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 «Топорище» ДЛЯ САПР «КОМПАС-3D 2022»**

Проект системы по дисциплине

«ОСНОВЫ РАЗРАБОТКИ САПР»

Выполнил:

студент гр. 589-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.В. Федяев

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

Руководитель:

к.т.н., доцент каф. КСУП

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

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

Томск 2022

**Оглавление**

[1 Описание САПР 3](#_Toc116637014)

[1.1 Описание программы 3](#_Toc116637015)

[1.2 Описание API 4](#_Toc116637016)

[1.3 Обзор аналогов 10](#_Toc116637017)

[2 Описание предмета проектирования](#_Toc116637018) 11

[3 Проект программы](#_Toc116637019) 13

[3.1 Диаграмма классов](#_Toc116637021) 13

[3.2 Макет пользовательского интерфейса](#_Toc116637022) 16

[Список литературы](#_Toc116637023) 18

**1 Описание САПР**

**1.1 Описание программы**

КОМПАС-3D – это российская импортонезависимая система трехмерного проектирования, ставшая стандартом для тысяч предприятий и сотен тысяч профессиональных пользователей

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].

**1.2 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 созданы две различные версии API - версии 5 и версии 7. К ним разработчик прилагает справочную систему по всем включенным в эту API интерфейса

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 - 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 ссылки 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 ссылку 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 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 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 | Ссылка 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 | Ссылка на интерфейс оси ksEntity |

**1.3 Обзор аналогов**

PTC Creo — это масштабируемый, функционально совместимый пакет программного обеспечения для конструирования изделий. Он позволяет группам конструкторов создавать, анализировать, просматривать и максимально использовать проекты изделий при дальнейшем конструировании, используя 2 - и 3-мерное моделирование CAD, параметрическое и прямое моделирование. [2] Классический интерфейс PTC Creo представлен на рисунке 1.1.

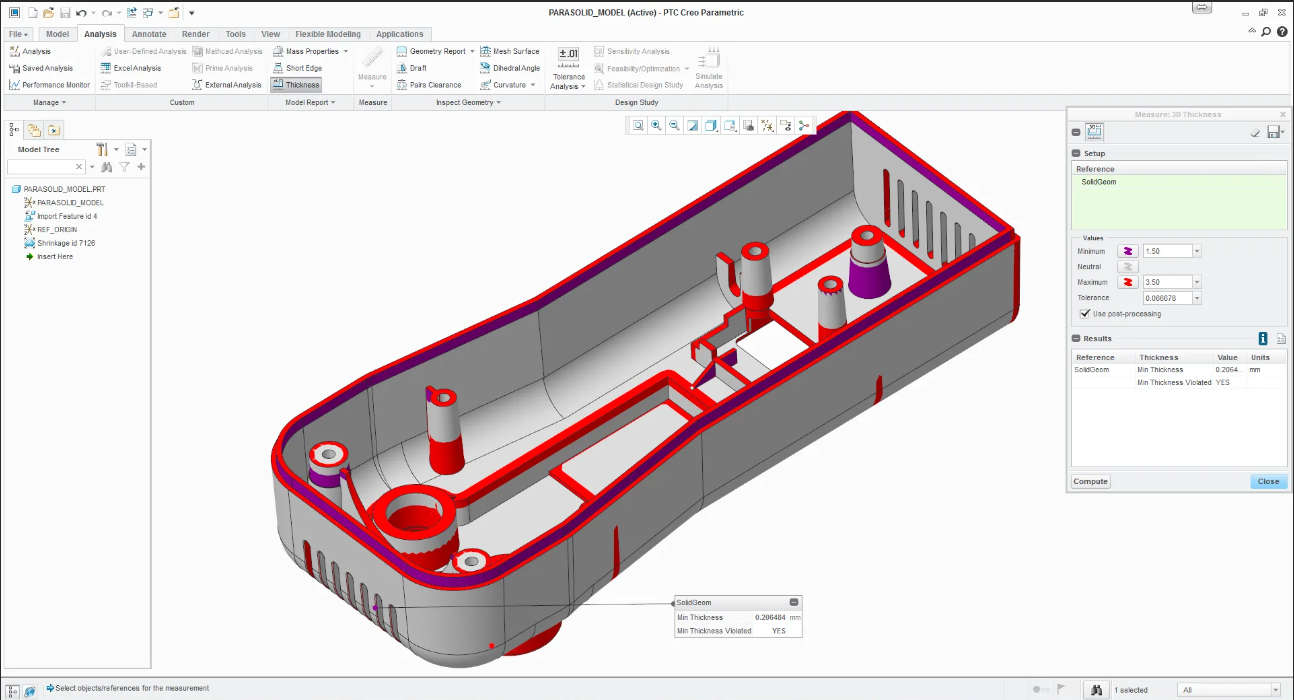


Рисунок 1.1 – Классический интерфейс PTC Creo

**2 ОПИСАНИЕ ПРОЕКТА ПРОЕКТИРОВАНИЯ**

Топорище – часть топора, благодаря которому можно рубить дрова. Существуют различные виды топорища. Изображение моделируемого объекта представлено на рисунке 2.1.

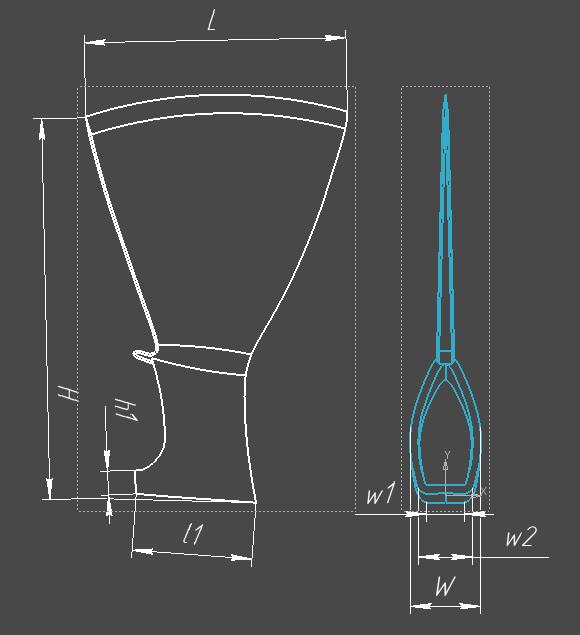


Рисунок 2.1 – Чертеж топорища

Измеряемые параметры для плагина:

* L – длина острой части топора (минимум – 135мм, максимум – 165мм);
* l1 – длина части топора l1 (Зависит от длины топора L: L/2.5 = l1 минимум – 48мм, максимум – 68мм);
* H – высота топора (минимум – 170мм, максимум – 215мм);
* h1 – высота части топора (не меньше 1/12 H, не больше 1/15 H, минимум – 12мм, максимум – 15мм);
* W – ширина топора (минимум – 35мм, максимум – 41мм);
* w1 – ширина кусочка топора (не больше ширины топора W и не меньше ширины топора w2, W > w1 > w2, минимум – 22мм, максимум – 28мм);
* w2 – ширина второго кусочка топора (не больше ширины топора w1, w1 **>** w2, минимум – 19мм, максимум – 25мм).

Модель топора представлена на рисунке 2.2.

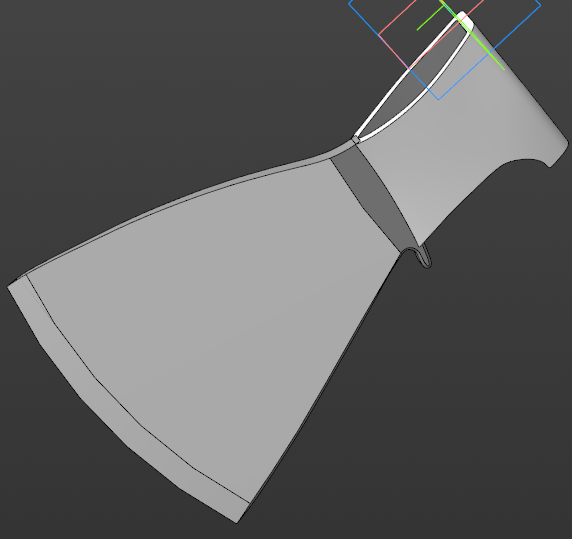


Рисунок 2.2- 3D модель топорища

**3 ПРОЕКТ СИСТЕМЫ**

**3.1 Диаграмма классов**

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур [2]. Диаграмма классов UML представлена на рисунке 3.1

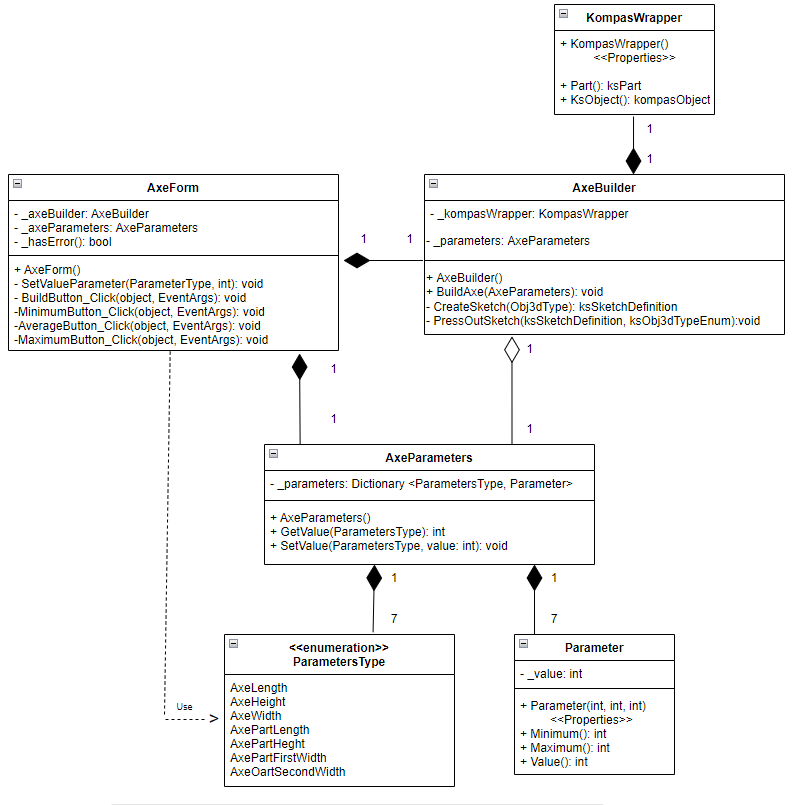


Рисунок 3.1 – Диаграмма классов UML

Таблица 3.1 – Описание полей, методов, сущностей класса “AxeForm”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_axeBuilder: AxeBuilder |  | Хранит в себе набор методов для построения топорища |
| - \_axeParameters:  AxeParameters |  | Устанавливает значение параметра |
| -\_hasError() | bool | Выдает ошибку |
| -SetValueParameter  (ParametersType, int) | Void | Устанавливает значение параметра |
| -BuildButton\_Click() | Void | Построение топорища по заданным параметрам |
| -MinimumButton\_Click() | Void | Устанавливает минимальное значение всех параметров |
| -AverageButton\_Click() | Void | Устанавливает среднее значение всех параметров |
| -MaximumButton\_Click() | Void | Устанавливает максимальное значение всех параметров |

Таблица 3.2 - Описание полей, методов, сущностей класса “AxeParameters”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_parameters: Dictionary <ParametersType, Parameter> |  | Хранит данные о каждом параметре модели из перечисления “AxeParameter” |
| + AxeParameters() |  | Конструктор для создания экземпляра класса |
| + GetValue(ParametersType) | Int | Устанавливает значение определённого параметра |
| + SetValue(ParametersType, value: int) | void | Возвращает значение определённого параметра |

Таблица 3.3 - Описание полей, методов, сущностей класса “Parameter”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_value: int |  | Поле, хранящее текущее значение |
| + Parameter(int, int, int) |  | Конструктор для создания экземпляра класса |
| + Minimum(): | Int | Возвращает минимально допустимое значения параметра |
| + Maximum(): | Int | Возвращает максимально допустимое значения параметра |
| + Value(): | Int | Возвращает и задаёт значение параметра |

Таблица 3.4 - Описание полей, методов, сущностей класса “AxeBuilder”

|  |  |  |
| --- | --- | --- |
| Название метода/поля | Возвращаемый тип | Описание |
| - \_kompasWrapper: KompasWrapper |  | Хранит в себе методы необходимые для связи с КОМПАС 3D |
| - \_parameters: AxeParameters |  | Хранит данные о каждом параметре модели из перечисления “AxeParameter” |
| + BuilderAxe(AxeParameters) | Void | Построение топора по заданным параметрам |
| - CreateSketch(obj3dType) | ksSketchDefinition | Возвращает интерфейс параметров эскиза |
| -PressOutSketch (ksSketchDefinition, int) | Void | Выдавливает эскиз |

**3.2 Макеты пользовательского интерфейса**

Макет пользовательского интерфейса представляет собой форму для ввода параметров топорища. Построение модели осуществляется путем нажатия на кнопку «Build». Если какое-то из полей не будет проходить проверку, то кнопка «Build» будет неактивна. Макет пользовательского интерфейса представлен на рисунке 3.2

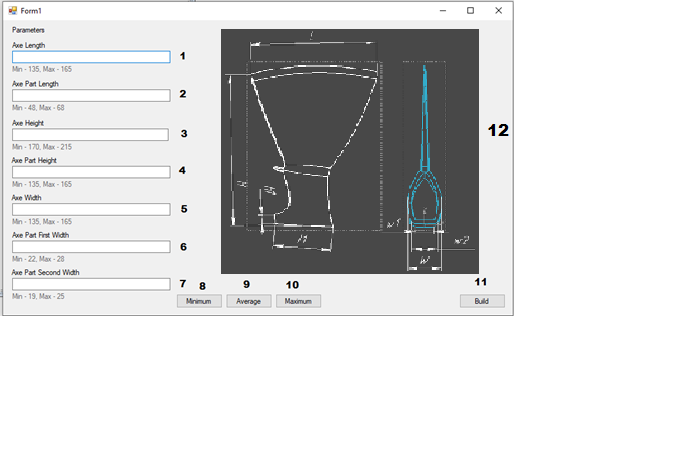


Рисунок 3.2 – Макет пользовательского интерфейса

1. Поле ввода параметра длины топора;
2. Поле ввода параметра части длины топора;
3. Поле ввода параметра высота топора;
4. Поле ввода параметра части высота топора;
5. Поле ввода параметра ширины топора;
6. Поле ввода параметра первой части ширины топора;
7. Поле ввода параметра второй части ширины топора;
8. Кнопка для установления минимального значения топора;
9. Кнопка для установления среднего значения топора;
10. Кнопка для установления максимального значения топора;
11. Кнопка для построения 3D модели топора;
12. Область для вывода чертежа.

После ввода некорректных значений, окно для ввода данных станет красным. Пример представлен на рисунке 3.3

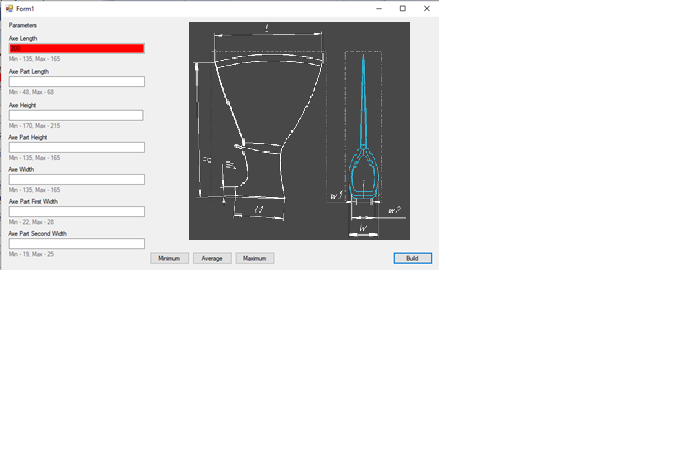


Рисунок 3.3 – Макет пользовательского интерфейса с некорректными данными

**Список литературы**

1. КОМПАС 3D [Электронный ресурс]. — Режим доступа: <https://kompas.ru/kompas-3d/about/> (дата обращения 14.10.2022).
2. PTC Creo [Электронный ресурс]. — Режим доступа: <https://junior3d.ru/article/creo.html> (дата обращения 14.10.2022).
3. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 14.10.2022).