Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ**

**УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**РАЗРАБОТКА ПЛАГИНА «МОДЕЛЬ ПАСАЖИРСКОГО САМОЛЁТА»**

Пояснительная записка по курсовому проекту

по дисциплине «Разработка САПР»

Студент гр. 580-2

\_\_\_\_\_\_\_\_\_\_\_Н. Г. Богданов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2014 г.

Руководитель

м.н.с. ЛИКС каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_А. А. Калентьев

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2014 г

**РЕФЕРАТ**

Курсовой проект содержит: страниц –33 , источников – 16, рисунков – 17, таблиц – 11.

КОМПАС 3D, AIRLINER, ПЛАГИН, ПАССАЖИРСКИЙ САМОЛЁТ

Целью данной работы является разработка и реализация плагина «Модель пассажирского самолёта» на базе системы КОМПАС 3D V15, с использованием методов и свойств интерфейса API 5.

В результате был создан плагин «Модель пассажирского самолёта», стоящий трехмерную модель ведомого вала в системе КОМПАС 3D V15.

Пояснительная записка выполнена в текстовом редакторе Microsoft Office 2013.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОННИКИ (ТУСУР)

Утверждаю

Зав. кафедры КСУП

\_\_\_\_\_\_\_\_\_\_\_\_Ю.А. Шурыгин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_г.

ЗАДАНИЕ

на курсовой проект по дисциплине «Разработка САПР»

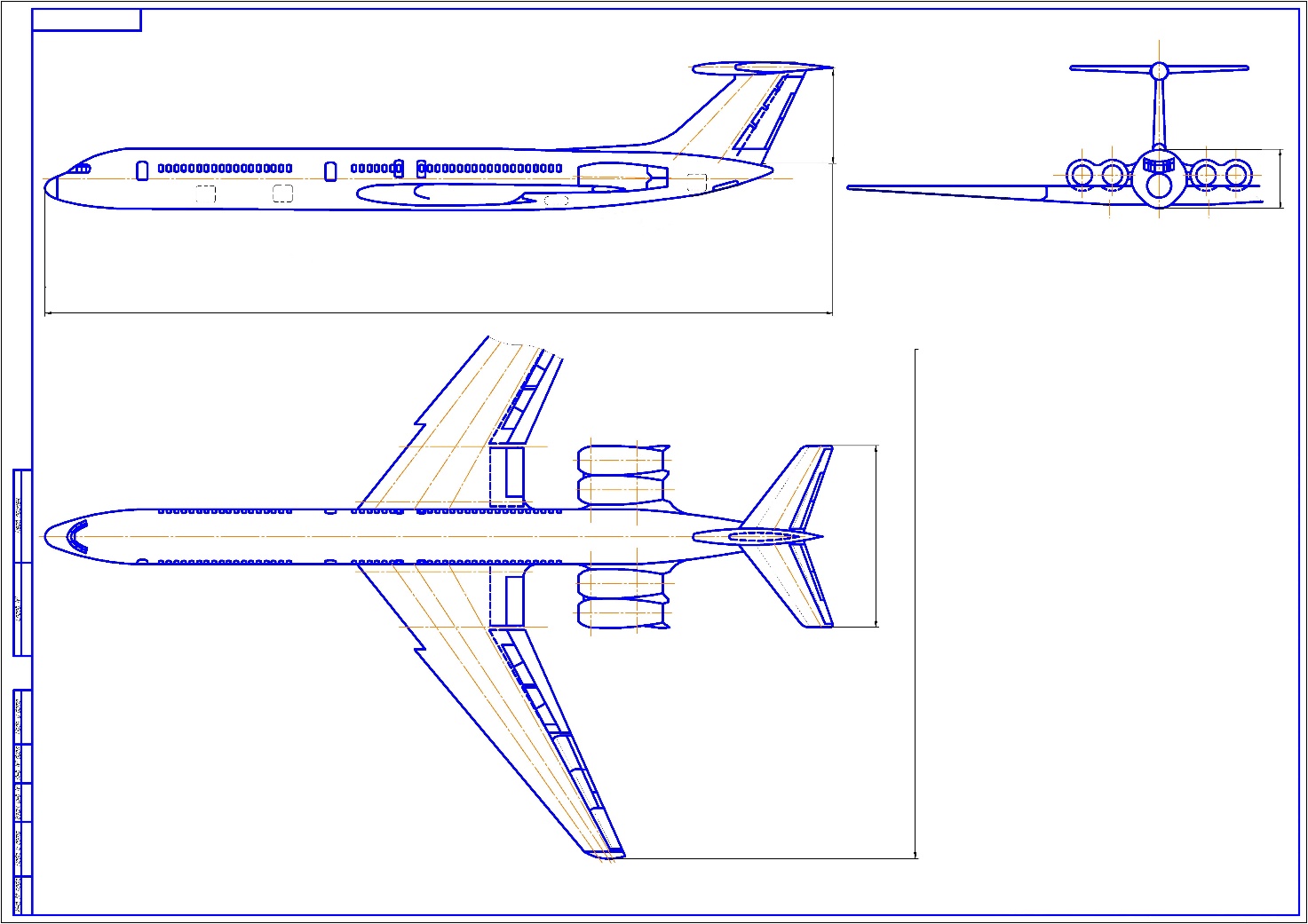
Студенту гр. 580-2 Богданову Николаю Григорьевичу

Тема: Создание библиотеки «пассажирский самолёт» на базе системы «Компас – 3D».

1. Цель работы:

Реализовать библиотеку для построения элемента «пассажирский самолёт».

1. Срок сдачи студентом работы: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2014 г.
2. Функциональные требования:
   1. Вывод диалогового окна для ввода следующих параметров:
      1. Длина самолёта (от 30 до 90 метров);
      2. Диаметр фюзеляжа (от 3 до 7 метров);
      3. Размах крыла (от 25 до 90 метров);
      4. Горизонтальное расположение крыла на корпусе (зависит от длины самолёта);
      5. Вертикальное расположение крыла на корпусе (зависит от диаметра фюзеляжа);
      6. Угол стреловидности крыла (от 20 до 40 градусов);
      7. Высота вертикального стабилизатора (3 до 8 метров);
      8. Длина горизонтального стабилизатора (5 до 10 метров);
      9. Количество двигателей (2, 4, 6 или 8);
   2. На основе введенных данных обеспечить построение корректной трехмерной модели в системе «Компас – 3D».
3. Конструкция самолёта представлена на рисунке 1.



7

5

3

6

2

1

1. Длина самолёта;
2. Диаметр фюзеляжа;
3. Размах крыла;
4. Угол крыла;
5. Высота вертикального стабилизатора;
6. Длина горизонтального стабилизатора;
7. Двигатель.

Рисунок 1 — Конструкция самолёта

1. Сфера применения

Библиотека предназначена для предприятий, целью которых, является проектирование и изготовление пассажирских самолётов.

1. Требования к программной части:
2. Программа должна работать на операционных системах: Windows 7 (x64), Windows 8.1 (x64);
3. Аппаратные требования для программы: процессор с тактовой частотой 2 ГГц или выше с поддержкой SSE2; 2 Гб оперативной памяти; 1,5 Гб пространства на жестком диске; .NET Framework 4.5;
4. Программа должна быть выполнена на языке C# в среде Microsoft Visual Studio 2012, .NET Framework 4.5, для системы Компас – 3D V15.
5. Содержание пояснительной записки

* титульный лист;
* реферат;
* техническое задание;
* содержание;
* введение;
* постановка задачи;
* основная часть;
* заключение;
* список литературы;
* приложения.

1. Дата выдачи задания: «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2014 г.

Руководитель м.н.с. ЛИКС кафедры КСУП:

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

Задание принял к исполнению:

Богданов Н. Г. \_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 8](#_Toc405553761)

[1 Постановка и анализ задачи 9](#_Toc405553762)

[1.1 Описание предмета проектирования 9](#_Toc405553763)

[1.2 Выбор инструментов и средств реализации 10](#_Toc405553764)

[1.3 Описание API КОМПАС 3D 12](#_Toc405553765)

[1.4 Назначение плагина 14](#_Toc405553769)

[2 Обзор аналогов плагинов для САПР КОМПАС 3D 15](#_Toc405553771)

[2.1 Плагин PDF 15](#_Toc405553772)

[2.2 Экспорт из Компас-3D в формате 3D PDF 15](#_Toc405553773)

[3 Описание технических и функциональных аспектов проекта 17](#_Toc405553774)

[3.1 Диаграмма вариантов использования (Use Cases) 18](#_Toc405553775)

[3.2 Диаграмма классов 19](#_Toc405553776)

[3.3 Диаграмма пакетов 21](#_Toc405553777)

[4 Описание программы для пользователя 22](#_Toc405553778)

[5 Тестирование 26](#_Toc405553779)

[5.1 Функциональное тестирование 26](#_Toc405553780)

[5.2 Unit-тестирование 30](#_Toc405553781)

[5.3 Нагрузочное тестирование 30](#_Toc405553782)

[Заключение 32](#_Toc405553783)

[Список использованных источников 33](#_Toc405553784)

[Приложение А 35](#_Toc405553785)

[Приложение Б 39](#_Toc405553786)

[Приложение В 40](#_Toc405553787)

# Введение

В настоящее время проектирование в своем понимании представляет собой процесс автоматизированный и в некотором роде программно-аппаратный. Проектировщику, который занимается разработкой сложного механизма, или устройства, требующего больших расчетов, математических вычислений при построении модели и высокой точности, подходят системы автоматизации проектных решений — САПР [1].

САПР позволяют уменьшить финансовые затраты на разработку макета (модели) проекта (объекта), а также сократить время, которое тратить проектировщик на создание модели объекта и составление проектной документации документации.

В каждой крупной САПР есть свой средства для разработки, который предоставляются с целью дать возможность разработчикам расширить функционал данной системы под свои конкретные нужды. Это позволяет в довольно короткие сроки автоматизировать работу и сократить время, деньги, силы проектировщиков и всей организации в целом, у которой стоит цель создания или проектирования изделий. Данным средством является API — программируемый интерфейс приложения [2]. Это набор готовых средств: классов, процедур, функций, структур и т.д. API позволяет определить функциональность, которую предоставляет приложение, при этом абстрагируясь от того, как она реализована.

Расширение функционала в основном подразумевает разработку плагина или библиотеки на основе предоставленного API. В данном курсовом проекте стоит задача разработки плагина для построения 3D модели ведомого вала в автоматизированном режиме. Плагин — независимо компилируемый программный модель, динамически подключаемый к основной программе, предназначенный для расширения или использования ее возможностей [3].

В качестве системы, которая предоставляет API и для которой стоит задача разработать плагин была взята САПР КОМПАС-3D.

# Постановка и анализ задачи

В рамках данного курсового проекта в соответствии с техническим заданием требовалось разработать плагин, который на основе входных параметров, в САПР КОМПАС-3D выполняет построение модели «Пассажирский самолёт».

## Описание предмета проектирования

Предметом проектирования является конструкция самолёта [8]. Самолет – это конструкция из фюзеляжа, крыльев и двигателей. Ниже на рисунке 3.1 представлен вид на 3D модель самолёта.

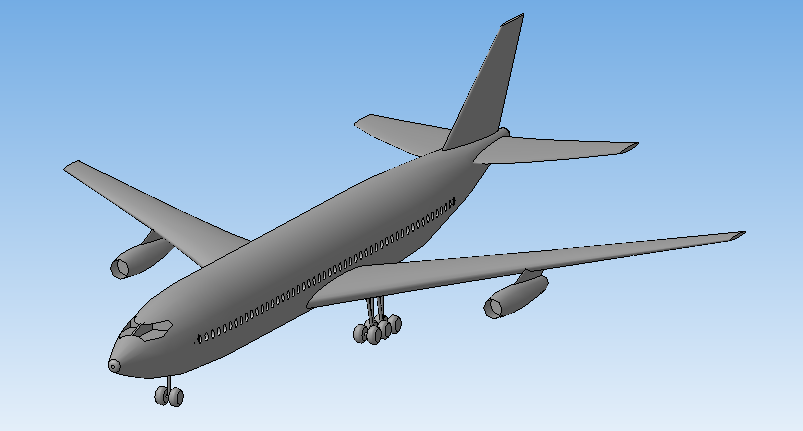


Рисунок 1.1 — Вид на 3D модель самолёта

Основой конструкции является фюзеляж, или корпус, который соединяет все части машины (крыло, стабилизатор, киль и т.д.).

## Выбор инструментов и средств реализации

В связи с требованием технического задания программа выполнена на языке C# в среде Microsoft Visual Studio 2012 [5], с использованием .NET Framework 4.5 [6], для системы КОМПАС 3D V15. Инструментом тестирования и создания модульных тестов был выбран стандартный обозреватель тестов [10] среды Microsoft Visual Studio 2012.

Взаимодействие плагина с системой КОМПАС 3D осуществляется посредством интерфейсов, называемых API. В КОМПАС 3D на данный момент существует API двух версий API 5 и API 7. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для курсового проекта была выбрана версия API 5, так как для полноценной реализации плагина «Шлицевая гайка» достаточно методов и свойств интерфейсов API 5.

«Компас-3D» — семейство [систем автоматизированного проектирования](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) с возможностями оформления проектной и конструкторской документации согласно стандартам серии [ЕСКД](https://ru.wikipedia.org/wiki/%D0%95%D0%B4%D0%B8%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8) и [СПДС](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B9_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8_%D0%B4%D0%BB%D1%8F_%D1%81%D1%82%D1%80%D0%BE%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%B0). Разрабатывается российской компанией «[Аскон](https://ru.wikipedia.org/w/index.php?title=%D0%90%D1%81%D0%BA%D0%BE%D0%BD&action=edit&redlink=1)». Название линейки является акронимом от фразы «комплекс автоматизированных систем», в торговых марках используется написание заглавными буквами — «КОМПАС».

«Компас» выпускается в нескольких редакциях: «Компас-График», «Компас-Строитель»(Ранее «Компас-СПДС»), «Компас-3D», «Компас-3D LT», «Компас-3D Home», «Компас-3D Учебная версия» . «Компас-График» может использоваться и как полностью интегрированный в «Компас-3D» модуль работы с чертежами и эскизами, и в качестве самостоятельного продукта, предоставляющего средства решения задач 2D-проектирования и выпуска документации. «Компас-3D LT» и «Компас-3D Home» предназначены для некоммерческого использования, «Компас-3D» без специализированной лицензии не позволяет открывать файлы, созданные в этих программах. Такая специализированная лицензия предоставляется только учебным заведениям.

Существуют коммерческие и некоммерческие версии Компас-3D. К коммерческим относятся: «Компас-3D», «Компас-График», «Компас-Строитель».

**Компас-3D**

Основные компоненты «Компас-3D» — собственно система трёхмерного твердотельного моделирования, универсальная система автоматизированного проектирования «Компас-График» и модуль проектирования спецификаций, а также набор специализированных библиотек и приложений.

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. [Параметрическая технология](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) позволяет быстро получать модели типовых изделий на основе однажды спроектированного прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства.

Ключевой особенностью «Компас-3D» является использование собственного математического ядра "C3D" и параметрических технологий, разработанных специалистами компании «Аскон».

**Компас-График**

Универсальная система автоматизированного 2D-проектирования «Компас-График» и модуль проектирования спецификаций, также набор 2D-библиотек.

Система «Компас-График» предназначена для создания чертежей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы, схем, спецификаций, таблиц, инструкций, расчётно-пояснительных записок, технических условий, текстовых и прочих документов. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства.

**Компас-Строитель**

КОМПАС-Строитель V15 — система автоматизированного проектирования для строительства. Это САПР, решающий задачи создания рабочей документации согласно всем стандартам СПДС. Продукт создан на основе КОМПАС-График. Возможности КОМПАС-Строитель позволяют работать с чертежами, созданными в других CAD-системах [7].

## Описание API КОМПАС 3D

Большинство применяемых в промышленности трехмерных САПР могут быть использованы как основа для построения специализированной САПР, решающей задачу расчета и проектирования конкретного класса изделий. При этом необходимо объединить расчетный модуль, определяющий размерные и иные параметры проектируемого объекта, с уже имеющимся в САПР, трехмерным геометрическим ядром (Рис. 1.2).

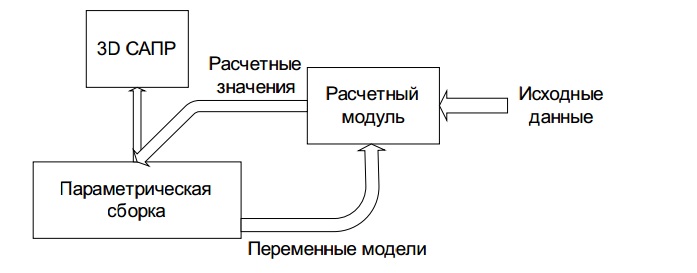


Рисунок 1.2 — Структура специализированной САПР

Для этого сначала создается параметрическая сборка проектируемого механизма, в которой ряд размеров вынесен в переменные модели. Расчетный модуль (это внешний exe-файл или подключаемая к САПР dll-библиотека, написанные, например, на Delphi) может рассчитать требуемые значения переменных модели и автоматически изменить их, в результате чего будет получен новый вариант 3D сборки. Таким образом, сразу же после расчета будет получена новая геометрия изделия. Разумеется, такой способ накладывает ограничения на функциональность специализированной САПР: можно только менять размеры, но не добавлять или удалять детали и/или их конструктивные элементы (скажем, не удастся сделать модель зубчатого колеса с произвольным числом зубьев). С другой стороны, в большинстве случаев работа конструктора как раз и сводится к модификации ранее созданной геометрии узла в соответствии с новыми расчетными данными, и здесь описываемая специализированная САПР полностью выполняет задачу автоматизации конструкторского труда, выполняя и расчет, и построение модели. Очевидно, главную сложность представляет не столько выполнение расчетов, сколько организация взаимодействия расчетного модуля и САПР. Исторически сложилось, что большинство современных САПР не поддерживают СОМ-технологию, что дополнительно затрудняет управление ими из внешней программы. Как правило, такое управление осуществляется при помощи технологии API (Application Programming Interface). API- технология предоставляет программисту набор процедур и функций для управления САПР, но не дает прямого доступа к свойствам и методам объектов внутри САПР, что делает код программы несколько более громоздким и менее понятным.

Для Компас-3D существует два вида API — версии 5 и версии 7. API 7 — это усовершенствованный и более новый вариант программных интерфейсов пятой версии. На самом деле обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда, полагаю, очевидно, что обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе.

В основном, для создания полноценных подключаемых модулей достаточно методов и свойств интерфейсов API 5 [8].

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно с помощью экспортной функции CreateKompasObject(). Методы этого интерфейса, реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа). Ниже на таблице 1.1 представлена часть методов интерфейса KompasObject.

Таблица 1.1 — Некоторые методы интерфейса KompasObject.

|  |  |
| --- | --- |
| ActiveDocument2D | Позволяет получить указатель на активный графический документ |
| ActiveDocument3D | Позволяет получить указатель на активный графический трехмерный документ |
| Document2D | Позволяет получить указатель на интерфейс графического документа (чертежа или фрагмента) |
| Document3D | Позволяет получить указатель на интерфейс трехмерного графического документа (детали или сборки) |
| GetDynamicArray | Возвращает указатель на интерфейс динамического массива |

## Назначение плагина

Разработка плагина для Компас-3D представляет процесс, который включает в себя использование интерфейсов Компас API и вложенных в них методов. Каждая основная операция представляется методом определенного интерфейса. Очевидно, что точно таким же образом можно автоматизировать выполнение любой последовательности любых трехмерных формообразующих операций, построение вспомогательной геометрии, формирование массивов и пр., что в результате даст вам готовую трехмерную модель изделия.

# Постановка и анализ задачи

## Плагин PDF

Данный плагин позволяет производить экспорт моделей и сборок из Компас-3D в формат PDF формат [9]. Основной особенностью является возможность интерактивного взаимодействия пользователя с сохраненной 3D сценой внутри PDF файла. Например, пользователь может вращать, масштабировать, передвигать детали и сборки внутри 3D PDF файла. Также доступно создание анимации сборки и разборки изделий. Это полезно для подготовки интерактивных сборочных инструкций, создания маркетинговых материалов, презентаций, а также для налаживания взаимодействия между проектировщиками и заказчиками. В подобных ситуациях традиционным подходом являлся экспорт сборки или детали Компас-3D в промежуточный формат и дальнейшее сохранение в формат 3D PDF. Используемый подход в плагине исключает использование промежуточных файлов для осуществления 3D преобразования, что существенно повышает качество выходной 3D модели в формате PDF. Ключевые возможности:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;
* создание анимаций имитирующих естественный порядок сборки и разборки создание имитации анимации гибки листовых тел;
* вставка в существующие PDF документы, содержащие основной текст, фоновые картинки, таблицы спецификаций, эмблемы, логотипы;
* пакетный режим для поочередной конвертации всех файлов.

## Экспорт из Компас-3D в формате 3D PDF

Пользователям Компас-3D стала доступна функция экспорта созданных трехмерных моделей и дальнейшего их использования в создании технической документации. Экспорт происходит в формате 3D PDF [10].

Главной особенностью является то, что пользователь по-прежнему имеет возможность интерактивно взаимодействовать с 3D сценой, находясь внутри файла 3D PDF. То есть пользователь может передвигать детали, вращать их, масштабировать, передвигать сборки внутри самого файла. Пользователь также может создать анимацию сборки и разборки изделия. Этот функционал очень удобен. Он используется при создании презентаций, маркетинговых материалов, при подготовке интерактивных сборочных конструкций. Он значительно упрощает взаимодействие между заказчиками и проектировщиками.

Компания Visual Technology Services Ltd. Из Великобритании разработала плагин PDF3D, предоставляющий доступ к описанным выше возможностям. Основной функционал плагина:

* сохранение сборок и деталей для интерактивного просмотра в формате 3D PDF с помощью программы Adobe Reader;
* создание анимации, имитирующей естественный порядок разборки и сборки;
* создание анимации, имитирующей гибки листового тела;
* возможность вставки в PDF документ, содержащий основной текст, фоновых картинок, таблиц, логотипов, эмблем, спецификаций и т.д.

КОМПAС-3D является открытой системой, что позволяет создавать дополнительные программные модули (пользовательские библиотеки) и применять их во время работы над документами. Таким образом, стандартные возможности чертежно-графического редактора и трехмерного моделирования могут быть дополнены исходя из тех специальных задач, которые приходится решать пользователю.

# Описание технических и функциональных аспектов проекта

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценария действий) использован стандарт UML[11].

UML [язык](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [графического](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B7%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) описания для [объектного моделирования](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) в области [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F). UML является языком широкого профиля, это – [открытый стандарт](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82), использующий графические обозначения для создания [абстрактной модели](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C) [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), 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-моделей возможна [генерация кода](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%BE%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F).

При использовании UML были построены: диаграмма использования, диаграмм классов, диаграмма пакетов.

## Диаграмма вариантов использования (Use Cases)

Диаграмма вариантов использования отражает возможный выбор действий (выбора состояния) пользователя внутри системы, установлено пятнадцать состояний (Рисунок 3.1).



Рисунок 3.1 — Диаграмма вариантов использования



## Диаграмма классов



Рисунок 3.2 — Диаграмма классов до модификации



Рисунок 3.3 — Диаграмма классов после модификации

## Диаграмма пакетов

Диаграмма пакетов служит для представления взаимодействия пакетов внутри модели системы [12].

Пакет General — является приемником данных от пользователя с формы и отвечает за построение объекта.

Пакет Manager — содержит в себе класс для вызова процедуры построения.

Пакет Data — содержит в себе класс хранения данных.



Рисунок 3.4 — Диаграмма пакетов

# Описание программы для пользователя

Для построения модели пассажирского самолёта необходимо запустить плагин. Плагин представляет собой пользовательскую форму с элементами управления значениями параметров объекта. Запуск построения и САПР Компас-3D осуществляется кнопкой «Построить». Интерфейс программы представлен на рисунке 4.1.

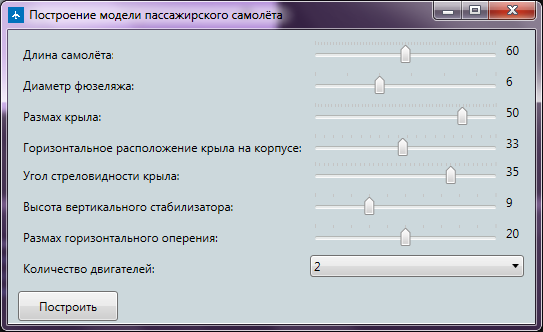


Рисунок 4.1 — Интерфейс программы

После нажатия кнопки «Построить» откроется Компас-3D, после чего начнется построение модели.

# Тестирование

## Функциональное тестирование

В ходе функционального тестирования плагина «Пассажирский самолёт» проверялась корректность построения модели самолёта при минимальных и максимальных задаваемых параметрах.

Ниже представлены рисунки построенных моделей с минимальными и максимальными заданными параметрами (рисунок 5.1 и 5.2)

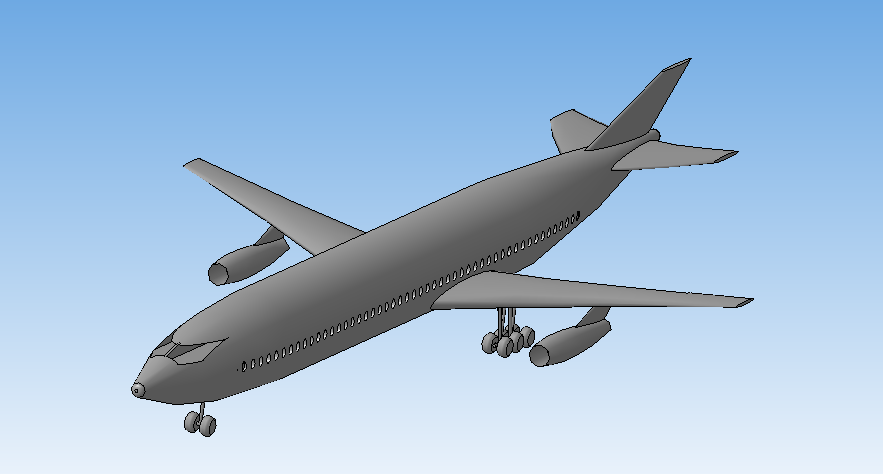


Рисунок 5.1— Самолёт с минимальными заданными параметрами

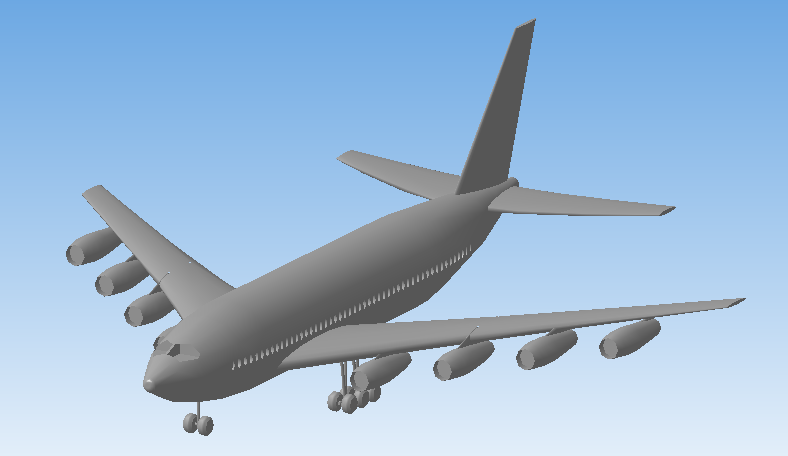


Рисунок 5.2 — Самолёт с максимальными заданными параметрами

Тестирование показало, что был получен желаемый результат в построении модели.

## Unit-тестирование

В целях проверки корректности работы методов и свойств классов при помощи встроенного обозревателя тестов в Visual Studio было проведено unit-тестирование [14]. Проверялись конструкторы и доступные поля и методы, возвращающие значения.

* InfAirUT – класс тестирующий свойства и методы класса DataContainer, описание которого в Приложении Б (Таблица Б.1).

## Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [15]. Для тестирования выбран класс Stopwatch из пространства имен System.Diagnostics [16]. Методы данного класса позволяют с высокой точностью определить работу определенных частей кода. Был произведен нагрузочный тест из 5 повторений с изменением параметров.

Тест проводился с стандартными параметрами:

* длина самолёта – 60 метров;
* диаметр фюзеляжа – 6 метров;
* размах крыла – 50 метров;
* горизонтальное расположение крыла на корпусе – 33;
* угол стреловидности крыла – 35°
* высота вертикального стабилизатора – 9;
* размах горизонтального оперения – 20.

Изменялось лишь количество двигателей (2, 4, 6, 8). Измерялось количество затраченной оперативной памяти на одну сборку модели и затраченное время за все 5 построений. Результаты измерений занесены в диаграммы, представленные ниже (рисунок 5.3 и 5.4).

Рисунок 5.3 — Потребление RAM за одну сборку

Рисунок 5.3 —Время построения 5 сборок

# Заключение

В ходе проектирования системы были изучены основные этапы проектирования программного продукта, изучена предметная область предмета проектирования, аналоги предмета проектирования, также было изучено API системы автоматизированного проектирования Компас-3D. В результате полученных знаний возможна реализация плагина для автоматизации построения объекта «Пассажирский самолёт» в Компас-3D.

# Список использованных источников

1 Норенков И.П. «Основы автоматизированного проектирования». Издательство: МГТУ; Москва.:, 2002. – 336 с.

2 API [Электронный ресурс]. — Режим доступа: <http://en.wikipedia.org/wiki/Application_programming_interface> (дата обращения 12.11.2014)

3 Плагин [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 11.11.2014)

4 Ведомый вал [Электронный ресурс]. — Режим доступа: <http://auto.meta.ua/useful/dictionary/%C2> (дата обращения 05.11.2014)

5 Visual Studio 2010 [Электронный ресурс]. — Режим доступа: <http://www.microsoft.com/ru-ru/softmicrosoft/VisualStudioExpress.aspx> (дата обращения 05.11.2014)

6 .NET Framework [Электронный ресурс]. — Режим доступа: <http://www.microsoft.com/ru-ru/download/details.aspx?id=17718> (дата обращения 15.11.2014)

7 Компас (САПР) [Электронный ресурс]. — Режим доступа: <http://kompas.ru/> (дата обращения 15.11.2014)

8 API5 [Электронный ресурс]. — Режим доступа: <http://www.e-reading.link/chapter.php/127274/102/Kidruk_-_KOMPAS-3D_V10_na_100__.html> (дата обращения 10.11.2014)

9 Плагин PDF [Электронный ресурс]. — Режим доступа: <http://gkmsoft.ru/allcatalog/pdf2dkompas_plugin/> (дата обращения 11.11.14)

10 3D PDF [Электронный ресурс]. — Режим доступа: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения 30.03.14)

11 UML [Электронный ресурс]. — Режим доступа: <http://www.uml.org/> (дата обращения 01.11.2014)

12 Диаграмма пакетов [Электронный ресурс]. — Режим доступа: <http://blogs.rsdn.ru/effective/5545045> (дата обращения 01.11.2014)

13 ГОСТ 3480-76 [Электронный ресурс]. — Режим доступа: <http://www.gosthelp.ru/gost/gost33761.html> (дата обращения 10.10.2014)

14 Unit-тестирование [Электронный ресурс]. — Режим доступа: <http://www.protesting.ru/testing/levels/component.html> (дата обращения 10.10.2014)

15 Нагрузочное тестирование [Электронный ресурс]. — Режим доступа: <http://www.protesting.ru/automation/performance.html> (дата обращения 10.10.2014)

16 System.Diagnostics [Электронный ресурс]. — Режим доступа: <http://msdn.microsoft.com/ru-ru/library/system.diagnostics(v=vs.110).aspx> (дата обращения 10.10.2014)

# Приложение А

(Справочное)

В таблицах приложения для обозначения модификаторов доступа полей приняты следующие условные знаки:

* «#» − обозначение protected (защищенного) поля;
* «−» − обозначение private (открытого) поля;
* «+» − обозначение public (открытого) поля.
* Таблица А.1 – Описание полей и методов класса MainWindow

|  |  |
| --- | --- |
| Наименование | Описание |
| + InfAirliner: InfAirliner | Данные о размерах частей самолёта и их расположения |
| + Manager: Manager | Взаимодействие между плагином и Компас3D |
| + MainWindow() | Конструктор |
| + ChangeInfAirliner() | Изменение данных о частях самолёта |
| - SliderLengthAir\_ValueChanged(object, RoutedPropertyChangedEventArgs<double>) | Отслеживание изменение значения в “SliderLengthAir” и установка граничных значений на других “Slider” в зависимости от изменённого значения |
| - SliderWingpan\_ValueChanged(object, RoutedPropertyChangedEventArgs<double>) | Отслеживание изменение значения в “ SliderWingpan ” и установка граничных значений на других “Slider” в зависимости от изменённого значения |

* Таблица А.2 – Описание полей и методов класса InfAirliner

|  |  |
| --- | --- |
| Наименование | Описание |
| + LengthOfAircraft: double | Длина самолёта |
| + FuselageDiameter: double | Диаметр фюзеляжа |
| + Wingspan: double | Размах крыла |
| + HorizontalPositionWing: double | Горизонтальное расположение крыла |
| + SweepbackAngle: double | Угол стреловидности крыла |
| + HeightOfKeel: double | Высота киля |
| + LengthOfHorizontalStabilizer: double | Длина вертикального стабилизатора |
| +InfAirliner(double lengthOfAircraft, double fuselageDiameter, double wingspan, double horizontalPositionWing, double verticalPositionWing, double sweepbackAngle, double heightOfKeel, double lengthOfHorizontalStabilizer) | Коструктор инициализирующий данные |

* Таблица А.3 – Описание полей и методов класса Manager

|  |  |
| --- | --- |
| Наименование | Описание |
| + Manager() | Конструктор |
| - \_creadeModel: CreatingModel | Создание модели |
| + OpenKompas3D() | Открытие Компас3D |
| + InitializeModel(InfAirliner infAirliner) | Инициализация модели |

* Таблица А.4 – Описание полей и методов класса CreatingModel

|  |  |
| --- | --- |
| Наименование | Описание |
| - \_document3D: ksDocument3DNotify7 | Документ, содержащий 3D-модель или сборку |
| - \_kompasObject: KompasObject | Главным интерфейсом API системы КОМПАС |
| - \_infAirliner: InfAirliner | Данные о размерах частей самолёта и их расположения |
| - \_entityCollection: List<ksEntity> | Коллекция эскизов для операций |
| - \_mirrorCopyEntityCollection: List<ksEntity> | Коллекция эскизов для резкального отражения |
| - \_part: ksPart | 3D модель |
| + CreatingModel(InfAirliner) | Конструктор |
| - CreateFus() | Создание фюзеляжа самолёта |
| - CreateWing() | Создание крыла самолёта |
| - CreateEngines() | Создание двигателей самолёта |
| - CreateEngine(int curEng) | Создание одного двигателя самолёта с порядковым номером |
| - CreateMountEngine(int curEng, double engPos, double horizLocationEng) | Создание крепления двигателя к крылу самолёта. Входные параметры: порядковый номер двигателя, составляющая вертикальной позиции, составляющая горизонтальной позиции |
| - CreateTail() | Создание хвостовой части |
| - CreateWindows() | Создание окон |
| - CreateSideWindows(double xc, double yc, double a, double b, double angle, ksEntity plane, double numWindows): ksEntity | Создание боговых окон. Параметры: расположение окон, плоскость расположения, количество окон |
| - CreateChassis() | Создание шасси |
| - CreateWheel(double xc, double yc, double r, double move): ksEntity | Создание колёс. Входные параметры: расположение колеса, радиус колеса, плоскость нахождения |
| - CreateShaft(double xc, double yc, double r, double move): ksEntity | Создание вала соединяющий колёса. Входные параметры: расположение вала, радиус вала, плоскость нахождения |
| - CreateRack(double xc, double yc, double r): ksEntity | Создание стойки шасси. Входные параметры: расположение стойки, радиус стойки |
| - CreateStiffeningPlate(double xc, double yc, double r): ksEntity | Создание ребра жёсткости. Входные параметры: расположение ребра жёсткости, радиус ребра жёсткости |
| - CreateFoldingStrut(double xc, double yc, double r): ksEntity | Создание складывающего подкоса. Входные параметры: расположение складывающего подкоса, радиус складывающего подкоса |
| - CreateCircle(double x, double y, double radius, ksEntity plane):ksEntity | Создание окружности. Входные параметры: расположение окружности, радиус окружности, плоскость нахождения |
| - CreateEllipse (double xc, double yc, double a, double b, double angle, ksEntity plane):ksEntity | Создание эллипса. Входные параметры: расположение эллипса, ширина эллипса, длина эллипса, угол наклона эллипса, плоскость нахождения |
| - CreateRectangle (double x, double y, double height, double width, double angle, ksEntity plane):ksEntity | Создание прямоугольника. Входные параметры: расположение прямоугольника, ширина прямоугольника, длина прямоугольника, угол наклона прямоугольника, плоскость нахождения |
| - GetPlane(Obj3dType typePlane): ksEntity | Возвращает плоскость. Входные параметры: тип плоскости |
| - MovePlane(ksEntity entityPlane, bool direction, double move): ksEntity | Смещение плоскости |
| - ExtrudeLoft(List<ksEntity> entityCollection): ksEntity | Выдавливание по сечениям. Входные параметры: список сечений |
| - CutExtruded(ksEntity entityCut, bool side, int typeDirection, double depth): ksEntity | Вырезание выдавливанием. Входные параметры: эскиз |
| - Extrusion(ksEntity entity, bool forward, int typeDirection, double depth): ksEntity | Операция выдавливание. Входные параметры: эскиз, направление выдавливания, тип выдавливания, длина выдавливания |
| - CopyBody(ksEntity entity, ksEntity plane) | Зеркальное копирование тела относительно плоскости. Входные параметры: список копируемых частей, плоскость, относительно которой зеркалируется |
| - Rounding(double radius, int numEdge) | Скругление рёбер. Входные параметры: радиус скругления, номер ребра |

# Приложение Б

(Справочное)

В таблицах приложения для обозначения модификаторов доступа полей приняты следующие условные знаки:

* «+» − обозначение public (открытого) поля.

Таблица Б.1 – Описание полей и методов класса InfAirUT

|  |  |
| --- | --- |
| Наименование | Описание |
| + TestLengthOfAircraft() | Тестирование инициализации поля «LengthOfAircraft» |
| + TestFuselageDiameter () | Тестирование инициализации поля «FuselageDiameter» |
| + TestWingspan() | Тестирование инициализации поля «FuselageDiameter» |
| + TestHorizontalPositionWing () | Тестирование инициализации поля «HorizontalPositionWing» |
| + TestVerticalPositionWing () | Тестирование инициализации поля «VerticalPositionWing» |
| + TestSweepbackAngle () | Тестирование инициализации поля «SweepbackAngle» |
| + TestHeightOfKeel () | Тестирование инициализации поля «HeightOfKeel» |
| + TestLengthOfHorizontalStabilizer () | Тестирование инициализации поля «LengthOfHorizontalStabilizer» |
| + TestTypeQuantityOfEngine () | Тестирование инициализации поля «TypeQuantityOfEngine» |