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 «Корпус спиннера» ДЛЯ САПР «КОМПАС-3D 2022»**

Проект системы по дисциплине

«Основы разработки САПР»

Выполнил:

Студент гр. 589-3

\_\_\_\_\_\_\_\_\_ Высоцкая О.В.

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ Калентьев А.А.

(оценка) \_\_\_\_\_\_\_\_\_

Томск 2022 г

Оглавление

[1 ОПИСАНИЕ САПР 3](#_Toc120237614)

[1.1 Описание САПР «КОМПАС-3D v19» 3](#_Toc120237615)

[1.2 Анализ API 4](#_Toc120237616)

[1.3 Обзор аналогов 10](#_Toc120237617)

[2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ 12](#_Toc120237618)

[3 ПРОЕКТ СИСТЕМЫ 14](#_Toc120237619)

[3.1 Диаграмма классов 14](#_Toc120237620)

[3.2 Макеты пользовательского интерфейса 17](#_Toc120237621)

[Список использованных источников 19](#_Toc120237622)

# 1 ОПИСАНИЕ САПР

# 1.1 Описание САПР «КОМПАС-3D v21»

Система автоматизированного проектирования [1] — автоматизированная система, реализующая информационную технологию выполнения функций проектирования, представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности.

**КОМПАС-3D v21** — 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 [2].

# 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 [3].

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.

Далее будут приведены самые важные для использования в программе методы и свойства интерфейсов. Наиболее вероятно, программа не ограничится их использованием, а будет также применять более локальные методы и свойства:

Таблица 1.1 – Интерфейсы, используемые при разработке

|  |  |
| --- | --- |
| Название интерфейса | Описание интерфейса |
| KompasObject | Интерфейс API КОМПАС |
| ksEntity | Интерфейс элемента модели (оси, плоскости, формообразующего элемента) |
| ksDocument2D | Интерфейс графического документа системы КОМПАС |
| ksSketchDefinition | Интерфейс параметров эскиза |
| ksDocument3D | Интерфейс документа-модели |
| ksPart | Интерфейс детали или подсборки в составе сборки |
| ksBaseExtrusionDefinition | Интерфейс параметров основания - элемента выдавливания |
| ksCircularPartArrayDefinition | Интерфейс операции копирования по окружности |

В нижеописанных таблицах представлены методы, которые будут использоваться при разработке плагина, а также описание входных параметров данных методов (таблицы 1.2 – 1.15).

Таблица 1.2 – Используемые методы интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| ActivateControllerAPI() | bool | Метод для активации API КОМПАС-3D |
| Visible() | bool | Свойство видимости приложения |

Таблица 1.3 – Используемые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |

Таблица 1.4 – Используемые методы интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| ksCircle (double xc, double yc, double rad, long style) | Ссылка на окружность – в случае удачного завершения, 0 – в случае неудачи | Создать окружность |

Таблица 1.5 – Описание входных параметров, используемых методов интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| ksCircle (double xc, double yc, double rad, long style) | xc, yc | Координаты центра окружности |
| rad | Радиус окружности |
| style | Стиль линии |

Таблица 1.6 – Используемые методы интерфейса ksSketchDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| BeginEdit() | bool | Войти в режим редактирования эскиза (ksDocument2D) |
| EndEdit() | bool | Выйти из режима редактирования эскиза |

Таблица 1.7 – Используемые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Create (bool invisible, bool typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| GetPart (int type) | ksPart | Получить ссылку на интерфейс компонента в соответствии с заданным типом |

Таблица 1.8 – Описание входных параметров, используемых методов интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| Create (bool invisible, bool typeDoc) | invisible | Признак режима редактирования документа (TRUE – невидимый режим, FALSE – видимый режим) |
| typeDoc | Тип документа (TRUE – деталь, FALSE – сборка) |
| GetPart (int type) | type | Тип компонента из перечисления: pInPlace\_Part – компонент, редактируемый на месте; pNew\_Part – новый компонент; pEdit\_Part –редактируемый компонент; pTop\_Part – главный компонент, в составе которого находится новый или редактируемый или указанный компонент (например, сборка, в составе которой находится редактируемая деталь) |

Таблица 1.9 – Используемые методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| GetDefaultEntity (short objType) | ksEntity | Получить ссылку на интерфейс объекта, создаваемого системой по умолчанию |
| NewEntity (short objType) | ksEntity | Создать новый интерфейс объекта и получить ссылку на него |

Таблица 1.10 – Описание входных параметров, используемых методов интерфейса ksPart

|  |  |
| --- | --- |
| Входной параметр | Описание параметра |
| objType | Тип объекта |

Таблица 1.11 – Используемые типы объектов в методах интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Метод | Тип объекта | Название объекта |
| GetDefaultEntity (short objType) | o3d\_planeXOY | Плоскость XOY |
| o3d\_axisOZ | Ось OZ |
| NewEntity (short objType) | o3d\_sketch | Эскиз |
| o3d\_baseExtrusion | Базовая операция выдавливания |
| o3d\_cutExtrusion | Вырезать выдавливанием |
| o3d\_circularCopy | Операция копирования по концентрической сетке |

Таблица 1.12 – Используемые методы интерфейса ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | bool | Установить параметры выдавливания в одном направлении |
| SetSketch (LPDISPATCH sketch) | bool | Задать ссылку на интерфейс эскиза элемента |

Таблица 1.13 – Описание входных параметров, используемых методов интерфейса ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | forward | Направление выдавливания: TRUE - прямое направление, FALSE - обратное направление |
| type | Тип выдавливания |
| depth | Глубина выдавливания |
| draftValue | Угол уклона |
| draftOutward | Направление уклона: FALSE – уклон наружу, TRUE – уклон внутрь |
| SetSketch (LPDISPATCH sketch) | sketch | Указатель на интерфейс эскиза ksEntity |

Таблица 1.14 – Используемые методы интерфейса ksCircularPartArrayDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetCopyParamAlongDir (long count, double step, bool, factor, bool dir) | bool | Установить параметры копирования |
| SetAxis (LPDISPATCH axis) | bool | Установить указатель на ось копирования |

Таблица 1.15 – Описание входных параметров, используемых методов интерфейса ksCircularPartArrayDefinition

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание входного параметра |
| SetCopyParamAlongDir (long count, double step, bool factor, bool dir) | count | Количество копий |
| step | Шаг |
| factor | Признак полного шага |
| dir | Направление |
| SetAxis (LPDISPATCH axis) | axis | Указатель на интерфейс оси ksEntity |

# 1.3 Обзор аналогов

SolidWokrs (cолидворкc, от англ. solid — твёрдое тело и англ. works — работать) — программный комплекс САПР для автоматизации работ промышленного предприятия на этапах конструкторской и технологической подготовки производства. Обеспечивает разработку изделий любой степени сложности и назначения [4].

Программу начали разрабатывать в 1993 году, она начала продаваться в 1995 и составила конкуренцию таким продуктам, как AutoCAD и Autodesk Mechanical Desktop, SDRC I-DEAS, Компас-3D и Pro/ENGINEER [4].

Принцип построение корпуса спиннера и в Компас-3D, и в SolidWorks очень похожи [5].

Принцип построение корпуса спиннера в SolidWorks показан на рисунке 1.1.

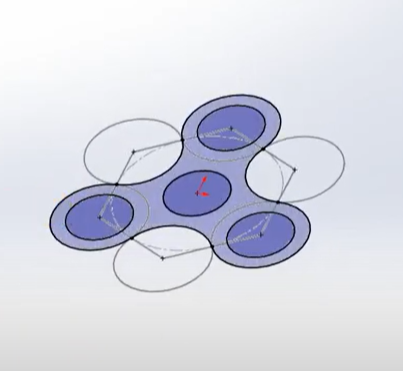


Рисунок 1.1 – Принцип построение корпуса спиннера в SolidWorks

# 2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ

Спиннер (англ. fidget spinner, hand spinner) — развлекательная вращающаяся игрушка. В центре спиннера находится металлический или керамический подшипник, радиально расположены несколько лопастей/крыльев или утяжелителей. [6]

Схема корпуса спиннера представлена на рис. 2.1:

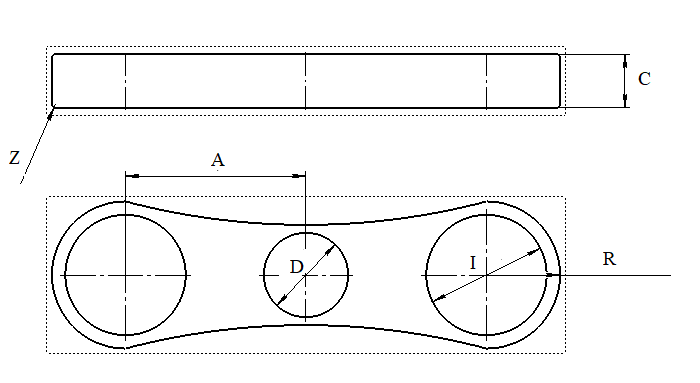


Рисунок 2.1 – Чертёж модели «Корпус спинера»

Измеряемые параметры для плагина:

1. диаметр внутренних колец спинера ***I*** (мин – 30 мм, макс – 60 мм);
2. толщина корпуса спинера ***С*** (мин – 10 мм, макс – 30 мм);
3. длина спинера ***A*** (зависит от диаметра внутренних колец: мин – I (мин) \* 1.5, макс – I (макс) \* 1.5);
4. диаметр центрального кольца спинера ***D*** (зависит от диаметра внутренних колец: мин – I (мин) \* 0.7, макс – I (макс) \* 0.7);
5. радиус внешних колец ***R*** (зависит от диаметра внутренних колец: мин – (I (мин) + 10) / 2, макс – (I (макс) + 10) / 2);
6. скругление (радиус) корпуса спинера и его колец ***Z*** (мин – 0.5 мм, макс – 2 мм).

# 3 ПРОЕКТ СИСТЕМЫ

# 3.1 Диаграмма классов

Унифицированный язык моделирования (UML – unified modeling language) — 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 [7].

Всего UML содержит 12 видов диаграмм, одним из которых являются диаграммы классов. UML-диаграммы классов описывают такие пользовательские типы данных, как классы, структуры и перечисления, а также связи между ними. Фактически, диаграммы классов представляют архитектуру программы в лаконичной графической форме, удобной для анализа и проектирования.

Диаграмма классов UML представлена на рисунке 3.1.

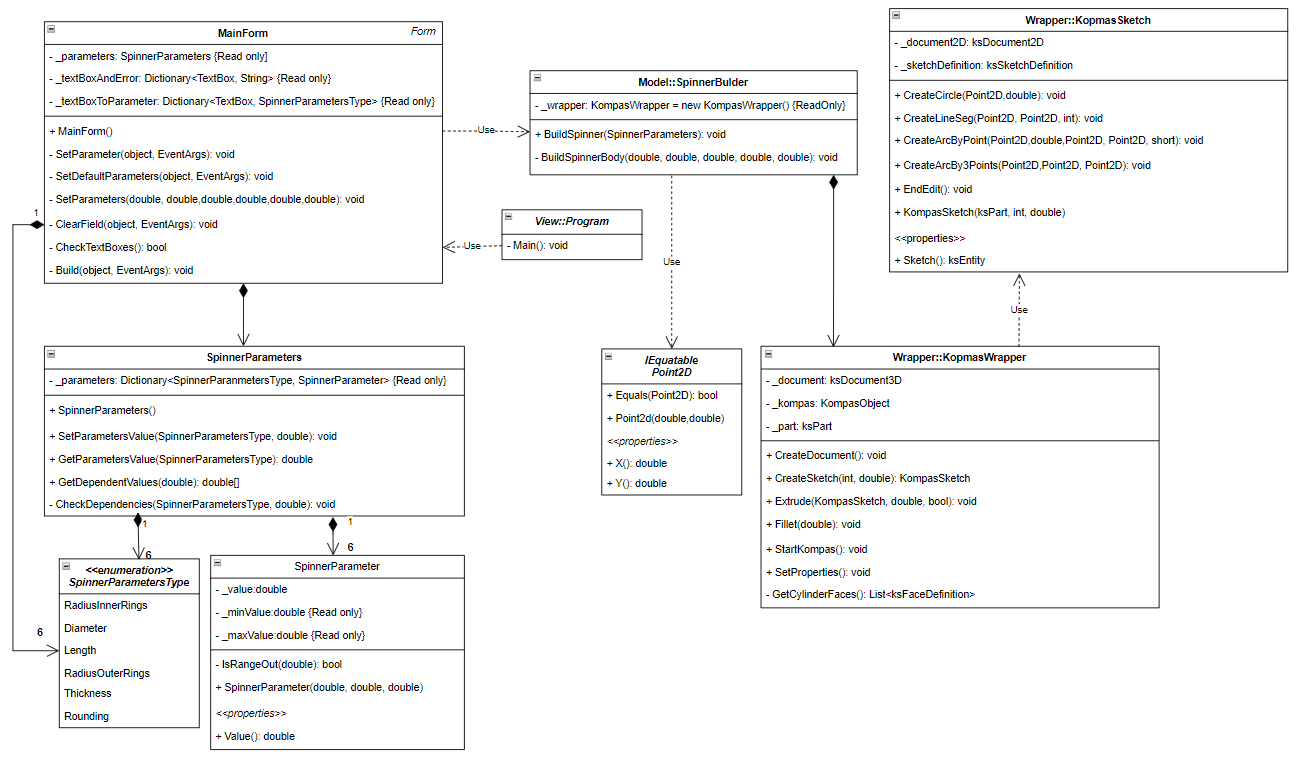


Рисунок 3.1 – Диаграмма классов UML

Далее в таблицах 3.1 – 3.8 представлено описание классов.

Таблица 3.1 – Описание полей, методов, сущностей класса «MainForm»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Словарь, который хранит в себе набор методов для построения корпуса спиннера |
| \_textBoxAndError |  | Словарь, который хранит в себе набор TextBox и строку |
| \_textBoxToParameter |  | Словарь, который хранит в себе набор TextBox и соответствующий ему тип параметра из перечисления «SpinnerParametersType» |
| SetParameter(object, EventArgs) | void | Устанавливает значение параметра |
| SetDefaultParameters(double, double, double, double, double, double) | void | Устанавливает значения параметров в TextBox |
| SetParameters(object, EventArgs) | void | Устанавливает значения параметров для кнопок (мин, ср, макс) |
| CheckTextBoxes() | void | Проверка текстовых полей на корректность ввода параметров |
| ClearField(object, EventArgs) | void | Очищает TextBox |
| Build (object, EventArgs) | void | Строит корпус спиннера по заданным параметрам |

Таблица 3.2 – Описание полей, методов, сущностей класса «SpinnerParameters»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Словарь, который хранит данные о каждом параметре модели из перечисления «SpinnerParametersType» |
| SpinnerParameters() |  | Конструктор для создания экземпляра класса |
| SetParameterValue(SpinnerParameterType, double) | void | Устанавливает значение определённого параметра |
| GetParameterValue(SpinnerParameterType) | double | Возвращает значение определённого параметра |
| GetDependentValue(double) | double[] | Возвращает значения зависимых параметров |
| CheckDependencies(SpinnerParameterType, double) | void | Проверка зависимых параметров |

Таблица 3.3 – Описание полей, методов, сущностей класса «SpinnerParameter»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_value |  | Хранит текущее значение |
| \_minValue |  | Хранит минимальное значение |
| \_maxValue |  | Хранит максимальное значение |
| SpinnerParameter(double, double, double) |  | Конструктор для создания экземпляра класса |
| Value() | double | Возвращает текущее значение параметра. Задает новое значение параметра |
| IsRangeOut(double) | bool | Проверка корректности введенных значений параметров |

Таблица 3.4 – Описание полей, методов, сущностей класса «SpinnerBuilder»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_wrapper |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| BuildSpinner(SpinnerParameters) | void | Построение корпуса спинера по заданным параметрам |
| BuildSpinnerBody(double, double, double, double, double) | void | Построение тела корпуса спиннера |

Таблица 3.5 – Описание полей, методов, сущностей класса «KompasWrapper»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_document |  | Документ 3D |
| \_kompas |  | Компас 3D объект |
| \_part |  | Деталь |
| StartKompas() | void | Запускает Компас 3D |
| CreateDocument() | void | Создает документ 3D |
| SetProperties() | void | Задает свойства детали: цвет и имя |
| CreateSketch(int, double) | KopmasSketch | Создает эскиз |
| Extrude(KompasSketch, double, bool) | void | Выдавливание эскиза |
| Fillet(double) | void | Скругляет края |
| GetCylinderFaces() | List<ksFaceDefinition> | Возвращает все цилиндрические грани детали. |

Таблица 3.6 – Описание полей, методов, сущностей класса «KompasSketch»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_document2D |  | Документ 2D |
| \_sketchDefinition |  | Параметры эскиза |
| Sketch() | ksEntity | Возвращает эскиз |
| KompasSketch(ksPart, int, double) |  | Конструктор эскиза |
| EndEdit() | void | Закрывает эскиз |
| CreateCircle(Point2D, double, int) | void | Создает окружность в эскизе |
| CreateLineSeg(Point2D, Point2D, int) | void | Создает линию в эскизе |
| CreateArcBy3Points(Point2D, Point2D, Point2D) | void | Создает дугу по трем точкам в эскизе |
| CreateArcByPoint(Point2D, double, Point2D, Point2D, short) | void | Создает дугу по центру и двум конечным точкам в эскизе |

Таблица 3.7 – Описание полей класса «SpinnerParametersType»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| DiameterInnerRings |  | Диаметр внутренних колец |
| Diameter |  | Диаметр центрального кольца |
| Length |  | Длина корпуса спиннера |
| RadiusOuterRings |  | Радиус внешних колец |
| Thickness |  | Толщина корпуса спиннера |
| Rounding |  | Скругление |

Таблица 3.8 – Описание полей, методов, сущностей класса «Point2D»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| X | double | Возвращает значение оси x |
| Y | double | Возвращает значение оси y |
| Point2D(double, double) |  | Конструктор Point2D |
| Equals(Point2D) | bool | Проверка на равенство объектов класса |

# 3.2 Макеты пользовательского интерфейса

На рисунке 3.2 представлен макет пользовательского интерфейса.

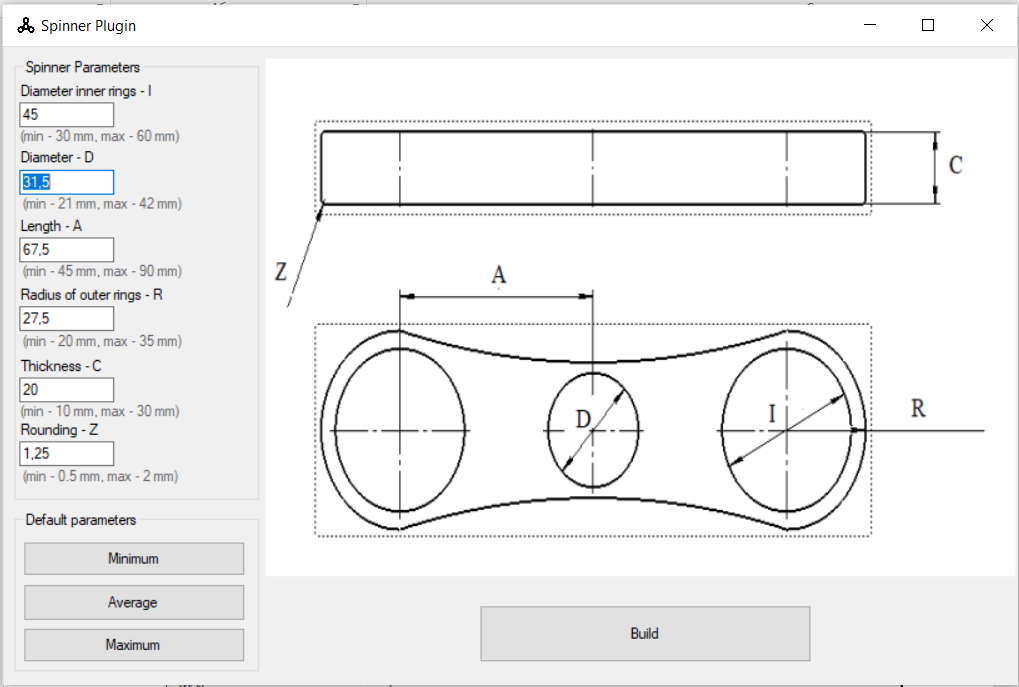


Рисунок 3.2 – Макет пользовательского интерфейса

С помощью данного окна пользователь может изменять параметры будущей 3D модели корпуса спиннера.

Сверху полей ввода находится название компонента, за который поле отвечает, а снизу корректные размеры – минимальный, средний и максимальный.

При нажатии на кнопку «Minimum» будет создана 3D модель с минимальными корректными размерами. При нажатии на кнопку «Average» будет создана 3D модель со средними корректными размерами. При нажатии на кнопку «Maximum» будет создана 3D модель с максимальными корректными размерами. При нажатии на кнопку «Build» будет построена 3D модель по заданным параметрам. Чертёж модели справа необходим для лучшего понимания расположения вводимых размеров.

На рисунке 3.3 представлено окно с некорректно введёнными данными.

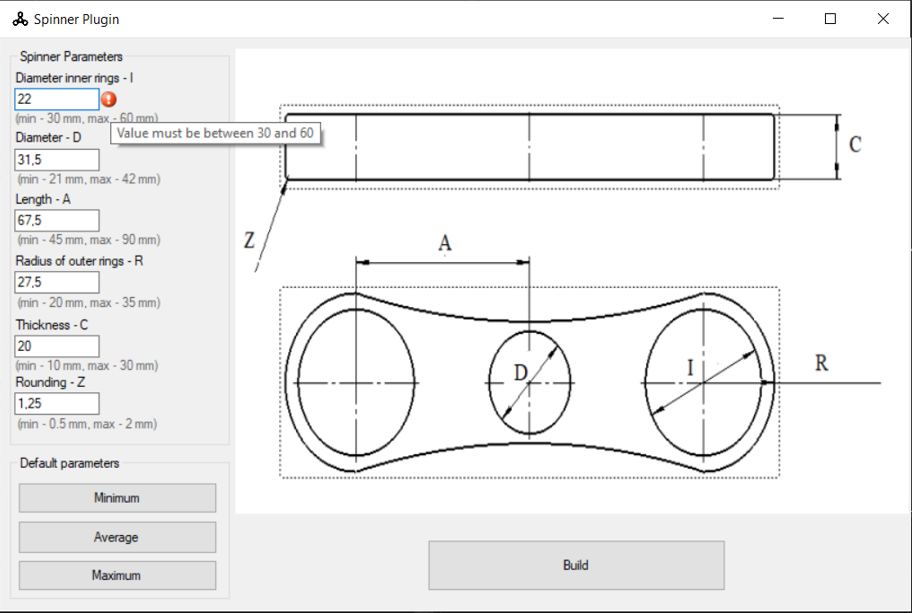


Рисунок 3.3 – Макет пользовательского интерфейса с некорректным вводом данных

В случае некорректного ввода появится соответствующее уведомление о некорректности введенного значения, в котором указывается в каком диапазоне необходимо ввести значение. Если значения введены неверно, кнопка «Build» выведет сообщение о некорректном значении.

# Список использованных источников

1. САПР — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 20.10.2022).

2. КОМПАС 3D — Википедия. [Электронный ресурс]. — Режим доступа: 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 20.10.2022).

3. API — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 20.10.2022).

4. SolidWorks. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/SolidWorks (дата обращения 20.10.2022).

5. Построение спинера в SolidWorks. [Электронный ресурс]. — Режим доступа: https://www.infoconnector.ru/hobby/projects/spinner-postroenie-v-solidworks (дата обращения 20.10.2022).

6. Спиннер — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Спиннер (дата обращения 20.10.2022).

7. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 20.10.2022).