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)

БИБЛИОТЕКА ЗВЁЗДНОГО ИСТРЕБИТЕЛЯ T-65 «X-WING» ДЛЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

КОМПАС-3D

Проект системы по дисциплине

«Основы разработки САПР»

Студент гр. 589-1

\_\_\_\_\_\_\_\_\_ Ю. Н. Кобзарь

\_\_\_\_\_\_\_\_\_

Руководитель

преподаватель каф., к.т.н.

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

(оценка) \_\_\_\_\_\_\_\_\_

Томск 2022 г

**Оглавление**

[1 ОПИСАНИЕ САПР 3](#_Toc115991120)

[1.1 Описание программы 3](#_Toc115991121)

[1.2 Описание API 4](#_Toc115991122)

[1.3 Обзор аналогов 16](#_Toc115991123)

[1.3.1 Плагин «Airplane Design Workbench» для FreeCAD 16](#_Toc115991124)

[1.3.2 Плагин «Rocket Workbench» для FreeCAD 17](#_Toc115991125)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 19](#_Toc115991126)

[3 ПРОЕКТ ПРОГРАММЫ 21](#_Toc115991127)

[3.1 Диаграмма классов 21](#_Toc115991128)

[3.2 Макеты пользовательского интерфейса 27](#_Toc115991129)

[Список использованных источников 29](#_Toc115991130)

# 1 ОПИСАНИЕ САПР

# 1.1 Описание программы

«Компас» – это семейство систем автоматизированного проектирования, универсальная система автоматизированного проектирования, позволяющая в оперативном режиме выпускать чертежи изделий, схемы, спецификации, таблицы, инструкции, расчётно-пояснительные записки, технические условия, текстовые и прочие документы [1].

Разрабатывается российской компанией «Аскон». Название линейки является акронимом от фразы «комплекс автоматизированных систем». В торговых марках используется написание заглавными буквами: «КОМПАС» [2].

Функциональные особенности и возможности приложения:

* импорт готовых проектов из сети интернет;
* русифицированный интерфейс;
* удобная рабочая область и широкий набор профессиональных инструментов;
* формирование документации;
* можно создавать модели любого уровня сложности;
* подходит для создания моделей с дальнейшей печатью на 3D-принтере;
* интеграция с CAD, CAM, CAE пакетами;
* автоматическое формирование длинных однообразных моделей;
* поддержка проектирования электрических схем, валов, пружин и т.д.

# 1.2 Описание API

Сегодня встречаются задачи, решение которых не реализованы в CAD-системах. Чаще всего это очень узкоспециализированные задачи, которые встречаются на каком-то конкретном предприятии или подотрасли. Для решения подобных задач вы можете использовать КОМПАС-3D как платформу и на базе него создать свое приложение, которое позволит вам автоматизировать решение таких задач. Для создания таких приложений в КОМПАС-3D есть открытый API [3].

API – описание способов взаимодействия одной компьютерной программы с другими. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений. Проще говоря, это набор компонентов, с помощью которых компьютерная программа (бот или же сайт) может использовать другую программу [4].

Для реализации плагина будет достаточно методов и свойств интерфейсов API 5 [5]. В таблице 1.1 представлены интерфейсы, которые будут использованы при разработке библиотеки. В таблицах 1.2 – 1.30 представлены методы и свойства описанных выше интерфейсов, которые будут использоваться при разработке плагина, и представлено описание входных параметров этих методов.

Таблица 1.1 – Интерфейсы, используемые при разработке

|  |  |
| --- | --- |
| Название интерфейса | Описание интерфейса |
| KompasObject | Интерфейс API КОМПАС |
| ksPart | Интерфейс детали или подсборки в составе сборки |
| ksDocument3D | Интерфейс документа-модели |

Продолжение таблицы 1.1

|  |  |
| --- | --- |
| ksEntity | Интерфейс элемента модели (оси, плоскости, формообразующего элемента) |
| ksSketchDefinition | Интерфейс параметров эскиза |
| ksDocument2D | Интерфейс графического документа системы КОМПАС |
| ksBaseExtrusionDefinition | Интерфейс параметров основания - элемента выдавливания |
| ksEntityCollection | Интерфейс динамического массива объектов заданного типа, выбранных в документе |
| ksCutExtrusionDefinition | Интерфейс параметров вырезанного элемента выдавливания |
| ksChamferDefenition | Интерфейс параметров фаски |
| ksFilletDefinition | Интерфейс параметров элемента "скругление" |
| ksBossRotatedDefinition | Интерфейс приклеенного элемента вращения |

Таблица 1.2 – Используемые методы интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| ActiveDocument3D() | 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 ksPart

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| GetDefaultEntity (short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |

Продолжение таблицы 1.3

|  |  |  |
| --- | --- | --- |
| EntityCollection (short objType) | ksEntity | Получить указатель на интерфейс динамического массива объектов заданного типа, выбранных в документе |
| NewEntity  (short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |
| Update() | bool | Изменить свойства компонента (используя ранее установленные свойства) |
| SetAdvancedColor (long color, double ambient, double diffuse, double specularity, double shininess, double transparency, double emission) | bool | Установить параметры цвета и визуальных свойств объекта |

Таблица 1.4 – Описание входных параметров, используемых методов интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Метод | Входной параметр | Описание параметра |
| GetDefaultEntity (short objType) | objType | Тип объекта |
| EntityCollection (short objType) | objType | Тип объекта |
| NewEntity  (short objType) | objType | Тип объекта |
| SetAdvancedColor (long color, double ambient, double diffuse, double specularity, double shininess, double transparency, double emission) | color | Цвет |
| ambient | Общий свет |
| diffuse | Диффузия |
| specularity | Зеркальность |
| shininess | Блеск |
| transparency | Прозрачность |
| emission | Излучение |

Таблица 1.5 – Используемые типы объектов в методах интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| **Метод** | **Тип объекта** | **Название объекта** |
| GetDefaultEntity  (short objType) | o3d\_planeXOY | Плоскость XOY |
| EntityCollection (short objType) | o3d\_face | Поверхность |
| o3d\_edge | Ребро |
| NewEntity  (short objType) | o3d\_sketch | Эскиз |
| o3d\_baseExtrusion | Базовая операция выдавливания |
| o3d\_cutExtrusion | Вырезать выдавливанием |
| o3d\_ chamfer | Операция «фаска» |
| o3d\_fillet | Операция «скругление» |
| o3d\_edge | Ребро |
| o3d\_bossRotated | Приклеивание вращением |

Таблица 1.6 – Используемые свойства интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| name | string | Имя детали или подсборки в составе сборки |

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 или редактируемый или указанный компонент (например, сборка, в составе которой находится редактируемая деталь) |

Таблица 1.9 – 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 1.10 – Используемые методы интерфейса ksDocument2D

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемое значение | Описание |
| ksCircle (double xc, double yc, double rad, long style) | Указатель на окружность – в случае удачного завершения, 0 – в случае неудачи | Создать окружность |
| ksLineSeg (double x1, double y1, double x2, double y2, long style) | Указатель на отрезок – в случае удачного завершения, 0 – в случае неудачи | Создать отрезок |
| ksArcByPoint (double xc, double yc, double rad, double x1, double y1, double x2, double y2, short direction, long style) | Указатель на дугу – в случае удачного завершения, 0 – в случае неудачи | Создать дугу по центру и конечным точкам |

Таблица 1.11 – 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 |
| ksLineSeg (double x1, double y1, double x2, double y2, long style) | x1, y1 | Координаты первой точки отрезка |
| x2, y2 | Координаты второй точки отрезка |
| style | Стиль линии |

Продолжение таблицы 1.11

|  |  |  |
| --- | --- | --- |
| ksArcByPoint (double xc, double yc, double rad, double x1, double y1, double x2, double y2, short direction, long style) | xc, yc | Координаты центра дуги |
| rad | Радиус дуги |
| x1, y1 | Координаты начальной точки дуги |
| x2, y2 | Координаты конечной точки дуги |
| direction | Направление отрисовки дуги: «1» против часовой стрелки, «-1» по часовой стрелке |
| style | Стиль линии |

Таблица 1.12 – Используемые методы интерфейса ksSketchDefinition

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

Таблица 1.13 – 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 |
| SetThinParam (bool thin, short thinType, double normalThickness, double reverseThickness) | bool | Изменить параметры тонкой стенки |
| SetSketch (LPDISPATCH sketch) | bool | Задать указатель на интерфейс эскиза элемента |

Таблица 1.14 – 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 |
| SetThinParam (bool thin, short thinType, double normalThickness, double reverseThickness) | thin | Признак тонкостенной операции |
| thinType | Направление построения тонкой стенки |
| normalThickness | Толщина стенки в прямом направлении, |
| reverseThickness | Толщина стенки в обратном направлении |
| SetSketch (LPDISPATCH sketch) | sketch | Указатель на интерфейс эскиза ksEntity |

Таблица 1.15 – Используемые свойства интерфейсаksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| directionType | short | Направление выдавливания  dtNormal - прямое направление  dtReverse – обратное направление |

Таблица 1.16 – Используемые типы объектов в методах интерфейса ksBaseExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| **Метод** | **Тип объекта** | **Название объекта** |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | etBlind | Строго на глубину |

Таблица 1.17 – Используемые методы интерфейса ksEntityCollection

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SelectByPoint (double x, double y, double z) | bool | Исключить из массива все объекты, не содержащие точку с заданными координатами |
| Add (LPDISPATCH entity) | bool | Добавить объект в массив |
| First() | ksEntity | Получить первый объект в массиве |
| Last() | ksEntity | Получить указатель на интерфейс последнего объекта в массиве |

Таблица 1.18 – Описание входных параметров, используемых методов интерфейса ksEntityCollection

|  |  |  |
| --- | --- | --- |
| **Метод** | **Входной параметр** | **Описание входного параметра** |
| SelectByPoint (double x, double y, double z) | x, y, z | координаты точки |
| Add (LPDISPATCH entity) | entity | Указатель на интерфейс эскиза ksEntity |

Таблица 1.19 – Используемые методы интерфейса ksCutExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | bool | Установить параметры выдавливания в одном направлении |

Продолжение таблицы 1.19

|  |  |  |
| --- | --- | --- |
| SetSketch (LPDISPATCH sketch) | bool | Задать указатель на интерфейс эскиза элемента |

Таблица 1.20 – Описание входных параметров, используемых методов интерфейса ksCutExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| **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 1.21 – Используемые свойства интерфейсаksCutExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| directionType | short | Направление выдавливания  dtNormal - прямое направление  dtReverse – обратное направление |

Таблица 1.22 – Используемые типы объектов в методах интерфейса ksCutExtrusionDefinition

|  |  |  |
| --- | --- | --- |
| **Метод** | **Тип объекта** | **Название объекта** |
| SetSideParam (bool forward, short type, double depth, double draftValue, bool draftOutward) | etBlind | Строго на глубину |

Таблица 1.23 – Используемые методы интерфейсаksChamferDefenition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Array() | Указатель на интерфейс ksEntityCollection | Получить указатель на интерфейс массива ребер и граней, на которых строится фаска |
| SetChamferParam (BOOL transferTRUE, double distance1, double distance2) | bool | Изменить параметры фаски |

Таблица 1.24 – Описание входных параметров, используемых методов интерфейса ksChamferDefenition

|  |  |  |
| --- | --- | --- |
| **Метод** | **Входной параметр** | **Описание входного параметра** |
| SetChamferParam (BOOL transferTRUE, double distance1, double distance2); | transfer | Признак направления фаски, |
| distance1 | Размер первого катета фаски, |
| distance2 | Размер второго катета фаски, |

Таблица 1.25 – Используемые свойства интерфейсаksChamferDefenition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| tangent | bool | Признак продолжения фаски по касательным ребрам |

Таблица 1.26 – Используемые методы интерфейсаksFilletDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| Array() | Указатель на интерфейс массива объектов ksEntityCollection | Получить указатель на интерфейс массива скругляемых объектов (граней и ребер) |

Таблица 1.27 – Используемые свойства интерфейсаksFilletDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| tangent | bool | Признак продолжения скругления по касательным ребрам |
| radius | double | Радиус скругления |

Таблица 1.28 – Используемые методы интерфейсаksBossRotatedDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| SetSideParam (short forward, double angle); | bool | Изменить параметры вращения в одном направлении |
| SetSketch (LPDISPATCH sketch) | sketch | Изменить указатель на интерфейс эскиза элемента |

Таблица 1.29 – Описание входных параметров, используемых методов интерфейсаksBossRotatedDefinition

|  |  |  |
| --- | --- | --- |
| **Метод** | **Входной параметр** | **Описание входного параметра** |
| SetSideParam (short forward, double angle); | forward | Направление вращения:  TRUE - прямое,  FALSE - обратное. |
| angle | Угол вращения |
| SetSketch (LPDISPATCH sketch) | sketch | Указатель на интерфейс эскиза ksEntity |

Таблица 1.30 – Используемые свойства интерфейса ksBossRotatedDefinition

|  |  |  |
| --- | --- | --- |
| Название | Возвращаемый тип | Описание |
| directionType | short | Направление выдавливания  dtNormal - прямое направление |

# 1.3 Обзор аналогов

# 1.3.1 Плагин «Airplane Design Workbench» для FreeCAD

Данный экспериментальный плагин предназначен для проектирования крыльев и объектов самолетов [6][7]. На рисунке 1.1 представлен ввод параметров проектируемого крыла, на рисунке 1.2 представлено крыло, построенное с помощью данного плагина.

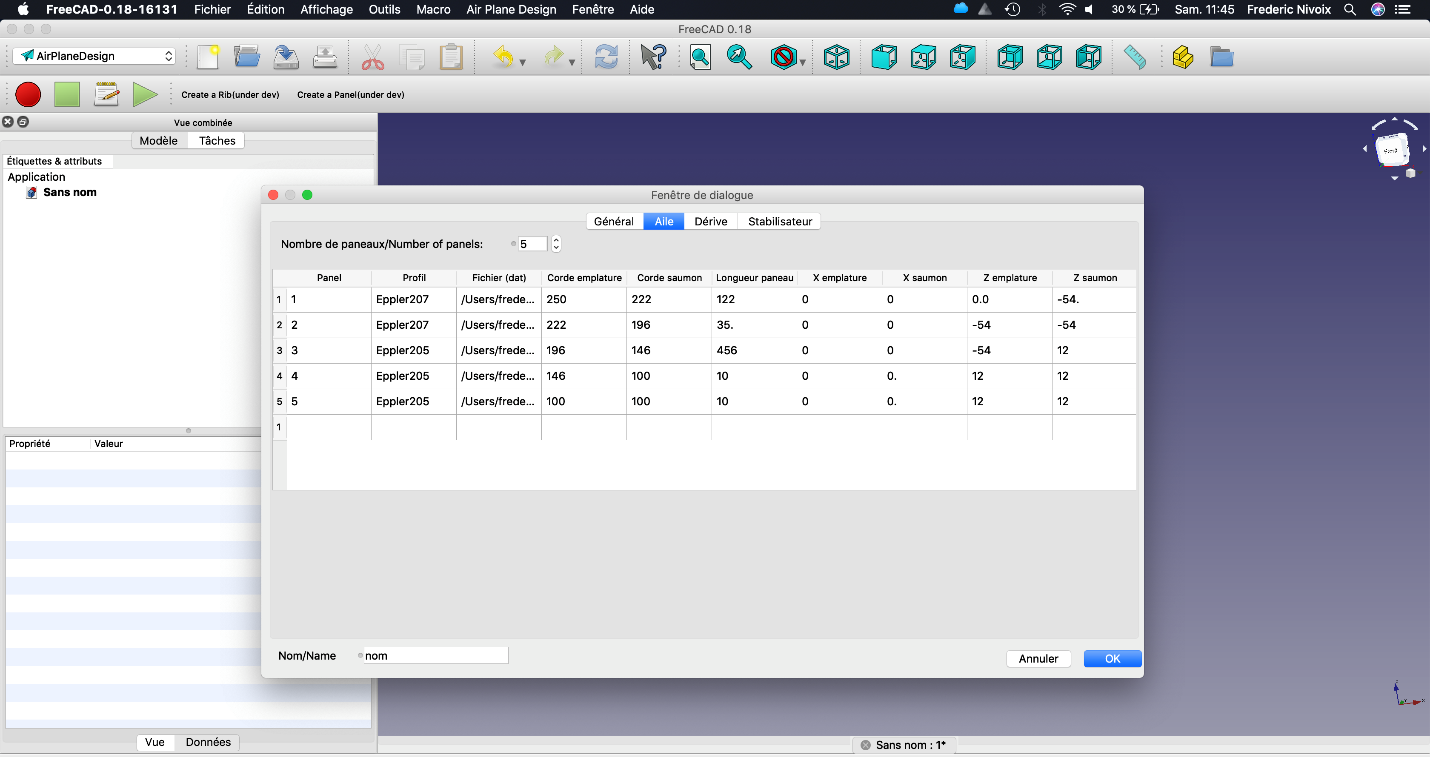


Рисунок 1.1 – Ввод параметров крыла через пользовательский интерфейс

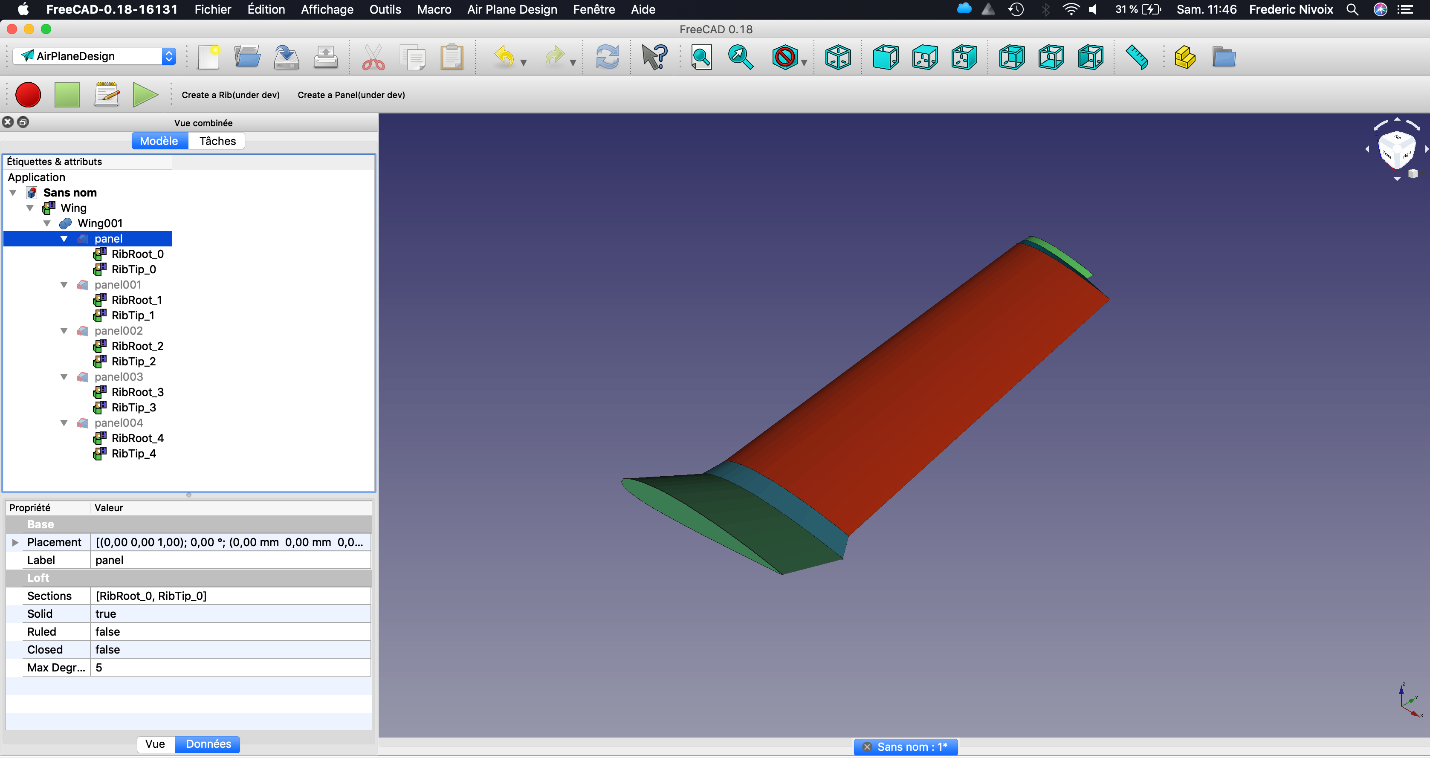


Рисунок 1.2 – Построенное крыло с помощью плагина во FreeCAD

# 1.3.2 Плагин «Rocket Workbench» для FreeCAD

Rocket Workbench – это набор инструментов для проектирования ракет и их компонентов. Целевая аудитория – производители моделей и любительских ракет всех типов и размеров. В настоящее время инструмент используется в основном для создания отдельных компонентов, а не полных ракет. Детали создаются путем указания параметров в диалоге задачи [6][8].

Преимущество использования Rocket Workbench заключается в том, что детали можно создавать с гораздо меньшим количеством операций [8]:

* носовой обтекатель – создание носового обтекателя;
* переход – создание перехода;
* трубка корпуса – создание полую трубку корпуса;
* центрирующее кольцо – создание центрирующего кольца;
* переборка – создание перегородки;
* плавник – создание плавника;
* руководство по запуску – создание руководства по запуску.

На рисунке 1.3 показаны примеры объектов, которые можно создавать с помощью данного плагина.

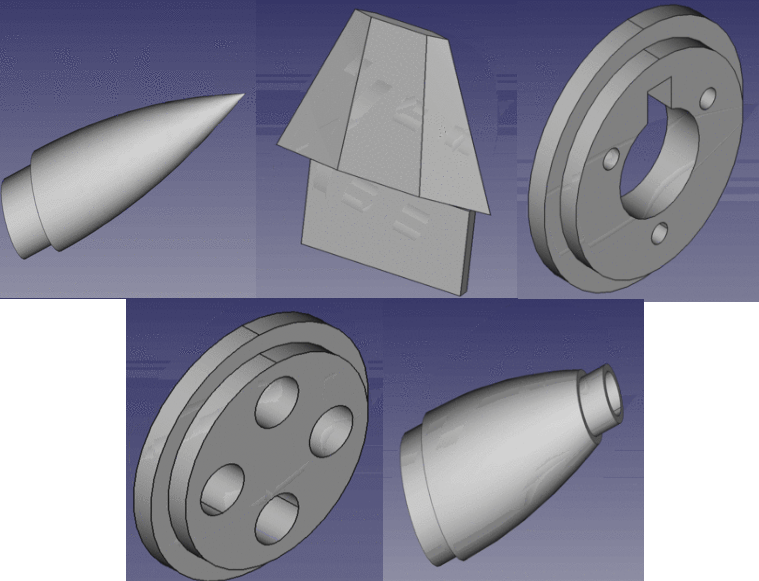


Рисунок 1.3 – Примеры созданных объектов с помощью плагина

# 2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Выбранным предметом для проектирования является звёздный истребитель T-65 «X-Wing» из вселенной Star Wars. Оригинал разрабатываемого объекта представлен на рисунке 2.1.



Рисунок 2.1 – Звёздный истребитель T-65 «X-Wing»

Трёхмерная модель звёздного истребителя T-65 «X-Wing», разработанная в системе автоматизированного проектирования Компас-3D представлена на рисунках 2.2 и 2.3.

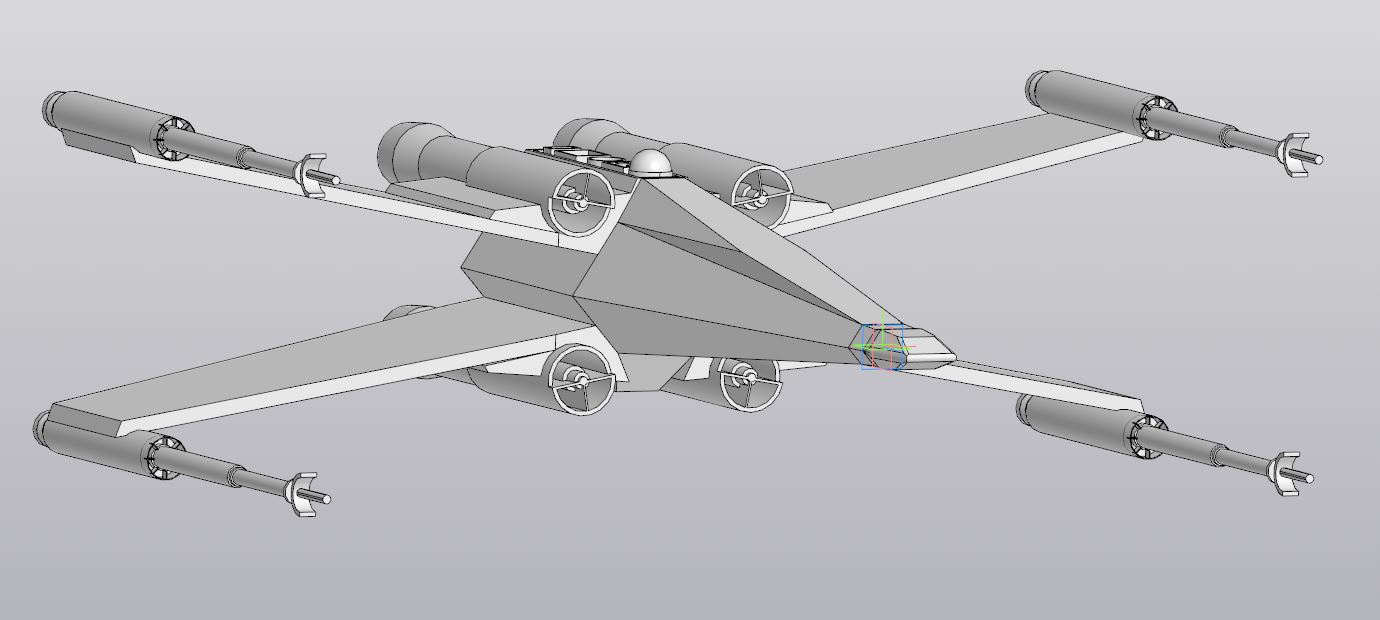


Рисунок 2.2 – Звёздный истребитель T-65 «X-Wing» в виде спереди

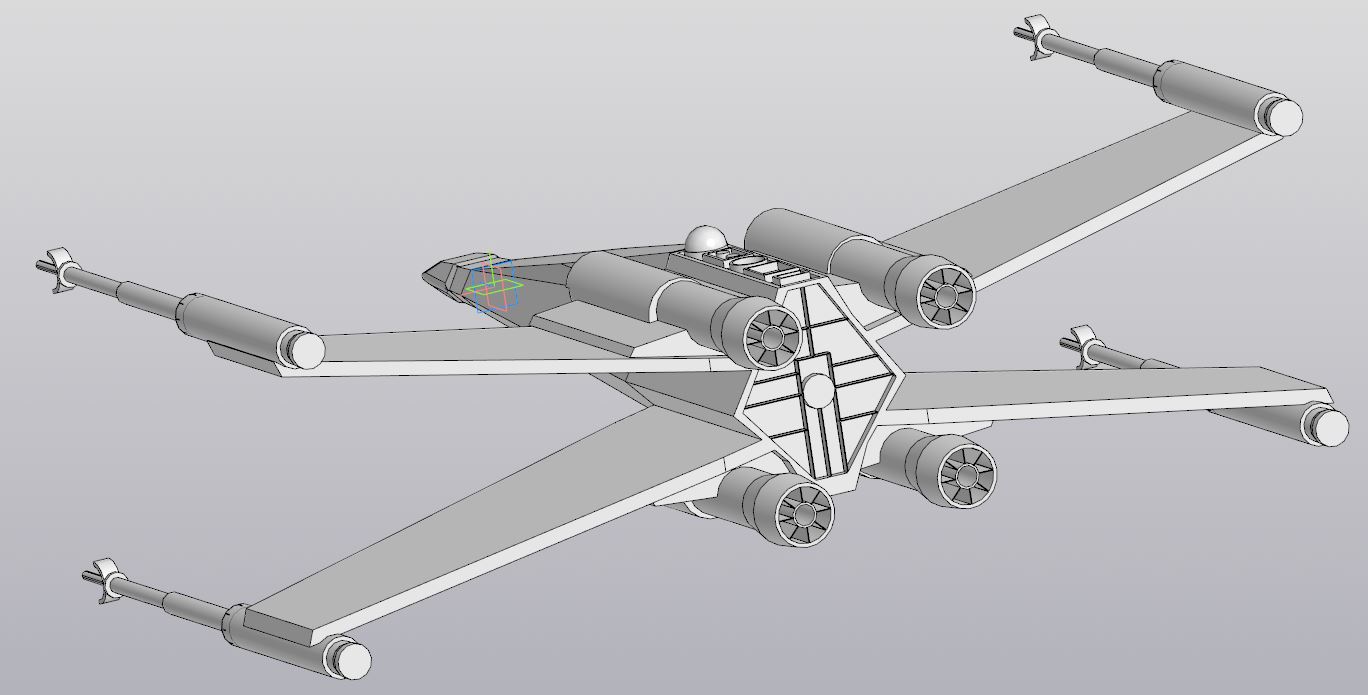


Рисунок 2.3 – Звёздный истребитель T-65 «X-Wing» в виде сзади

На рисунке 2.4 представлен 2D-чертеж трёхмерной модели звёздного истребителя T-65 «X-Wing».

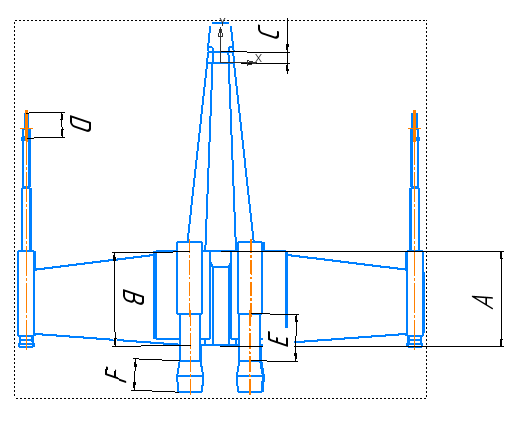


Рисунок 2.4 – Чертёж звёздного истребителя T-65 «X-Wing»

# 3 ПРОЕКТ ПРОГРАММЫ

# 3.1 Диаграмма классов

**UML (Unified Modeling Language)** – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем. Словарь UML включает три вида строительных блоков: диаграммы, сущности, связи [9].

**Sparx Systems Enterprise Architect** – это инструмент визуального моделирования и проектирования, основанный на OMG UML. Платформа поддерживает: проектирование и построение программных комплексов; моделирование бизнес-процессов; и моделирование отраслевых доменов. Он используется предприятиями и организациями не только для моделирования архитектуры своих систем, но и для обработки реализации этих моделей на протяжении всего жизненного цикла разработки приложений [10].

С помощью инструмента Enterprise Architect были созданы диаграммы классов в нотации UML. Из-за громоздкости диаграмма всего проекта поделена на несколько частей. Класс KompasWrapper присутствует на двух частях диаграммы, рисунки 3.1-3.2; класс XWingBuilder присутствует на всех частях диаграммы; на четырёх частях, рисунки 3.2-3.5, присутствуют четыре вспомогательных класса фигур: Point2D, Point3D, Circle, Arc. Эти четыре класса и класс XWingBuilder связаны с классами констант, которые необходимы для построения детали, связи всех этих классов можно наблюдать на рисунках 3.2-3.5. Все диаграммы представлены на рисунках 3.1-3.5.

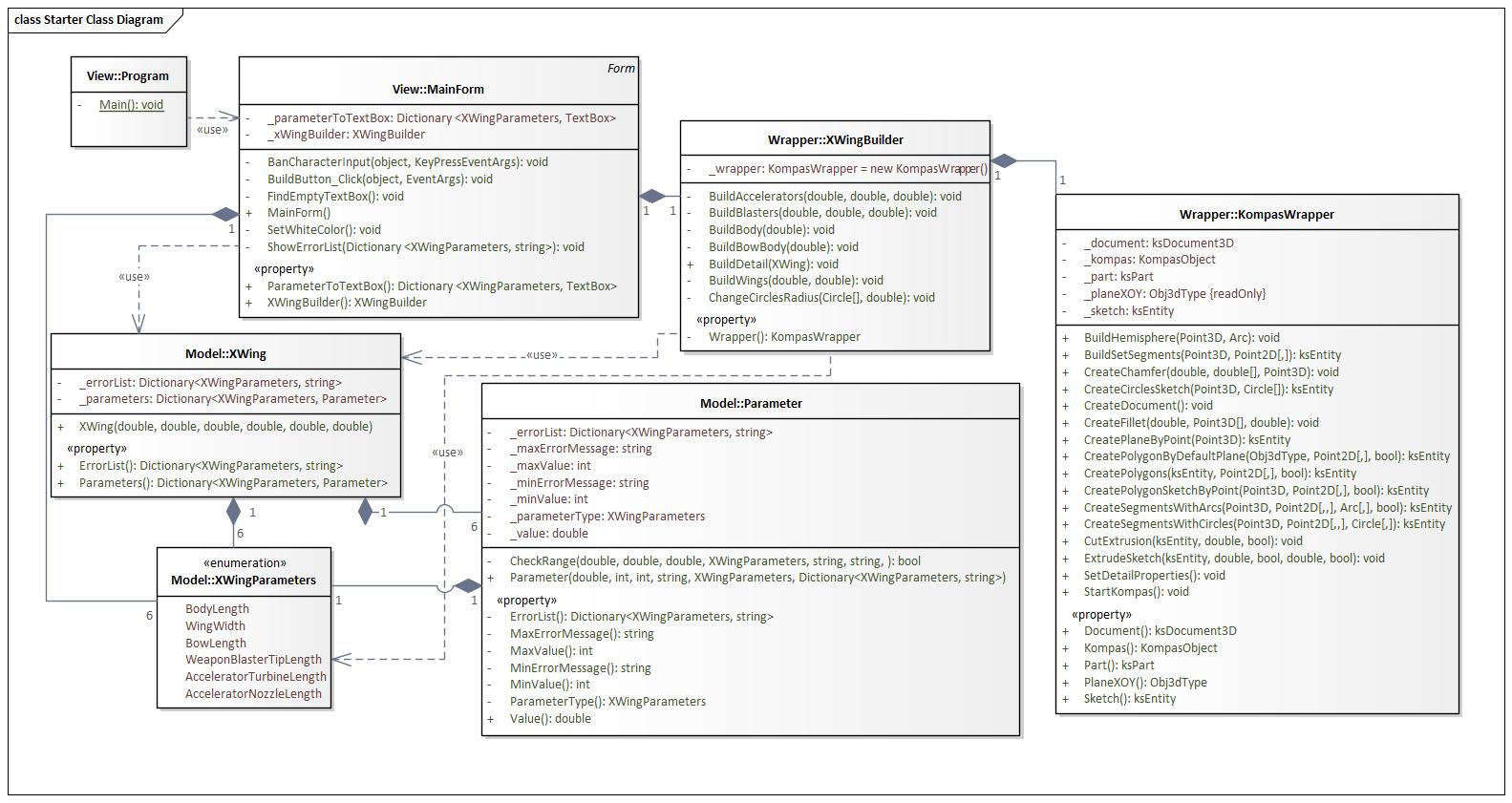


Рисунок 3.1 – Связь основной формы с классами параметров, списка ошибок, построителя, обёртки

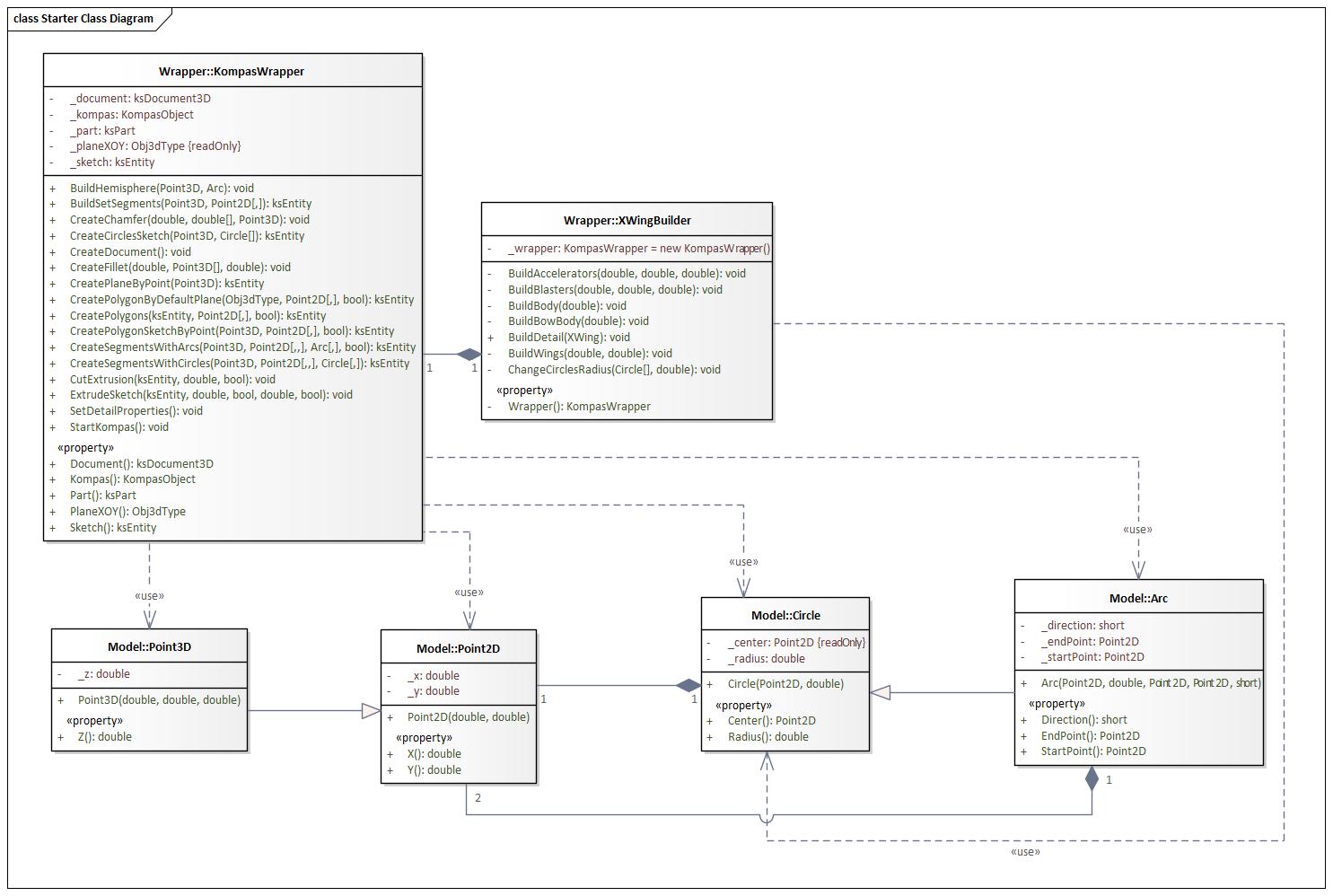
****

Рисунок 3.2 – Связь обёртки с построителем и классами геометрических фигур

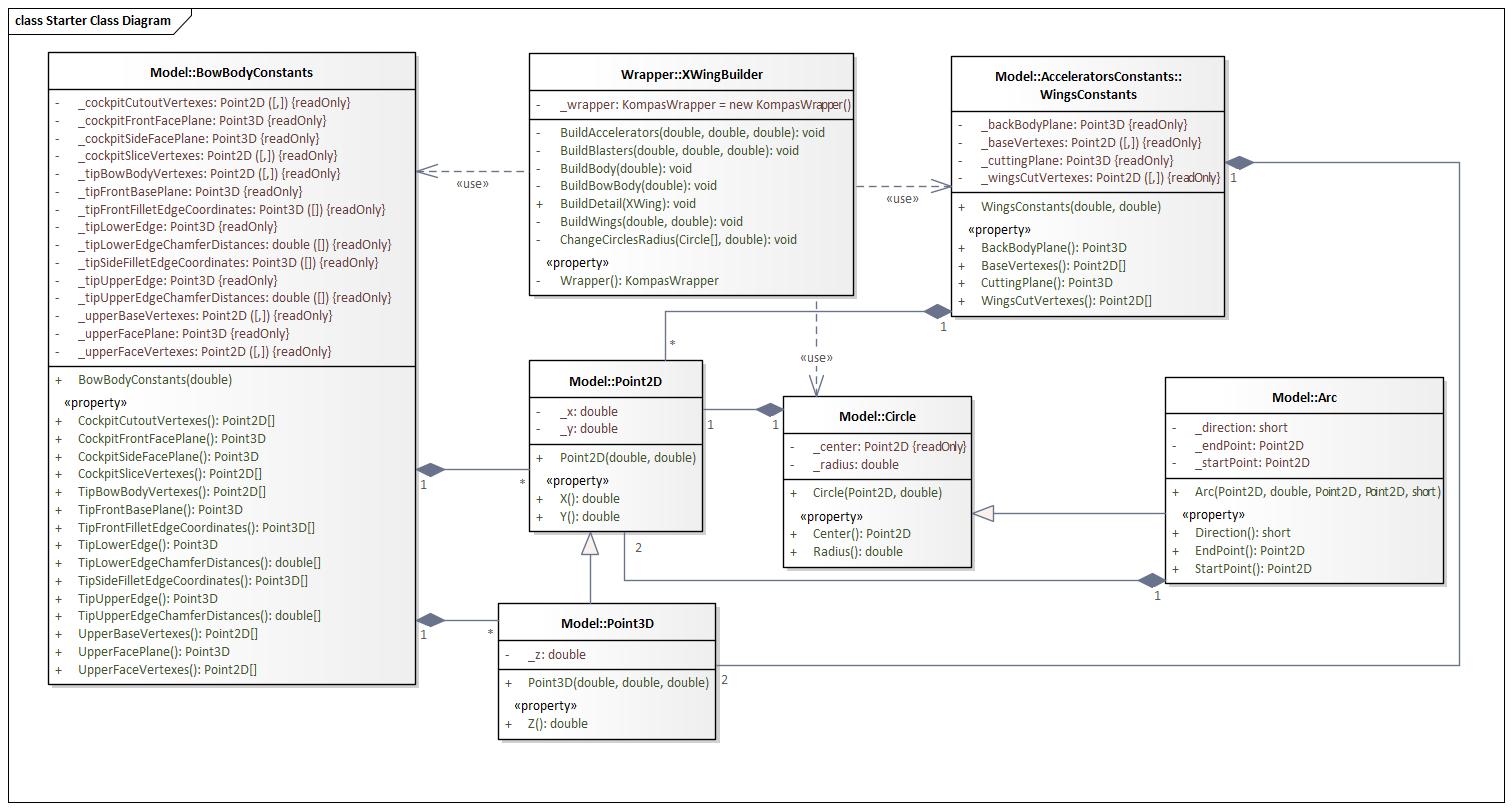
****

Рисунок 3.3 – Связь построителя с константами построения носовой части корпуса и крыльев

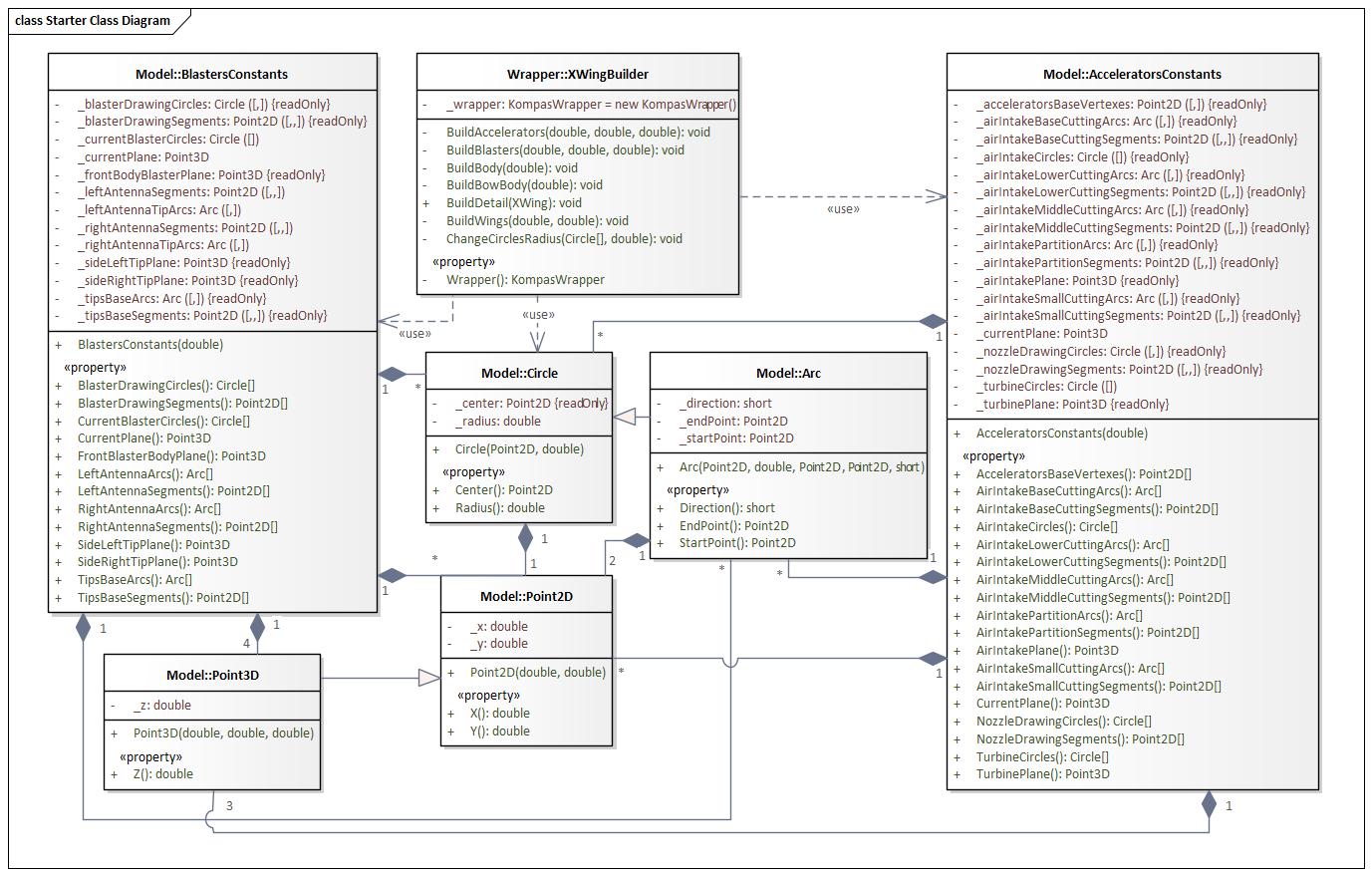


Рисунок 3.4 – Связи построителя с константами построения бластеров и ускорителей

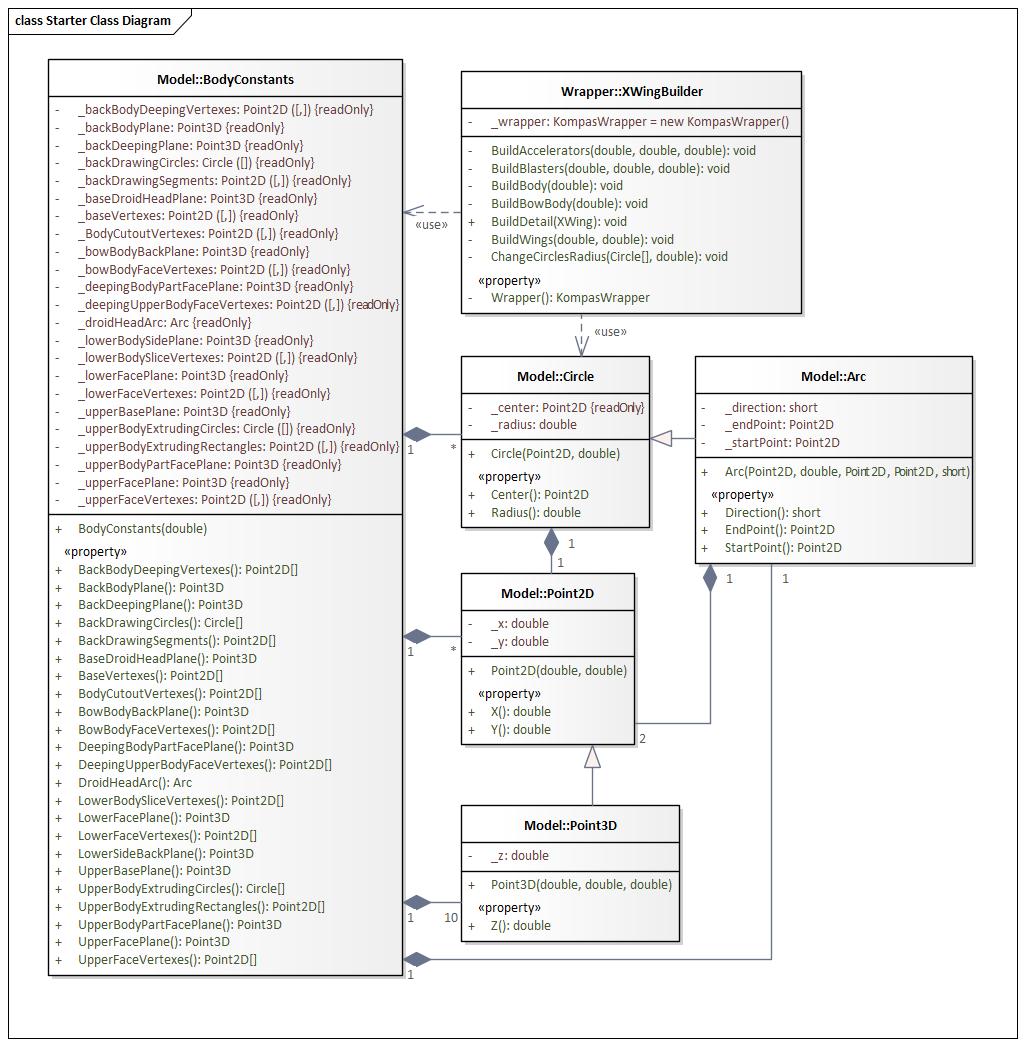
****

Рисунок 3.5 – Связь построителя с константами построения корпуса

Класс Parameter предназначен для описания любого параметра истребителя и его валидацию, класс XWing предназначен для объединения всех параметров воедино, также класс отвечает частично за валидацию. Класс KompasWrapper предназначен для взаимодействия с САПР «Компас-3D», в нём содержатся методы для связи с САПР и рисования в ней, класс XWingBuilder предназначен для построения звёздного истребителя, класс MainForm предназначен для описания пользовательского интерфейса. Перечисление XWingParameters необходимо для валидации параметров и корректной связи их с MainForm, статический класс ErrorList представляет из себя словарь, необходимый для вывода ошибок при вводе значений для построения детали в окно MainForm.

Классы Point2D, Point3D, Circle и Arc представляют собой соответственные фигуры точек двухмерного и трёхмерного пространства, круга и дуги. Классы BowBodyConstants, BodyConstants, WingsConstants, AcceleratorsConstants и BlastersConstants являются классами заранее вычисленных констант, необходимых для построения детали, поля этих классов через свойства передаются в соответствующие методы класса XWingBuilder. Связь между классами фигур и классами констант – композиция, кратность можно увидеть на диаграммах классов на рисунках 3.3-3.5. Класс XWingBuilder использует все эти классы.

# 3.2 Макеты пользовательского интерфейса

Пользовательский интерфейс (UI) – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы [11].

Плагин представляет собой пользовательскую форму с полями для ввода соответствующих параметров, макет можно увидеть на рисунке 3.6. Посредством кнопки «Построить» осуществляется запуск САПР «Компас-3D», на рабочей области которой построится трехмерная модель детали по заданным параметрам. Если построение детали выполняется несколько раз, то плагин не запускает несколько копий программы «Компас-3D», а создает в ней новый документ. В случае ввода значений параметров, не в ходящих в допустимый диапазон, поле для ввода окрашивается в розовый цвет и выводится окно, информирующее пользователя о некорректности введенного значения. Пример неправильного ввода представлен на рисунке 3.7.

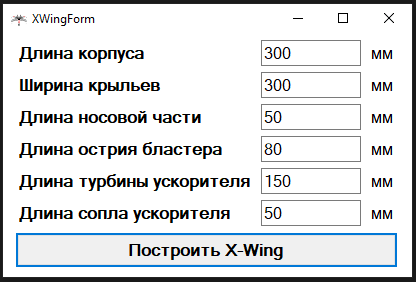


Рисунок 3.6 – Макет пользовательского интерфейса

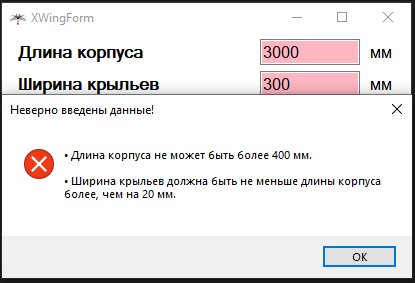


Рисунок 3.7 – Реакция приложения на ввод некорректных значений

# Список использованных источников

1. Официальный сайт системы автоматизированного проектирования «Компас-3D» [Электронный ресурс] – Режим доступа: https://kompas.ru/kompas-3d/about/ (дата обращения: 30.09.2022)
2. Компас (САПР) [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Компас\_(САПР) (дата обращения: 30.09.2022)
3. КОМПАС-3D для разработчиков [Электронный ресурс] – Режим доступа: https://kompas.ru/solutions/developers/ (дата обращения: 30.09.2022)
4. API [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/API (дата обращения: 30.09.2022)
5. ООО «АСКОН-Системы проектирования». Руководство пользователя KOMPAS-Invisible (API КОМПАС-3D), 2020. 5492 с.
6. External workbenches [Электронный ресурс] – Режим доступа: https://wiki.freecadweb.org/External\_workbenches (дата обращения: 06.10.2022)
7. Airplane Design Workbenchhttps [Электронный ресурс] – Режим доступа: https://github.com/FredsFactory/FreeCAD\_AirPlaneDesign (дата обращения: 06.10.2022)
8. Rocket Workbench [Электронный ресурс] – Режим доступа: https://wiki.freecadweb.org/Rocket\_Workbench (дата обращения: 06.10.2022)
9. UML-диаграммы классов [Электронный ресурс] – Режим доступа: https://prog-cpp.ru/uml-classes (дата обращения: 06.10.2022)
10. Enterprise Architect (software) [Электронный ресурс] – Режим доступа: https://en.wikipedia.org/wiki/Enterprise\_Architect\_(software) (дата обращения: 06.10.2022)
11. Интерфейс пользователя [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Интерфейс\_пользователя (дата обращения: 06.10.2022)