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 "ВАЛ-ШЕСТЕРНЯ" ДЛЯ САПР Компас-3D

Проект системы по дисциплине

«ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_А.С.Избышев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Руководитель:

к.т.н., доцент каф. КСУП

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

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Томск 2022

**Оглавление**

[1 Описание САПР 3](#_Toc119029927)

[1.1 Описание программы 3](#_Toc119029928)

[1.2 Анализ API 4](#_Toc119029929)

[2 Описание предмета проектирования 12](#_Toc119029930)

[3 Проект системы 14](#_Toc119029931)

[3.1 Описание технических и функциональных аспектов проекта 14](#_Toc119029932)

[3.2 Диаграмма классов 15](#_Toc119029933)

[3.3 Макет пользовательского интерфейса 19](#_Toc119029934)

[Список литературы 21](#_Toc119029935)

# 1 Описание САПР

# 1.1 Описание САПР «КОМПАС-3D»

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 [1]. 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.

# 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 [2].

Для КОМПАС-3D есть 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 указатель 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, 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 указатель 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 указатель 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**

Онлайн 3D – конфигуратор шестерней Gear Generator[3].

Gear Generator – это онлайн генератор для создания различных контуров шестерней, которые можно сконвертировать в 3D-модель в формате DXF, либо использовать как 2D-скетч в формате SVG для дальнейшего выдавливания в Inventor, Solidworks, Fusion 360 и других программах, поддерживающих эту функцию.

Генератор включает в себя всевозможные настраиваемые параметры: центр шестерни, диаметр, количество зубьев, профиль зуба и их направление. Также можно создавать внутренние шестерни.

Интерфейс программы показан на рисунке 1.1:



Рисунок 1.1 –Интерфейс Gear Generator

# 2 Описание предмета проектирования

Программа предназначена для автоматизации моделирования детали «Коническая вал-шестерня».

Вал-шестерня – комбинированный вид запчасти, состоящей непосредственно из самого вала и шестерни. Целевой задачей данной конструкции является передача крутящего усилия с одного вала на другой и поддержание шкивов, катков и прочих элементов в приводных механизмах, редукторах. Коническая вал-шестерня – это шестерня, в которой оси двух валов пересекаются, а зубчатая поверхность самой шестерни имеет коническую форму. Поверхность конических вал-шестерней представляет собой конус [4].

Изображение моделируемого объекта представлено на рисунке 2.1:



Рисунок 2.1 – Чертёж конической вал-шестерни

Изменяемые параметры для плагина:

**D1** – диаметр профиля шестерни (70-120 мм);

**W1** – ширина шестерни (70-200 мм);

**D2** – диаметр соединяющего цилиндра (85-110 мм);

**D3** – диаметр основного цилиндра (80-105 мм);

**D4** – диаметр вращательного цилиндра (10-55 мм);

**L1** – длина вращательного цилиндра (5-75 мм).

Ширина шестерни W1 и диаметр конического профиля D1 ограничены так, чтобы сохранять форму конуса и не допускать искажений модели.

Диаметр D2 и D3 не должны превышать выхода за профиль шестерни, а также зависимы между собой и второе значение должно быть больше первого на 5 см для правильного перехода между ними.

Диаметр D4 ограничен значением 10 см и предельно допустимым для перехода к следующему элементу значением 55 см. Длина L1 имеет границы в пределах нормы.

# 3 Проект системы

# 3.1 Описание технических и функциональных аспектов проекта

Плагин позволяет пользователю ввести вышеперечисленные значения через графический интерфейс. В программе предусмотрена проверка корректности введенных данных и сообщение пользователю о неправильно заполненных полях.

Если пользователем введены недопустимые значения параметров, то построение модели не начнётся.

При правильно введенных значениях результатом работы программы будет созданная по ним модель конической вал-шестерни.

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии) 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 [5].

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.

На рисунке 3.1 представлена диаграмма классов.

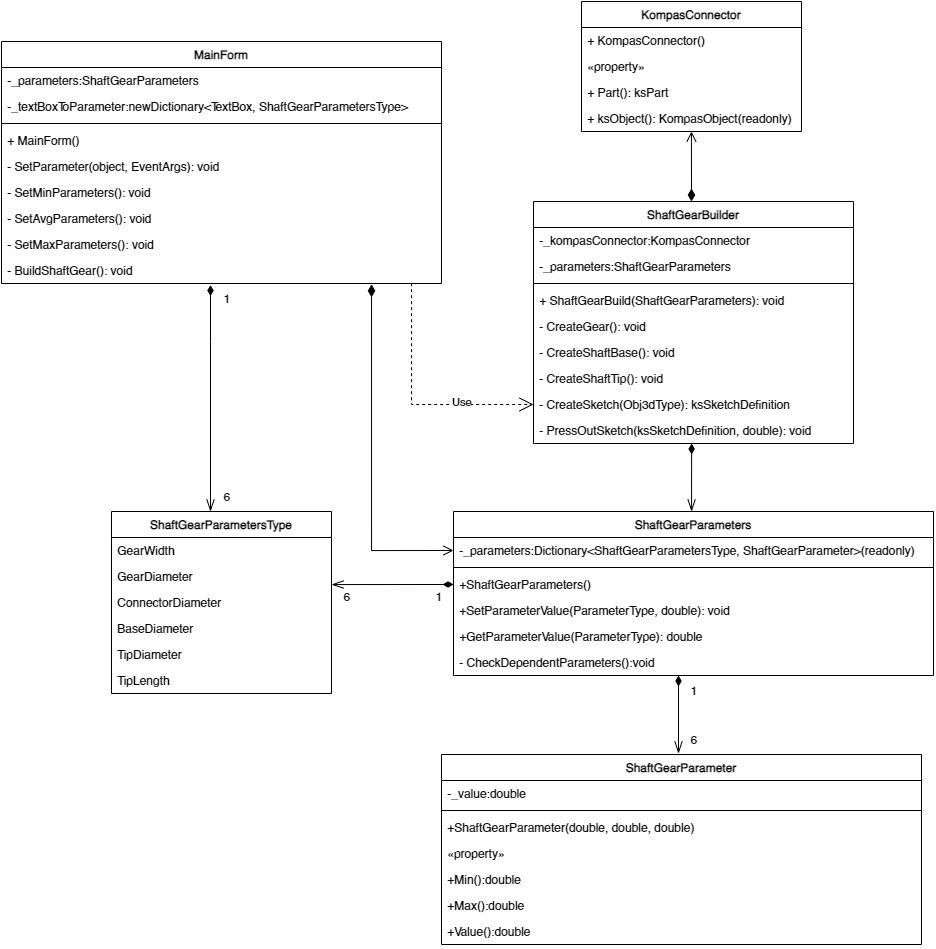


Рисунок 3.1 – Диаграмма классов UML

Далее в таблицах 3.1 – 3.4 представлено описание классов.

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

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Хранит в себе набор методов для построения вал-шестерни |
| \_textBoxToParameter |  | Хранит в себе набор TextBox и соответствующий ему тип параметра из перечисления «ShaftGearParametersType» |
| SetParameter(object, EventArgs) | void | Устанавливает значение параметра |
| SetMinParameters() | void | Устанавливает минимальное значение всех параметров |
| SetMaxParameters() | void | Устанавливает максимальное значение всех параметров |
| SetAvgParameters() | void | Устанавливает среднее значение всех параметров |
| Build() | void | Строит коническую вал-шестерню по заданным параметрам |

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

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_parameters |  | Хранит данные о каждом параметре модели из перечисления «ShaftGearParametersType» |
| ShaftGearParameters() |  | Конструктор для создания экземпляра класса |
| SetParameterValue(ParameterType, double) | void | Устанавливает значение определённого параметра |
| GetParameterValue(ParameterType) | double | Возвращает значение определённого параметра |

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

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| \_value |  | Хранит текущее значение |
| ShaftGearParameter(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 полей, методов, сущностей класса «SinkBuilder»

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| kompasConnector |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| \_parameters |  | Хранит данные о каждом параметре модели из перечисления «ShaftGearParameter» |
| ShaftGearBuild  (ShaftGearParameters) | void | Построение шестерни по заданным параметрам |
| CreateGear() | void | Построение основы шестерни |
| CreateShaftBase() | void | Построение основы вала |
| CreateShaftTip() | void | Построение вращательного цилиндра вала |

Окончание 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 конической вал-шестерни (рисунок 3.2). Построение модели осуществляется путем нажатия на кнопку «Build Model».

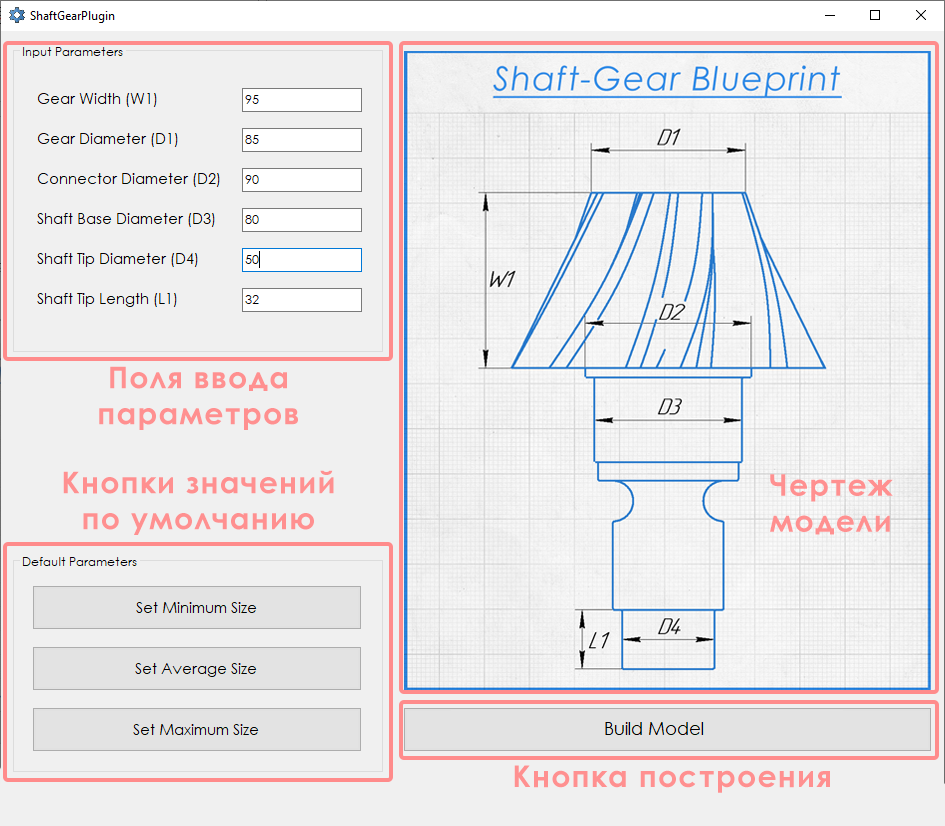


Рисунок 3.2 – Макет пользовательского интерфейса

С помощью данного окна пользователь может изменять параметры будущей 3D-модели конической вал-шестерни.

Рядом с полями ввода находятся название компонента, за который поле отвечает и его обозначение на чертеже.

На панели «Default Parameters» созданы 3 кнопки значений по умолчанию. При нажатии на кнопку «Set Minimum Size» будет создана 3D-модель с минимальными допустимыми размерами. При нажатии на кнопку «Set Average Size» будет создана 3D-модель со средними значениями размеров. При нажатии на кнопку «Set Maximum Size» будет создана 3D модель с максимальными корректными размерами.

Чертёж модели справа необходим для лучшего понимания расположения вводимых размеров – обозначения размеров соответствуют подписям полей ввода значений.

При вводе недопустимых значений рядом с полем ввода появляется предупреждение, при наведении на которое можно увидеть подробную информацию о причине его появления. На рисунке 3.3 продемонстрированы 2 предупреждения: первое из-за ввода слишком большого значения, выходящего за пределы допустимых; второе из-за нарушения зависимости параметров друг от друга.

Построение модели невозможно начать, пока все поля не будут корректно заполнены.

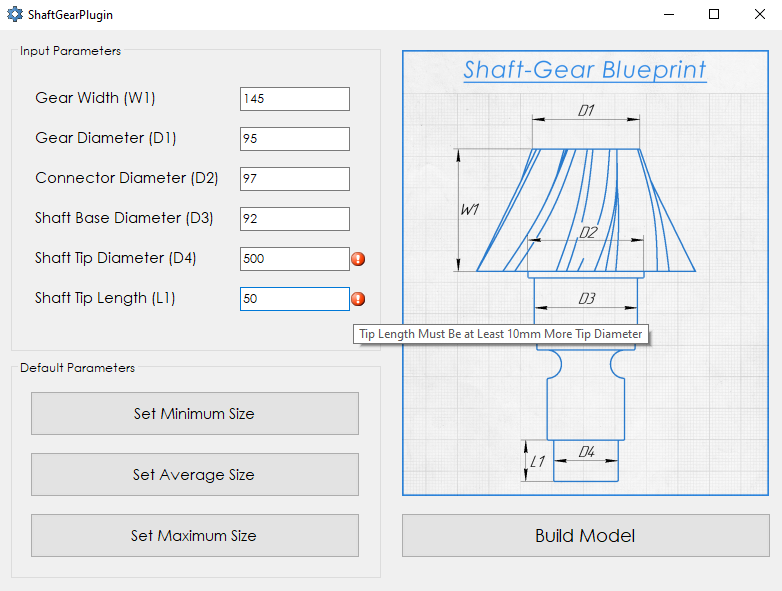


Рисунок 3.3 – Реализация предупреждений о недопустимых значениях

# Список литературы

1. Компас (САПР) — Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Компас_(САПР)> (дата обращения 02.11.2022).

2. API — Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 08.11.2022).

3. Генератор шестерни онлайн — Gear Generator [Электронный ресурс]. — Режим доступа: <https://geargenerator.com/beta> (дата обращения 18.11.2022).

4. Коническая вал-шестерня — Википедия. [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/Bevel_gear> (дата обращения 18.11.2022).

5. UML. [Электронный ресурс]. – Режим доступа: <http://www.uml.org/> (дата обращения 14.11.2022).