Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

РАЗРАБОТКА ПЛАГИНА “ГЕНЕРАТОР ОТВЕРТОК” ДЛЯ ПРОГРАММЫ КОМПАС 3D

Проект системы по дисциплине

«Основы разработки САПР»

Выполнил:

Студент гр. 589-3

\_\_\_\_\_\_\_\_\_ Бабушкин А.П.

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ Калентьев А.А.

(оценка) \_\_\_\_\_\_\_\_\_

Томск 2022 г

Оглавление

[1 ОПИСАНИЕ САПР 3](#_Toc116583343)

[1.1 Описание САПР «КОМПАС-3D v18.1» 3](#_Toc116583344)

[1.2 Анализ API 4](#_Toc116583345)

[1.3 Обзор аналогов 8](#_Toc116583346)

[2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ 10](#_Toc116583347)

[3 ПРОЕКТ СИСТЕМЫ 12](#_Toc116583348)

[3.1 Диаграмма классов 12](#_Toc116583349)

[3.2 Макеты пользовательского интерфейса 14](#_Toc116583350)

[Список использованных источников 18](#_Toc116583351)

# 1 ОПИСАНИЕ САПР

## 1.1 Описание САПР «КОМПАС-3D v18.1»

**КОМПАС-3D v18.1** —Одна из программ семейства систем автоматизированного проектирования «Компас», позволяющая создавать трехмерные модели и сборки из них. КОМПАС-3D использует собственное математическое ядро и параметрические технологии, что является особенностью данной программы. В нее, помимо системы трехмерного моделирования, также входит универсальная САПР «Компас-График», созданная для составления чертежей и графиков. [1]

## 1.2 Анализ API

API (англ. Application Programming Interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. Все эти коммуникации происходят с помощью функций, классов, методов, структур, а иногда констант одной программы, к которым могут обращаться другие.

Для КОМПАС-3D созданы две различные версии API - версии 5 и версии 7. К ним разработчик прилагает справочную систему по всем включенным в эту API интерфейсам.

Наиболее важными для разработки любого приложения могут посчитаться следующие интерфейсы:

1 – **IKompasAPIObject**: базовый интерфейс для всех интерфейсов КОМПАС API, кроме интер­фейсов событий и некоторых вспомогательных интерфейсов;

2 – **IAp****plic****ation**: интерфейс приложения КОМПАС-3D;

3 – **IDocuments**: коллекция документов, открытых в приложении КОМПАС-3D;

4 – **IKompasError**: интерфейс информации об ошибках системы КОМПАС-3D;

5 – **IModelObject**: базовый интерфейс для всех модельных объектов.

Также, интересными для разработки конкретно лабораторного приложения можно посчитать следующие интерфейсы:

1 – **IModelContainer**: позволяет работать с коллекциями 3D-объектов, входящих в состав 3D-объекта.;

2 – **ISurfaceContainer**: устанавливает и получает коллекции операций с поверхностями;

3 – **ISketch**: интерфейс взаимодействия с эскизом;

4 – **IPart7**: интерфейс компонента 3D документа;

5 – **IPlane3D**: интерфейс плоскости 3D.

Далее будут приведены самые важные для использования в лабораторной программе методы и свойства интерфейсов. Конечно же, программа не ограничится их использованием, а будет также применять более локальные методы и свойства, предсказать использование которых будет достаточно сложно:

Таблица 1.1 – Необходимые методы/свойства класса IKompasAPIObject

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| Application | Ссылка на интерфейс IApplication | Позволяет получить ссылку на приложение |

Таблица 1.2 – Необходимые методы/свойства класса IApplication

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| ActiveDocument | Ссылка на интерфейс IKompasDocument | Позволяет получить текущий активный документ |
| HideMessage | Константа из перечисления ksHideMessageEnum | Позволяет скрывать/показывать сообщения |
| KompasError | Ссылка на интерфейс IKompasError информации о ошибке системы КОМПАС | Позволяет получить информацию об ошибке системы КОМПАС |
| Visible | bool | Позволяет изменить видимость приложения |
| Quit | - | Позволяет закрыть приложение |

Таблица 1.3 – Необходимые методы/свойства класса IDocuments

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| Add | Ссылка на интерфейс IKompasDocument | Позволяет создать новый документ и добавить его в коллекцию |

Таблица 1.4 – Необходимые методы/свойства класса IKompasError

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| Clear | - | Позволяет сбросить ошибку |
| Report | - | Позволяет вывести сообщение о ошибке |
| Code | Из перечисления ErrorType и ErrorType3d. | Позволяет получить код ошибки |
| Description | string | Позволяет получить описание ошибки |
| Error3D | bool | Позволяет определить, какому из двух перечисле­ний соответствует код ошибки, полученный через свойство Code |

Таблица 1.5 – Необходимые методы/свойства класса IModelObject

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| Update | bool | Метод Update необходимо вызвать для вступления в силу примененных изменений объекта |
| Hidden | bool | Позволяет получить и установить состояние видимости объекта |
| Part | Ссылка на интерфейс IPart7 | Позволяет получить компонент, владеющий элементом |

Таблица 1.6 – Необходимые методы/свойства класса IModelContainer

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| AddObject | Ссылка на интерфейс IModelObject | Создает новый элемент 3D модели |

Таблица 1.7 – Необходимые методы/свойства класса ISurfaceContainer

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения | Описание |
| EvolutionSurfaces | Ссылка на интерфейс IEvolutions | Экспорт коллекции поверхностей выдавливания |

## 1.3 Обзор аналогов

Аналогом приложения для генерации однотипных 3d-моделей можно посчитать плагин BookGen для 3d-редактора Blender. В нем можно создавать объекты, которые будут заполняться случайно сгенерированными книгами. Можно изменять размеры генерируемых книг, материалы страниц и обложки, расстояние между книгами, а также прочие настройки. К каждой числовой настройке можно поменять свойство случайности, для придания набору книг неповторимости.

Концепция данного плагина, во многом, схожа с планируемым лабораторным приложением. Единственное глобальное отличие – приложение не будет генерировать объект их множества изменяемых объектов, а ограничится созданием лишь одного экземпляра изменяемого объекта. Соответственно, значения случайности в будущем приложении не понадобятся. [2]

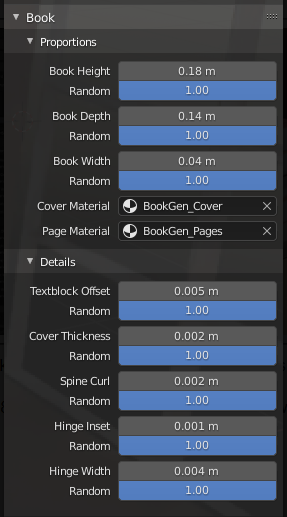


Рисунок 1.1 – Окно настроек объекта дополнения



Рисунок 1.2 – Сцена, сгенерированная при помощи данного плагина

Также, существует плагин для Blender, который может генерировать болты и винты различной резьбы с изменяемыми шляпками. Принцип в этом плагине во многом схожа с плагином выше, за исключением того, что он не может генерировать несколько болтов в одном объекте, что делает этот плагин даже более похожим на планируемое приложение. [3]



Рисунок 1.3 – Сцена, сгенерированная при помощи данного плагина

В ходе продолжительного поиска, было выявлено, что плагинов для КОМПАС-3D, делающие что-то похожее на запланированное к реализации в лабораторном приложении, либо не существует, либо они не находятся в открытом распространении.

# 2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ

Отвертка – инструмент, которым можно выкручивать винты. Состоит из рукоятки и стержня. Существуют отвертки с различными видами наконечников стержня.

Изображение моделируемого объекта:

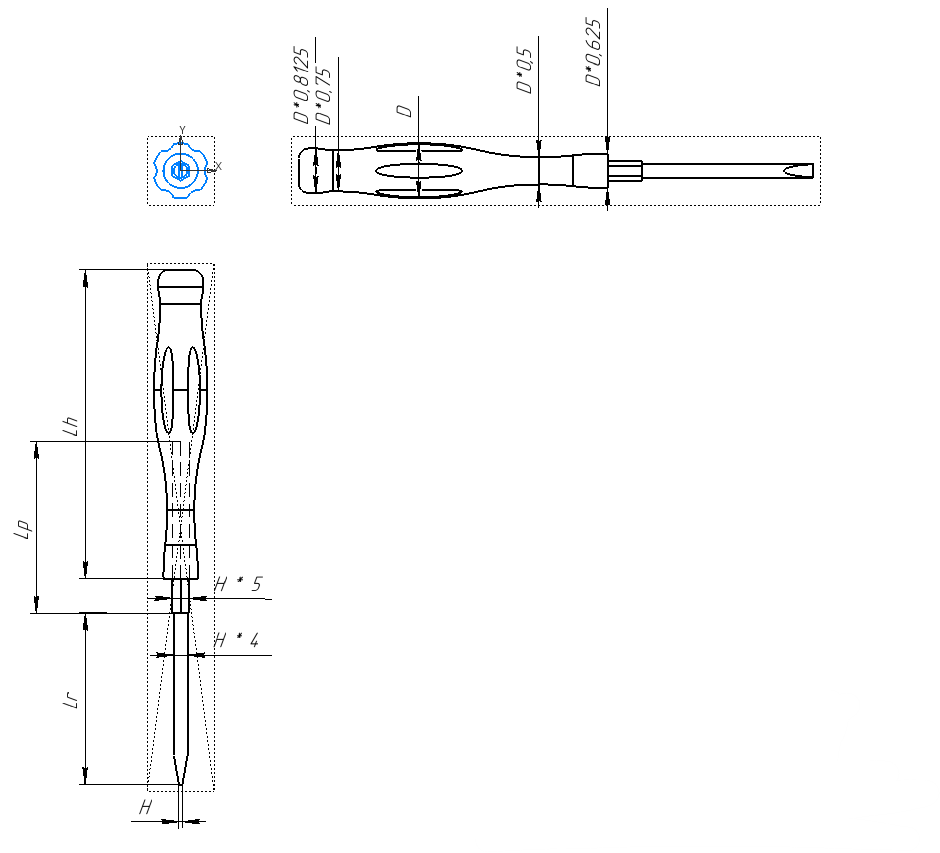


Рисунок 2.1 – Чертеж отвертки

Измеряемые параметры для плагина:

* Наконечник – На выбор будет представлено 3 наконечника из списка - плоский, крестовой и треугольный (см. рис. 3)
* H – Высота наконечника стержня отвертки (0,1 - 10 мм);

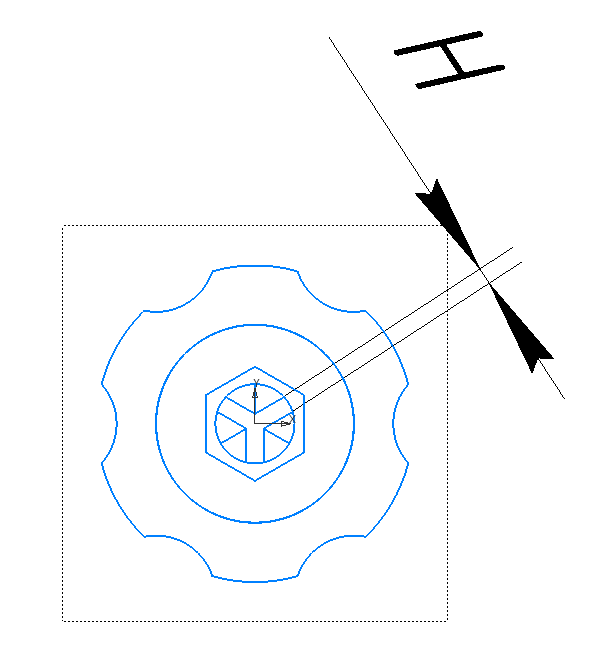
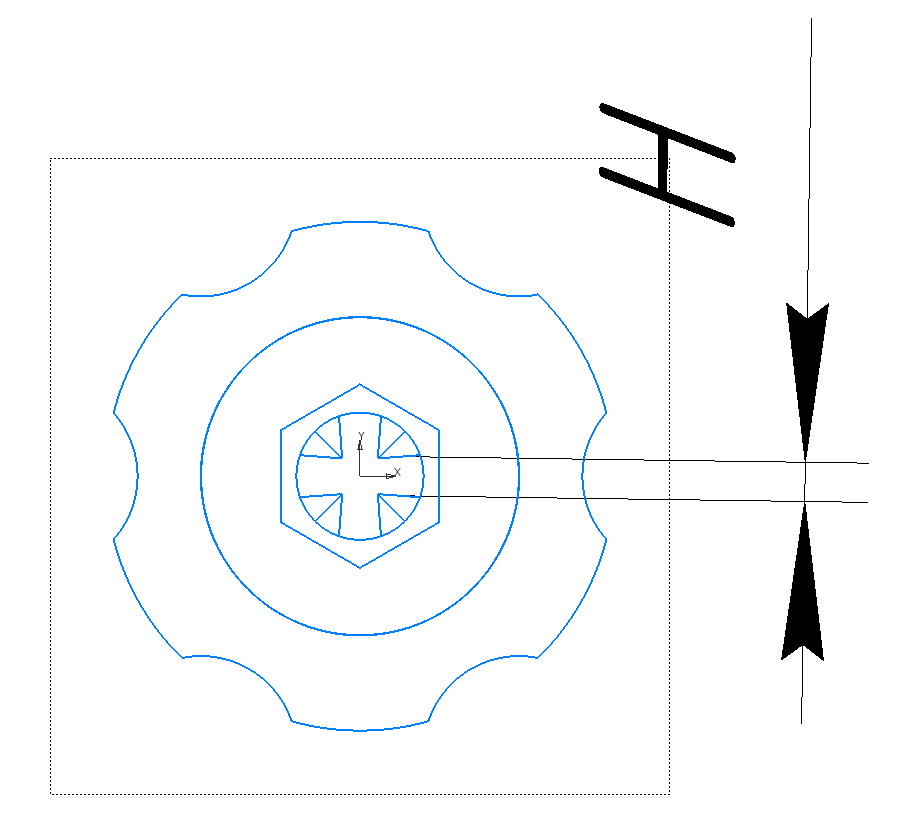
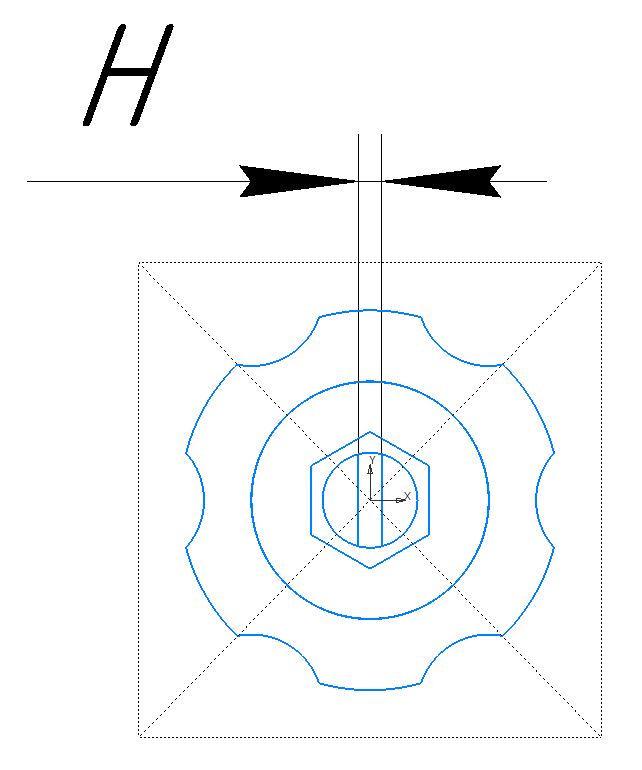


Рисунок 2.2 – Переменная H в зависимости от выбранного наконечника

* D – Самая широкая часть рукоятки (1,6 - 240 мм) - Не должна быть меньше, чем H \* 16 и больше, чем H \* 24;
* Lr – Длина внешней части стержня (1 - 400 мм) - Не должна быть меньше, чем H \* 20 и больше, чем H \* 400;
* Lh – Длина рукоятки отвертки (6 - 900 мм) - Не должна быть меньше, чем D \* 3,75 и больше, чем D \* 7,5;
* Lp – Длина внутренней части стержня (3 - 540 мм) - Либо эта переменная равна Lh \* 0,5, либо Lh \* 0,6. Выбор склоняется к тому, хочет ли пользователь, чтобы часть скрепляющей части стержня была видна снаружи, или нет;

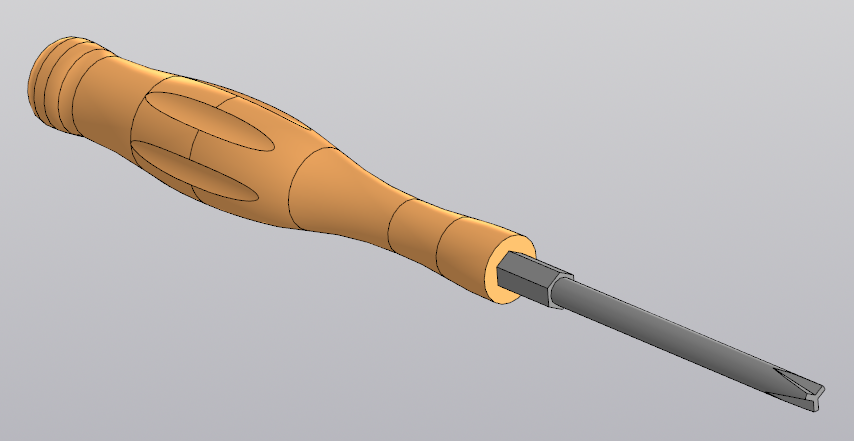


Рисунок 2.3 – Собранная отвертка

# 3 ПРОЕКТ СИСТЕМЫ

## 3.1 Диаграмма классов

Унифицированный язык моделирования (UML) – это язык моделирования общего назначения, предназначенный для обеспечения стандартного способа визуализации проектирования системы. В 1997 году UML был принят в качестве стандарта Object Management Group (OMG) и с тех пор управляется этой организацией. [4]

UML-диаграмма классов будущего лабораторного приложения представлена на рисунке 3.1:

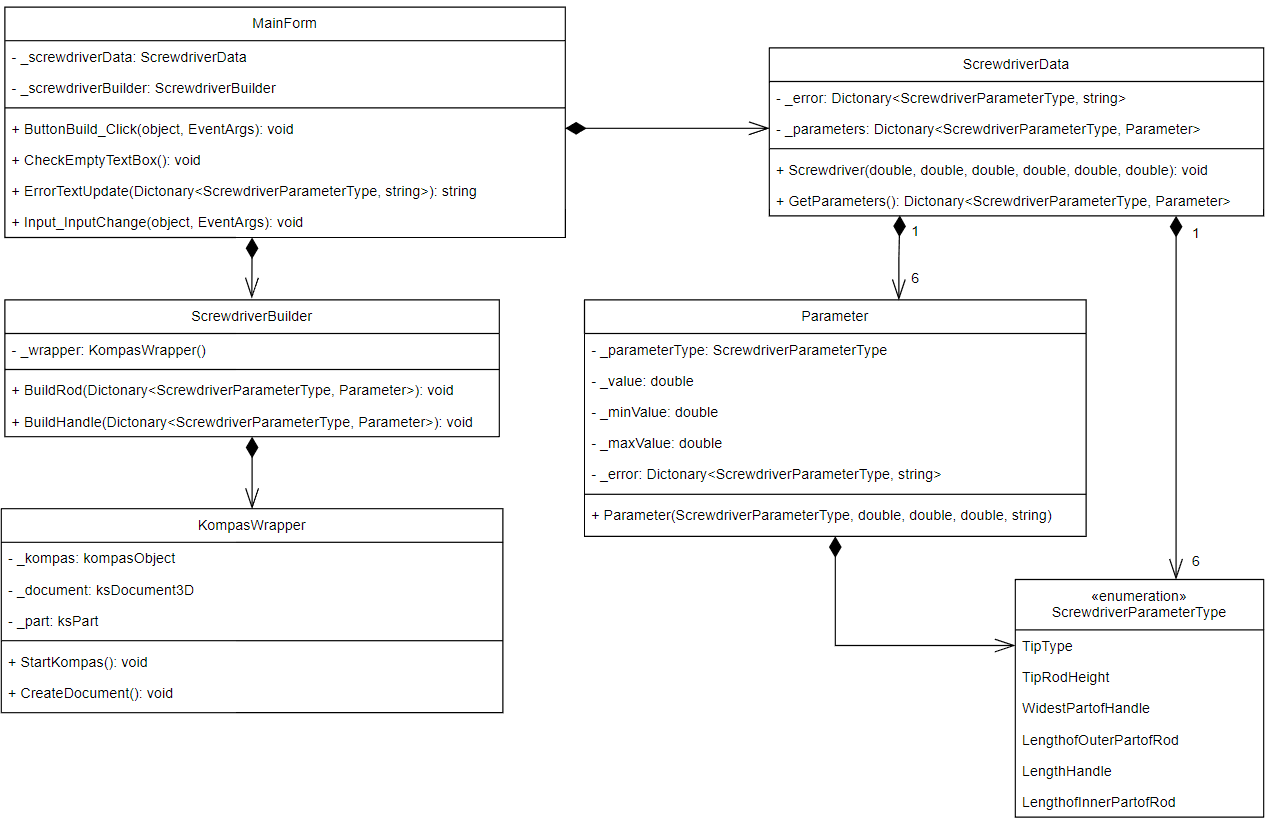


Рисунок 3.1 – UML-диаграмма лабораторного приложения

Перечисление ScrewdriverParameterType необходимо для того, чтобы программа знала, какие виды входных параметров могут существовать. Оно используется для создания словарей, а также его элемент содержатся в классе входного параметра Parameter. Также в нем находятся значение параметра, его возможный минимум и максимум, а также словарь с текстом ошибки. Все параметры, со всеми ошибками хранятся в классе ScrewdriverData. Пользователь взаимодействует с программой через форму MainForm. В ней, помимо класса ScrewdriverData, хранится класс ScrewdriverBuilder, созданный для взаимодействия с функциями API для КОМПАС 3D. ScrewdriverBuilder содержит в себе алгоритмы создания модели отвертки по входным параметрам. Сами функции, которые нужны для создания модели, будут взяты из KompasWrapper.

Таблица 3.1 – Описание методов/свойств класса Parameter

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения/хранимое значение | Описание |
| \_parameterType | ScrewdriverParameterType | Определяет, какой входной параметр описывает данный класс |
| \_value | double | Значение параметра |
| \_minValue | double | Минимально возможное значение параметра |
| \_maxValue | double | Максимально возможное значение параметра |
| \_error | Dictonary< ScrewdriverParameterType, string> | Словарь ошибок, возникших при редактировании параметра |

Таблица 3.2 – Описание методов/свойств класса ScrewdriverData

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения/хранимое значение | Описание |
| \_error | Dictonary< ScrewdriverParameterType, string> | Общий словарь всех ошибок, появившихся при редактировании набора параметров |
| \_parameters | Dictonary< ScrewdriverParameterType, Parameter> | Общий словарь всех входных параметров |
| GetParameters | Dictonary< ScrewdriverParameterType, Parameter> | Возвращает словарь всех входных данных |

Таблица 3.3 – Описание методов/свойств класса MainForm

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения/хранимое значение | Описание |
| \_screwdriver | Screwdriver | Хранит в себе объект во всеми входными параметрами и ошибками ввода данных |
| \_screwdriverBuilder | ScrewdriverBuilder | Хранит в себе методы создания модели |
| ButtonBuild\_Click | - | Активируется при нажатии кнопки «Build», вызывает проверки входных параметров и генератор модели |
| CheckEmptyTextBox | - | Проверяет наличие пустых текстовых строк, нужен для выведения ошибки о наличии таковых, если в словаре ошибок нет ошибок переменных |
| ErrorTextUpdate | string | Проверяет словарь ошибок и пустые строки, возвращает текст ошибки или ее отсутствия |
| Input\_InputChange | - | Активируется при изменении текста в любом объекте «TextBox» или в «RadioButton», вызывает проверки входных параметров и их изменение |

Таблица 3.4 – Описание методов/свойств класса ScrewdriverBuilder

|  |  |  |
| --- | --- | --- |
| Название метода/свойства | Тип возвращаемого значения/хранимое значение | Описание |
| \_wrapper | KompasWrapper | Хранит в себе элемент связи с КОМПАС 3D |
| BuildRod | - | Создает модель стержня отвертки при помощи словаря входных параметров |
| BuildHandle | - | Создает модель рукоятки отвертки при помощи словаря входных параметров, а затем соединяет получившиеся детали |

## 3.2 Макеты пользовательского интерфейса

При запуске программы пользователю откроется доступ к интерфейсу, представленному на рисунке 3.2:

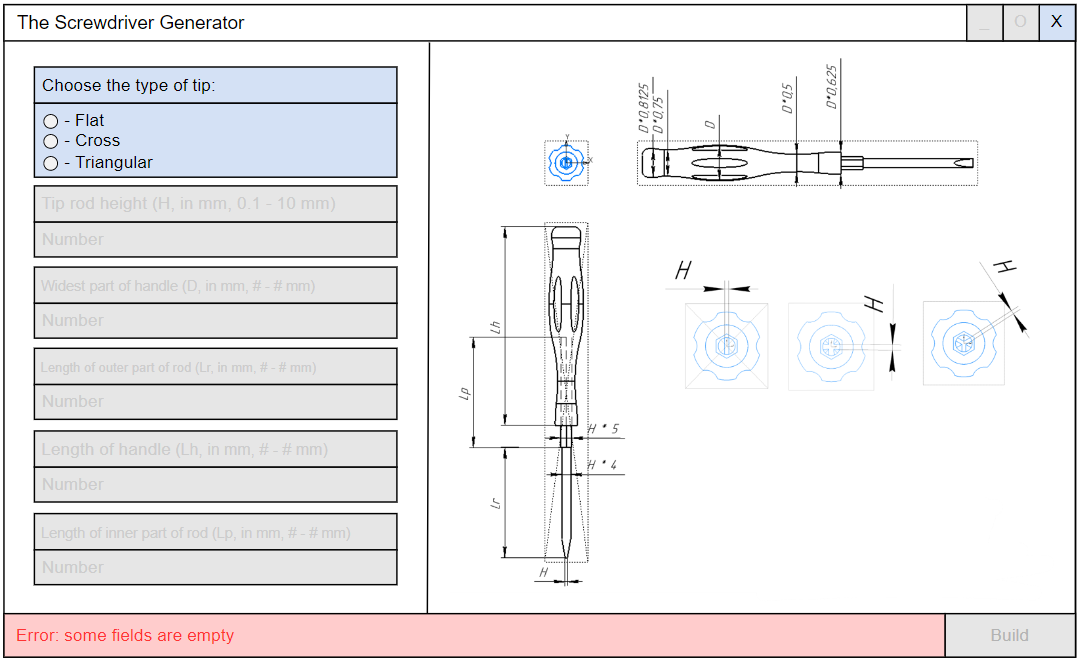


Рисунок 3.2 – Изначальный вид окна приложения

В нем пользователю предоставляется возможность оценить доступные для изменения переменные посредством изучения чертежа будущей отвертки. На основе этих знаний, он сможет в строго определенном порядке изменять входные данные для программы.

Программа имеет условные цветовые обозначения, для придания интуитивности в ее использовании: светло-серый цвет, к примеру, показывает, что это интерактивная часть программы, с которой на данный момент нельзя взаимодействовать. Пастельно-синий – интерактивная часть, с которой можно взаимодействовать. Красный цвет – цвет ошибки, а белый – не интерактивная часть.

При вводе первого необходимого данного – типа наконечника, выбор которых будет осуществляться через элемент «Radio-Button», к изменению будет доступно второе значение – высота наконечника стержня отвертки (значение H). Все данные, которые должны вноситься в программу, должны быть указаны в миллиметрах, а границы этих значений строго определены. Эта дополнительная информация отображена в скобках к описанию вводимых данных.

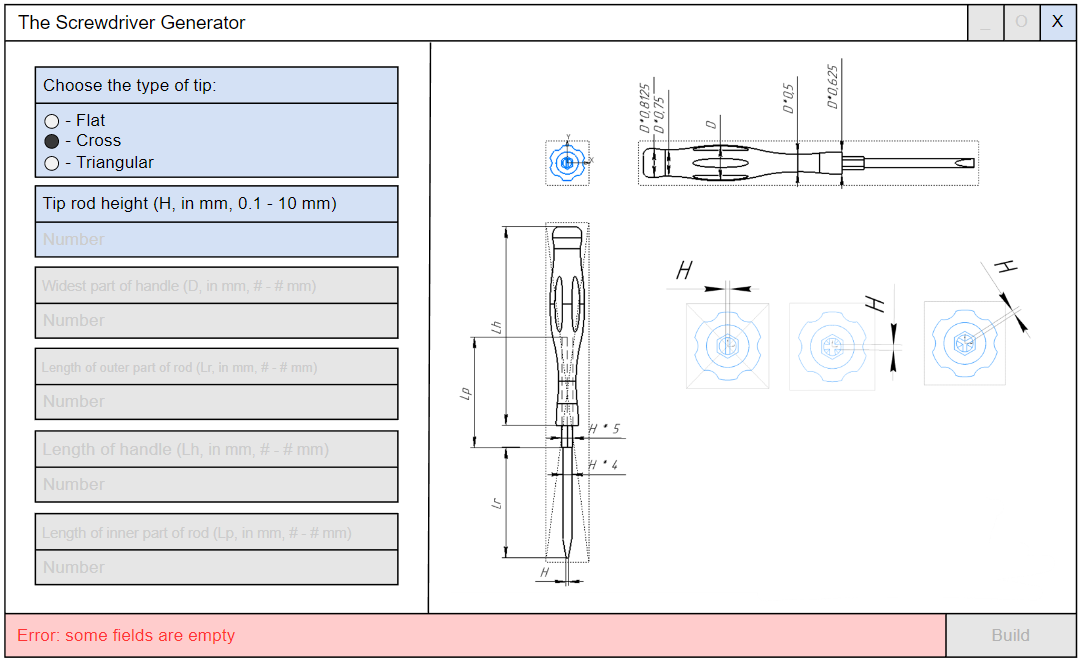


Рисунок 3.3 – Окно приложения после выбранного вида наконечника

Если после выбора вида наконечника пользователь введет допустимое значение H, то ему будет доступно для заполнения следующее окно ввода (причем, текст в описании третьего поля ввода поменяется, так как границы этого значения зависят от предыдущего), и так далее. Если пользователь ошибется, то поле ввода подсветится красным, а поле ошибок выведет, что именно пользователь сделал не так:

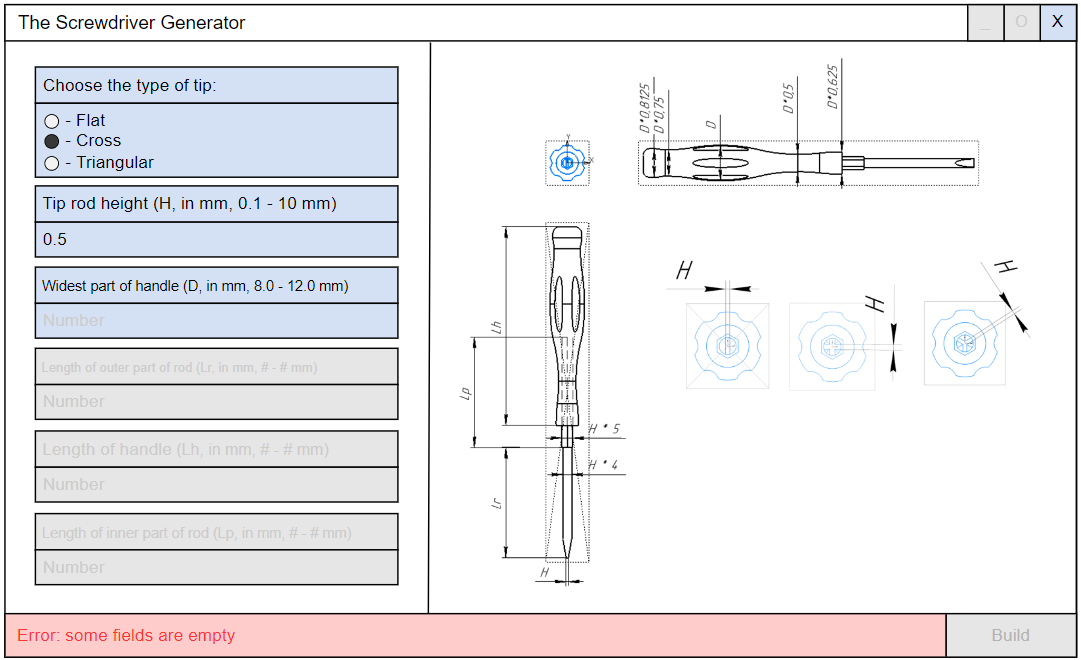


Рисунок 3.4 – Окно приложения после корректно введенного значения H

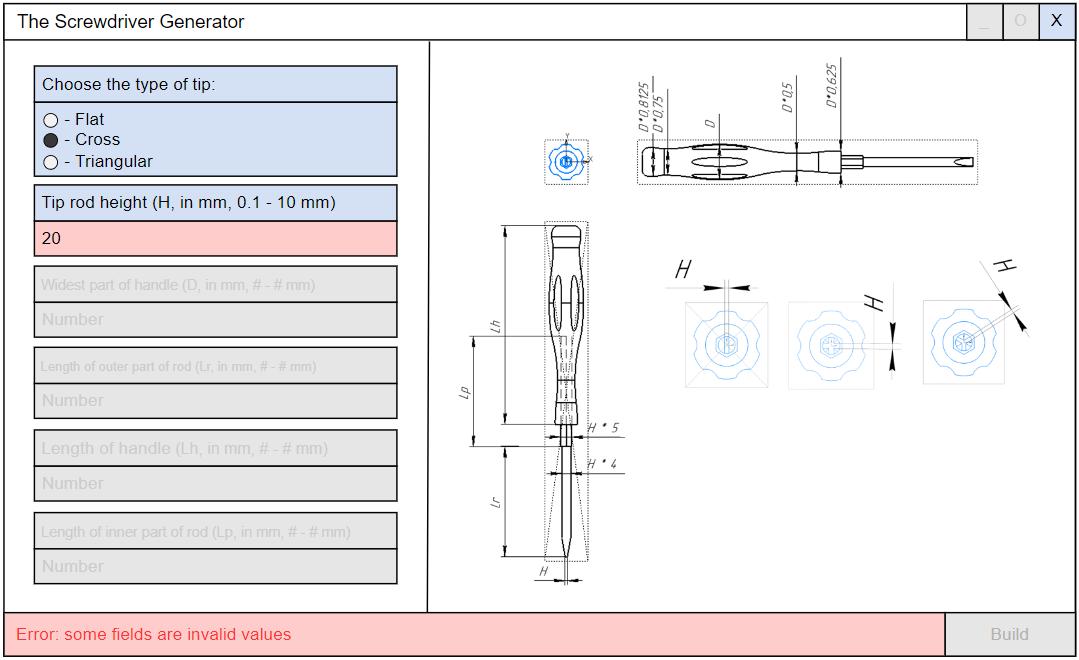


Рисунок 3.5 – Окно приложения после некорректно введенного значения H

Когда все входные параметры будут введены верно, окно ошибок перестает быть красным, а кнопка «Build» становится доступной для нажатия:

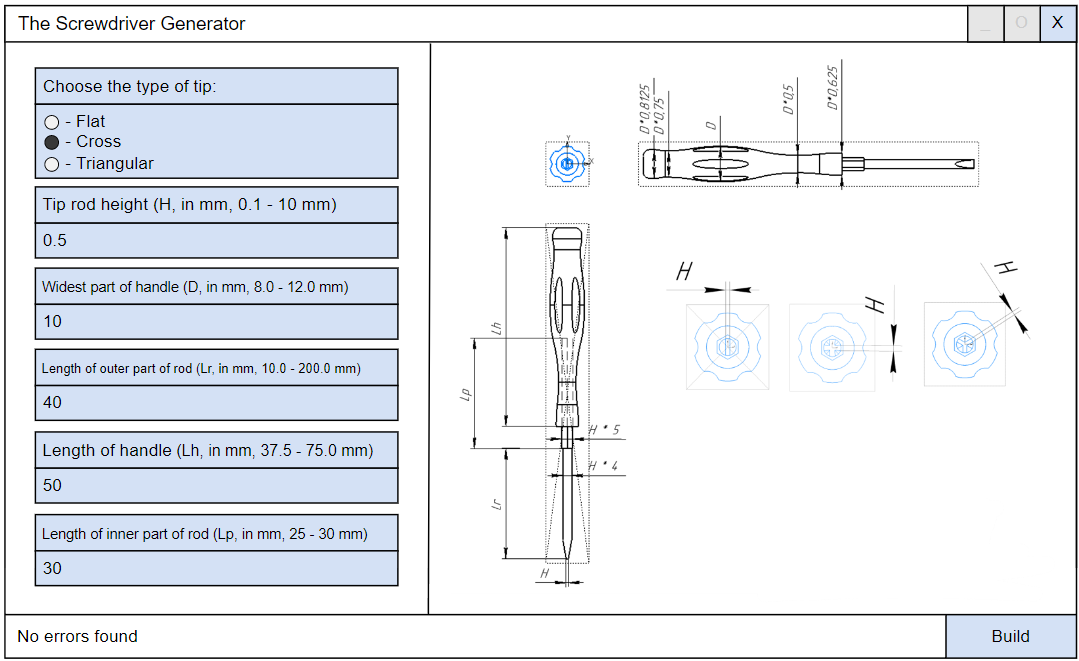


Рисунок 3.6 – Окно приложения после корректного введения всех значений

Если же вдруг пользователь сменит один из входных параметров, который перепишет границы других параметров так, что они не будут в установленных пределах, то тогда неверные переменные засветятся красным, зависимые от них переменные станут серыми и недоступными к правке.

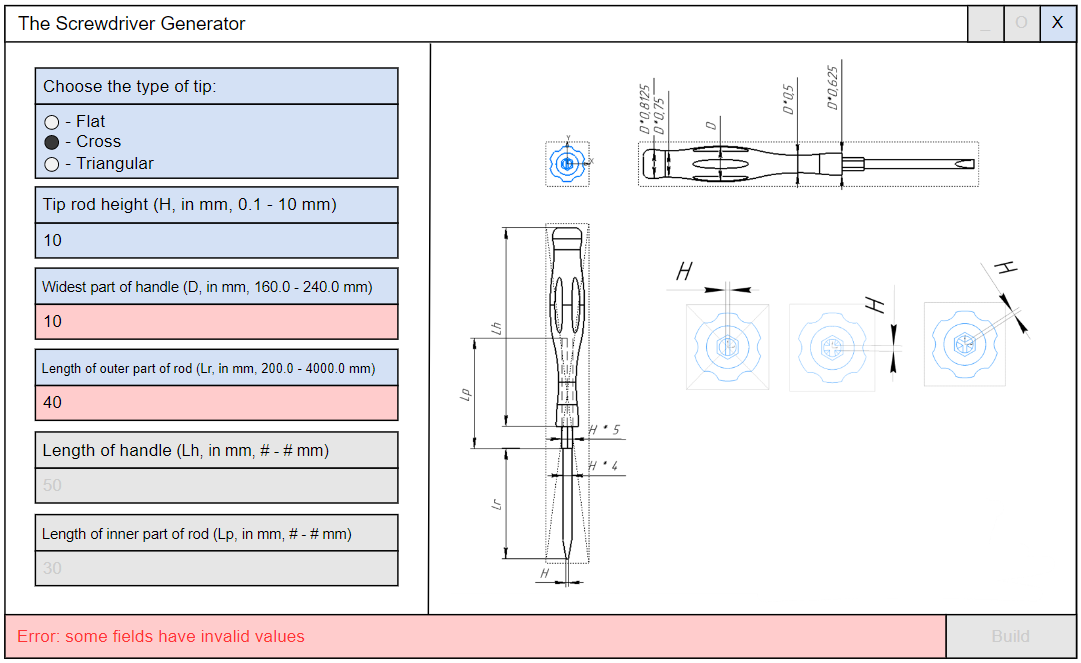


Рисунок 3.7 – Реакция программы после изменения H на некорректный

# Список использованных источников

1. Компас (САПР) // Википедия URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D1%81\_(%D0%A1%D0%90%D0%9F%D0%A0) (дата обращения: 13.10.2022).

2. BookGen // Oliver Weissbarth Software Engineer, 3D-Graphics Enthusiast URL: https://www.oweissbarth.de/software/bookgen/ (дата обращения: 13.10.2022).

3. Bolt Factory // Blender 3.3 Manual URL: https://docs.blender.org/manual/en/latest/addons/add\_mesh/boltfactory.html (дата обращения: 13.10.2022).

4. Unified Modeling Language // Wikipedia URL: https://en.wikipedia.org/wiki/Unified\_Modeling\_Language (дата обращения: 13.10.2022).