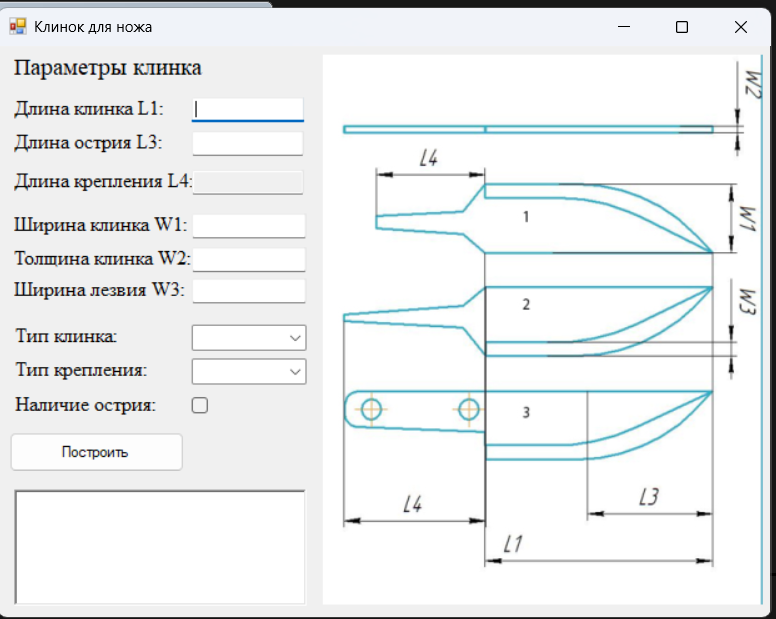


Рисунок 3.2 – Макет пользовательского интерфейса

Пользовательский интерфейс можно разделить на три части: область ввода параметров, окно вывода ошибок, область содержащая отображение всех параметров.

В области ввода данных пользователь вводить параметры для постройки клинка. Допустимые значения для параметров показываются при наведении на поле ввода.

При вводе некорректных данных, поле куда были ввели данные подсвечивается красным. Пример подсказки, что появляется при наведении на поле показан на рисунке 3.3. Пример подсвечивания некорректных параметров показан на рисунке 3.4.

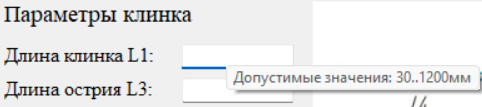


Рисунок 3.3. – Подсказка с допустимыми значениями

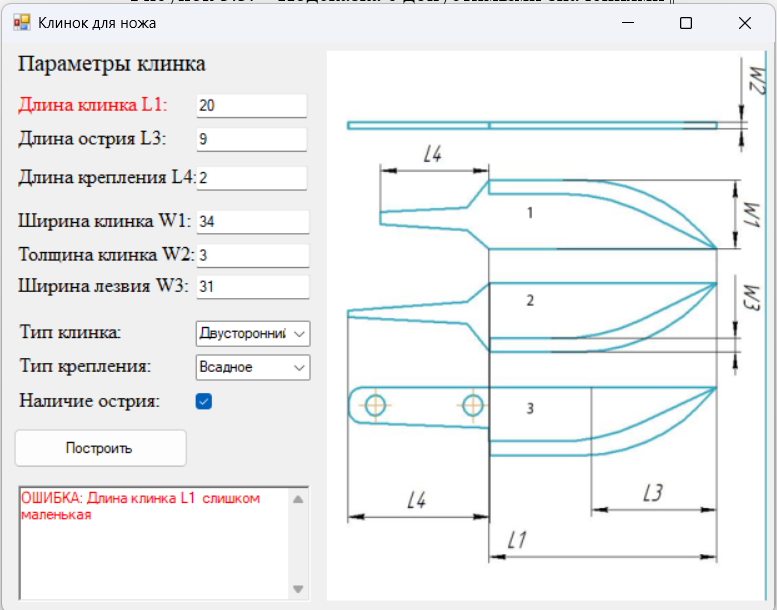


Рисунок 3.4 – Пример отображения ошибки

**4. Список источников**

1. КОМПАС-3D [Электронный ресурс]. − Режим доступа https://kompas.ru/kompas-3d/about/ (дата обращения 25.09.2025)

2. SDK КОМПАС-3D [Электронный ресурс]. − Режим доступа https://help.ascon.ru/KOMPAS\_SDK/23/ru-RU/index.html (дата обращения 03.10.2025)

3. ГОСТ Р 51500-99 [Электронный ресурс]. − Режим доступа https://docs.cntd.ru/document/1200027279(дата обращения 23.9.2025)

4. Плагин для компас 3D “Клинок”. [Электронный ресурс]. − Режим доступа https://github.com/eskorpado/kompas-blade-plugin (дата обращения 05.10.2025)

5.Веб сервис KnifePrint. [Электронный ресурс]. − Режим доступа [https://knifeprint.com](https://knifeprint.com/)(дата обращения 05.10.2025)

6. UML [Электронный ресурс]. − Режим доступа https://www.uml-diagrams.org/ (дата обращения 11.10.2025)