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 «Рулевая тяга» ДЛЯ САПР «КОМПАС-3D 2022»**

Проект системы по

по дисциплине «ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 580-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.С. Рыбин

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

Руководитель:

к.т.н., доцент каф. КСУП

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

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

**Оглавление**

[1 Описание САПР 3](#_Toc116637014)

[1.1 Описание программы 3](#_Toc116637015)

[1.2 Описание API 3](#_Toc116637016)

[1.3 Обзор аналогов 7](#_Toc116637017)

[2 Описание предмета проектирования 8](#_Toc116637018)

[3 Проект программы 9](#_Toc116637019)

[3.1 Описание технических и функциональных аспектов проекта 9](#_Toc116637020)

[3.2 Диаграмма классов 10](#_Toc116637021)

[3.3 Макет пользовательского интерфейса 14](#_Toc116637022)

[Список литературы 17](#_Toc116637023)

# 1 Описание САПР

# Описание программы

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей (в том числе, деталей, формируемых из листового материала путём его гибки) и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология позволяет быстро получать модели типовых изделий на основе проектированного ранее прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства.

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. [2]

# Описание 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 существует API под названием Kompas6API5. Для построения объекта в Компас через API будут использоваться следующие интерфейсы: ksExtrusionParam (таблица 1.1), ksCutExtrusionDefinition (таблица 1.2), ksBossExtrusionDefinition (таблица 1.2), ksEntity (таблица 1.3), KompasObject (таблица 1.4), ksDocument3D (таблица 1.5), ksDocument2D (таблица 1.6), ksSketchDefinition (таблица 1.7), ksPart (таблица 1.8).

Таблица 1.1 — Используемые свойства интерфейса ksExtrusionParam

|  |  |
| --- | --- |
| Название | Описание |
| depthReverse | Глубина выдавливания в обратном направлении |
| depthNormal | Глубина выдавливания в прямом направлении |

Таблица 1.2 — Используемые методы, и свойства интерфейса ksCutExtrusionDefinition и ksBossExtrusionDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| directionType |  |  | Свойство задающее направление выдавливания |
| SetSketch | sketch ссылка на интерфейс эскиза ksEntity | TRUE в случае успешного завершения,  FALSE в случае неудачи. | Задать ссылку на интерфейс эскиза элемента |
| ExtrusionParam |  | Ссылка на интерфейс ksExtrusionParam или  IExtrusionParam. | Получить ссылку на интерфейс параметров  элемента выдавливания |

Таблица 1.3 — Используемые методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| GetDefinition | Ссылка на интерфейс IDispatch | Получить ссылку на интерфейс параметров объектов и элементов |
| Create | TRUE в случае успешного завершения. | Создать объект в модели |

Таблица 1.4 — Используемые методы интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| ActivateControllerAPI | TRUE в случае успешного завершения. | Метод для активации API Компас-3D |
| Document3D | Ссылка на интерфейс ksDocument3D | Получить ссылку на интерфейс  документа трехмерной модели |

Таблица 1.5 — Используемые методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| Create | TRUE в случае успешного завершения. | Создать объект в модели |

Таблица 1.6 — Используемые методы интерфейса ksDocument2D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| ksLineSeg | x1, y1 координаты первой точки отрезка,  x2, y2 координаты второй точки отрезка,  style стиль линии. | Ссылка на отрезок | Создать отрезок |
| Create |  | TRUE в случае успешного завершения. | Создать объект в модели |

Таблица 1.7 — Используемые методы интерфейса ksSketchDefinition

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| SetPlane | plane ссылка на интерфейс базовой плоскости эскиза  ksEntity или IEntity | TRUE в случае успешного завершения. | Изменить базовую плоскость эскиза |
| BeginEdit | ссылка на интерфейс эскиза ksDocument2D | TRUE в случае успешного завершения. | Войти в режим редактирования эскиза |
| EndEdit |  | TRUE в случае успешного завершения. | Выйти из режима редактирования эскиза |

Таблица 1.8 — Используемые методы интерфейса ksPart

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| NewEntity | objType – тип объекта | Ссылка на интерфейс ksEntity | Создать новый интерфейс объекта и получить ссылку на него работать с плоскостью |
| GetDefaultEntity | objType – тип объекта | Ссылка на интерфейс ksEntity | Получить ссылку на интерфейс  объекта, создаваемого системой по умолчанию |

# Обзор аналогов

**«nanoCAD Механика»**

«nanoCAD Механика» является решением для машиностроительного проектирования, базирующимся на платформе «nanoCAD Plus» от российского разработчика – компании «Нанософт». Помимо богатого функционала для оформления конструкторской и технологической документации по ЕСКД и ЕСТД, программа располагает всеми основными инструментами цифрового проектирования. [4]

Модуль «Механика» имеет широкую библиотеку стандартных деталей, в том числе и штоков, выполненных по ГОСТ, и поддерживает возможность изменять параметры детали под необходимые нужды.

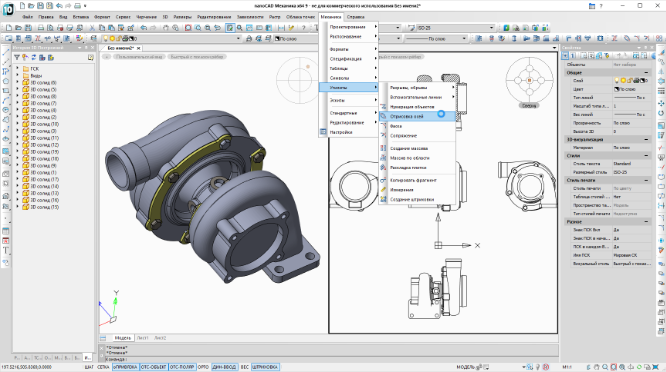


Рисунок 1.1 — Классический интерфейс «nanoCAD Механика»

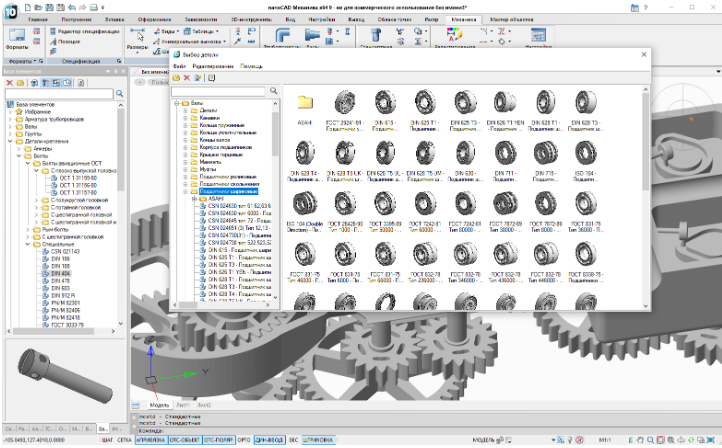


Рисунок 1.2 — Каталог стандартных деталей в nanoCAD

# 2 Описание предмета проектирования

Рулевая тяга — элемент привода механизма рулевого управления колесных транспортных средств, деталь в виде стержня, обеспечивающая передачу усилия от рулевого механизма на рычаги поворотных кулаков колес.[5]

На рисунке 2.1 представлен чертеж рулевой тяги.

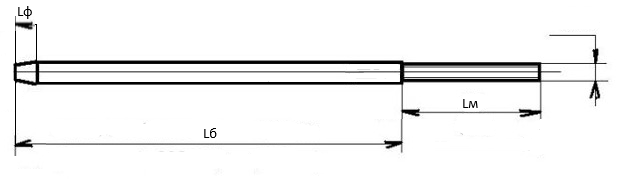


Рисунок 2.1 – Чертеж рулевой тяги

Параметры рулевой тяги:

1. Длина большой части Lб (от 300 мм до 380мм)
2. Длина малой части Lм – не больше 1/2 Lб, не меньше 1/4 (от 75 мм до 190мм)
3. Диаметр большой части Dб (от 20 мм до 30 мм)
4. Диаметр малой части Dм – не больше 70% Dб, не меньше 50% Dб (от 10 мм до 21 мм)
5. Длина фаски малой части Lф – от 7 мм до 10 мм, угол остается неизменным.

# 3 Проект программы

# 3.1 Описание технических и функциональных аспектов проекта

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML.

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML – моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот.[6]

При использовании UML были простроена диаграмма классов.

# 3.2 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.[7]

На рисунке 3.1 представлена диаграмма классов.

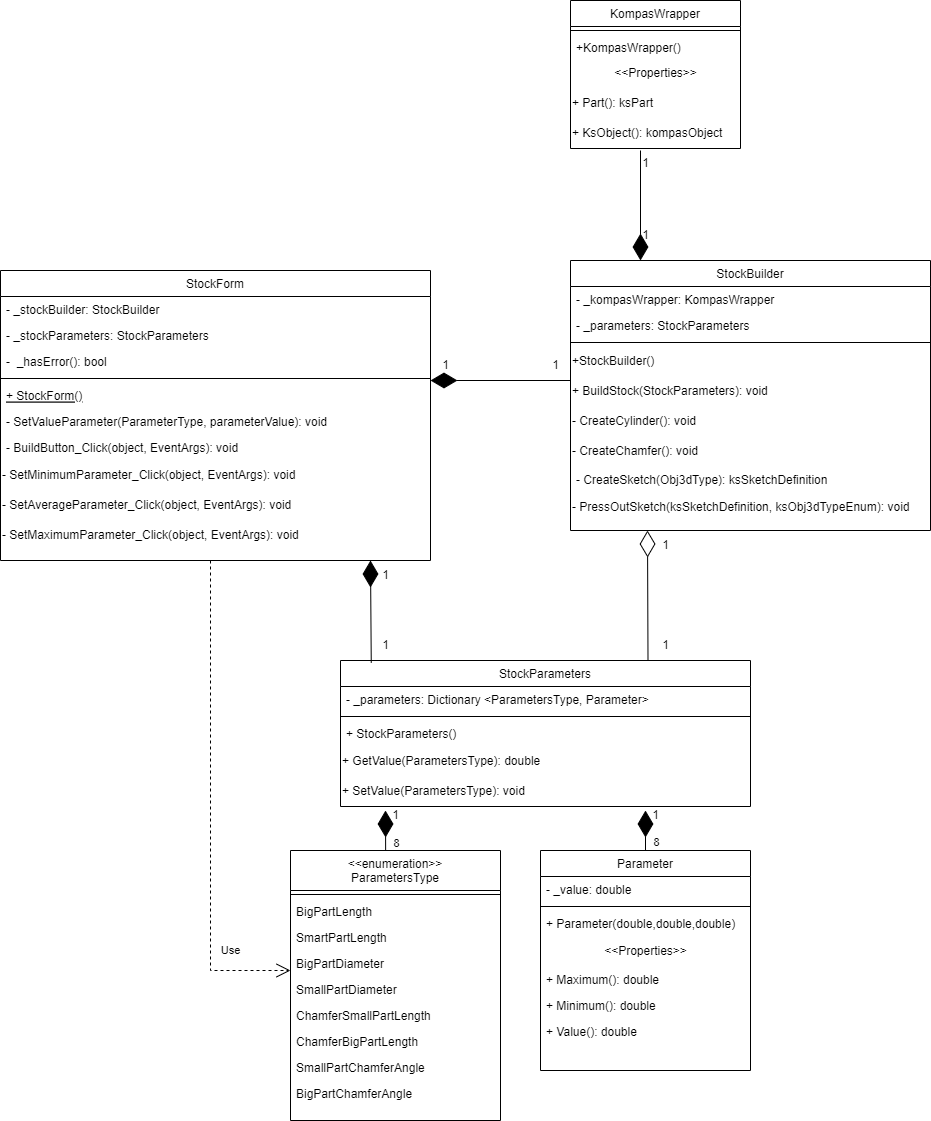


Рисунок 3.1 – Диаграмма классов

Краткое описание классов представлено в таблицах 3.1 – 3.4

Таблица 3.1 – Описание класса StockForm

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_stockBuilder: StockBuilder |  | Хранит экземпляр класса-построителя |
| \_stockParameters: StockParameters |  | Хранит экземпляр списка параметров детали |
| -SetValueParameter(ParameterType, parameterValue) | void | Устанавливает значение параметра |
| -SetMinimumParameters\_Click() | void | Устанавливает минимальное значение всех параметров |
| * SetMaximumParameters() | void | Устанавливает максимальное значение всех параметров |
| - SetAverageParameters() | void | Устанавливает среднее значение всех параметров |

Таблица 3.2 – Описание класса StockParameters

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters: Dictionary <ParametersType, Parameter> |  | Хранит словарь с ключом из перечисления ParametersType и значением Parameter |
| StockParameters() |  | Конструктор для создания экземпляра класса |
| SetValue(ParameterType, double) | void | Устанавливает значение определённого параметра |
| GetValue(ParameterType) | double | Возвращает значение определённого параметра |

Таблица 3.3 – Описание класса Parameter

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_value: double |  | Хранит текущее значение |
| Parameter(double, double, double) |  | Конструктор для создания экземпляра класса |
| Maximum() | double | Возвращает максимальное допустимое значение параметра |
| Minimum() | double | Возвращает минимальное допустимое значение параметра |
| Value() | double | Возвращает текущее значение параметра. Задает новое значение параметра |

Таблица 3.4 – Описание класса StockBuilder

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| * kompasWrapper: KompasWrapper |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| * \_parameters: StockParameters |  | Хранит данные о каждом параметре модели из перечисления «StockParameter» |
| + BuildStock(StockParameters) | void | Построение штока по заданным параметрам |
| * CreateCylinder() | void | Построение цилиндрической основы штока |
| * CreateChamfer() | void | Построение фаски |

Продолжение таблицы 3.4

|  |  |  |
| --- | --- | --- |
| CreateSketch(obj3dType) | ksSketchDefinition | Возвращает интерфейс параметров эскиза |
| PressOutSketch(ksSketchDefinition, double) | void | Выдавливает эскиз |

# 3.3 Макет пользовательского интерфейса

Макет пользовательского интерфейса представляет собой форму для ввода параметров штока. Построение модели осуществляется путем нажатия на кнопку «Build». Если какое-то из полей не будет проходить проверку, то кнопка «Build» будет неактивна.

На рисунке 3.2 представлен макет пользовательского интерфейса.

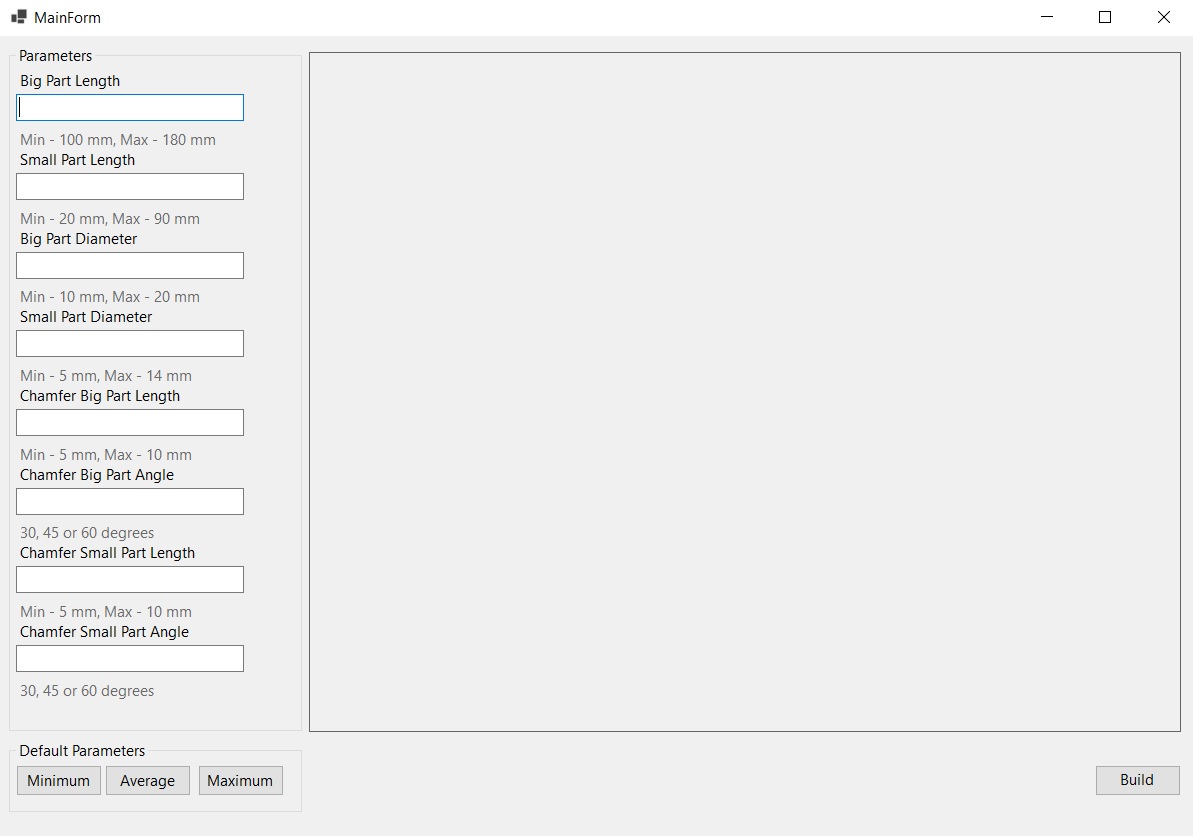


Рисунок 3.2 – Макет пользовательского интерфейса

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

Напротив полей ввода находится название компонента, за который поле отвечает и корректные размеры – минимальный 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.

После ввода некорректных значений поле TextBox изменит цвет, пример представлен на рисунке 3.3.

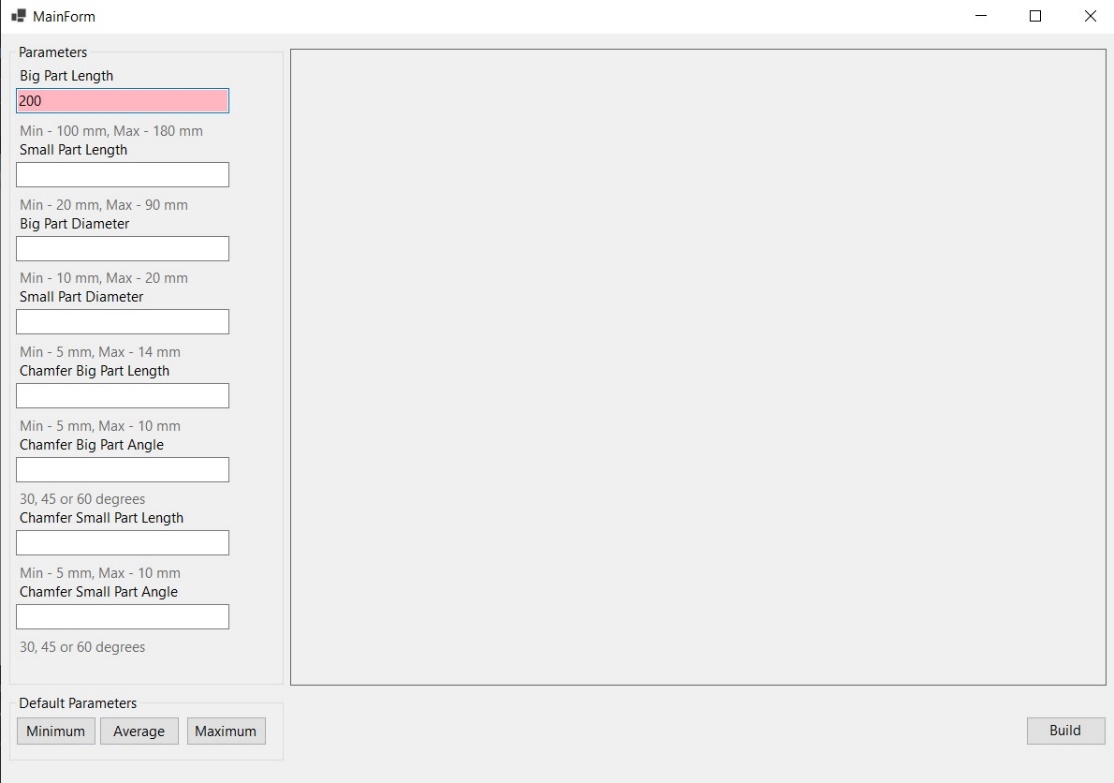


Рисунок 3.3 — Уведомление об ошибке

# Список литературы

1. САПР — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Система\_автоматизированного\_проектирования (дата обращения 13.10.2023).

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 20.10.2023).

3. API — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения 20.10.2023).

4. NanoCAD Модуль «Механика». [Электронный ресурс]. — Режим доступа: https://www.nanocad.ru/products/module\_mechanica/ (дата обращения 20.10.2023).

5. Рулевая тяга — Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%A8%D1%82%D0%BE%D0%BA (дата обращения 20.10.2023).

6. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 20.10.2023).