Министерство образования и науки 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 БИБЛИОТЕКИ «ПОДШИПНИК» ДЛЯ «КОМПАС-3D» V15.2.**

Пояснительная записка

по дисциплине «Основы разработки САПР»

Студент гр. 584-2

\_\_\_\_\_\_\_\_\_\_\_\_ Н.С Лях

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

дата

Руководитель

к.т.н., доцент каф. КСУП

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

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

дата

Министерство образования и науки 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)

Утверждаю:

Зав. кафедрой КСУП

\_\_\_\_\_\_\_\_Ю.А. Шурыгин

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

по дисциплине «Основы Разработки САПР»

Выдано: Студенту группы 584-2 Ляху Никите Сергеевичу

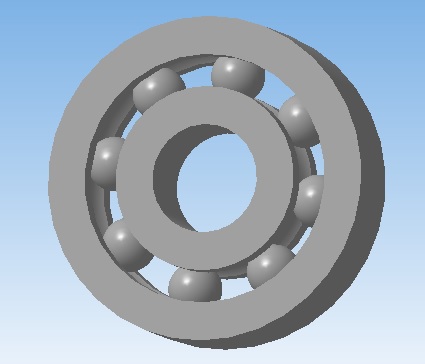
1. Тема проекта: Разработка плагина «Подшипник» для «Компас 3D v15.2».

Рисунок 1.1 – Подшипник

1. Срок сдачи студентом проекта: 31.12.2017.
2. Исходные данные:

Разработать плагин «Подшипник» для «Компас 3D v15.2».

1. Требования к плагину:

Плагин должен обеспечивать следующую функциональность:

* выводить диалоговое окно ввода для изменения следующих параметров:
* Диаметр внешнего обода – **d1** (рисунок 1.2);

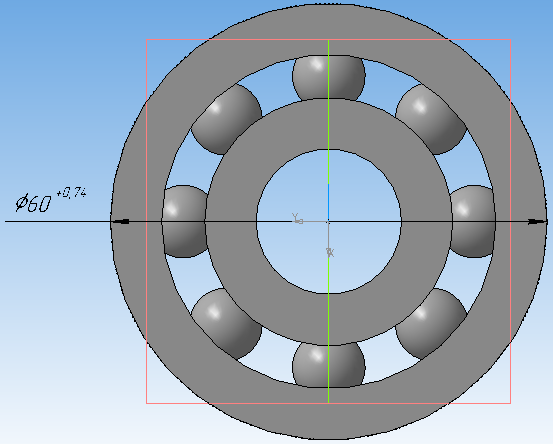


Рисунок 1.2 – Диаметр внешнего обода

* Диаметр внутреннего обода – **d2** (рисунки 1.3);

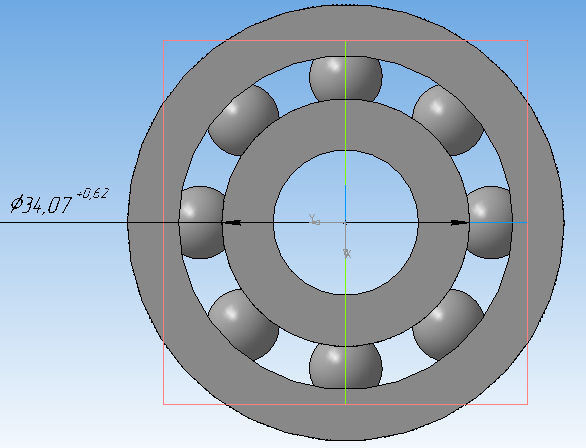


Рисунок 1.3 — Диаметр внутреннего обода

* Ширина внешнего обода – **a1** (рисунок 1.4)

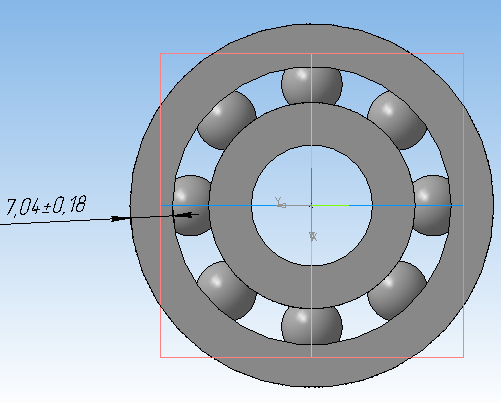


Рисунок 1.4 – Ширина внешнего обода – **a1**

* Ширина внутреннего обода – **a2** (рисунок 1.5);

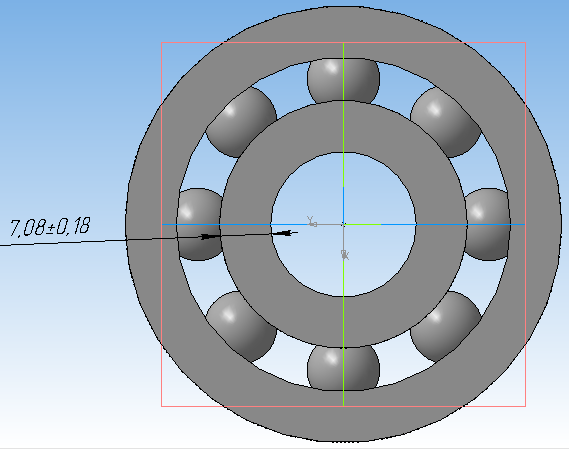


Рисунок 1.5 – Ширина внутреннего обода – **a2**

* Диаметр шарика – **d3** (рисунок 1.6)

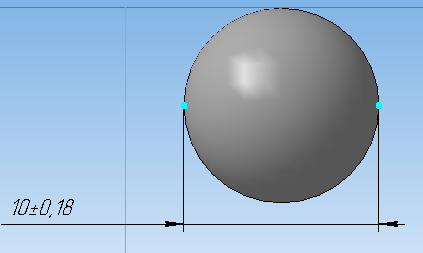


Рисунок 1.6 – Диаметр шарика – **d3**

* Глубина желоба – **b** (рисунок 1.7)

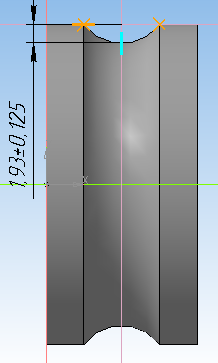


Рисунок 1.7 – Глубина желоба – **b**

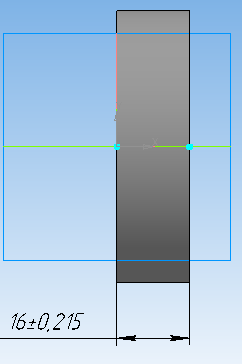
* Ширина подшипник – **а3** (рисунок 1.8)

Рисунок 1.8 – Ширина подшипника – **a3**

* контекстно зависимые ограничения: значение **b** не может быть больше величины **a1** и **a2**, диаметр **d3** не может быть больше ширины **а3**;
* обеспечивать построение трехмерной модели на графическом окне системы «Компас 3D v15.2» на основе введенных значений параметров;
* обеспечить корректность ввода данных и выводить информационное сообщение при вводе некорректных данных.

1. Сфера применения:

Плагин применим при изготовлении составных или сборочных деталей.

1. Минимальные требования к программной и аппаратной частям:

* программа должна работать на операционной системе: Windows 7 (x86 / x64) или более поздней версии;
* видеокарта с поддержкой Microsoft Direct3D 10® или более поздней версии;
* тип процессора: 64-разрядный процессор Intel® или AMD с тактовой частотой 2ГГц или выше;
* 8 Гб ОЗУ для сборок, которые содержат менее 500 деталей;
* Свободное место на жестком диске: 40 Гб.

1. Инструменты разработки:

* язык программирования — C#, версия .NET Framework — 4.5;
* среда разработки: Microsoft Visual Studio 2017.

Дата выдачи задания: 15.09.2017

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_А.А. Калентьев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2018 г.

Задание принял к исполнению:

\_\_\_\_\_\_\_\_\_\_Н.С. Лях

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2018 г.

**Реферат**

Индивидуальный проект 32 страниц, 30 иллюстраций, 8 таблиц, 10 использованных источников.

ПЛАГИН, ПОДШПНИК, САПР, КОМПАС-3D, ПРОЕКТИРОВАНИЕ, ПРОЕКТНАЯ ДОКУМЕНТАЦИЯ, API.

Объектом разработки является плагин «Подшипник» для САПР «КОМПАС-3D».

Цель проекта – создание плагина для построения трехмерной сборки деталей (подшипник) по введенным параметрам в «КОМПАС-3D».

В процессе выполнения проекта разработан плагин для построения подшипника в рабочей плоскости САПР «КОМПАС-3D», а также проектная документация к плагину.

Плагин предназначен для построения трехмерной модели подшипника. Взаимодействие с пользователем производится с помощью диалогового окна.

**Содержание**

[1 Введение 9](#_Toc504764353)

[2 Описание САПР 10](#_Toc504764354)

[2.1 Описание программы 10](#_Toc504764355)

[2.2 Описание API 10](#_Toc504764356)

[2.3 Обзор аналогов 13](#_Toc504764357)

[2.3.1 Плагин PDF 13](#_Toc504764358)

[2.3.2 Экспорт из КОМПАС-3D в формат 3D PDF 14](#_Toc504764359)

[3 Описание предмета проектирования 15](#_Toc504764360)

[4 Описание программы для пользователя 16](#_Toc504764361)

[5 Проект программы 16](#_Toc504764362)

[5.1 Диаграмма вариантов использования (Use Cases) 18](#_Toc504764363)

[5.2 Диаграмма классов 19](#_Toc504764364)

[5.3 Макет пользовательского интерфейса 20](#_Toc504764365)

[6 Тестирование программы 22](#_Toc504764366)

[6.1 Функциональные тесты 24](#_Toc504764367)

[6.2 Модульные тесты 28](#_Toc504764368)

[6.3 Нагрузочные тесты 29](#_Toc504764369)

[Список литературы 31](#_Toc504764370)

[Приложение А 32](#_Toc504764371)

# 1 Введение

В настоящее время проектирование в своем понимании представляет собой автоматизированный процесс 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 тратит проектировщик на создание модели объекта и составление проектной документации.

В каждой крупной САПР есть свой средства для разработки, которые предоставляются с целью дать возможность разработчикам расширить функциональность данной системы под свои конкретные нужды. 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 функциональности в основном подразумевает разработку плагина или библиотеки на основе предоставленного API. В данном проекте стоит задача разработки плагина для построения 3D модели подшипника в автоматизированном режиме. Плагин — независимо компилируемый программный модуль, 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» версии 15.2.

# 2 Описание САПР

## 2.1 Описание программы

КОМПАС-3D – система трехмерного проектирования, ставшая стандартом для тысяч предприятий, благодаря сочетанию простоты освоения и легкости работы с мощными функциональными возможностями твердотельного и поверхностного моделирования [4].

Ключевой особенностью продукта является использование собственного математического ядра С3D и параметрических технологий, разработанных специалистами АСКОН.

КОМПАС-3D обеспечивает поддержку наиболее распространенных форматов 3D-моделей (STEP, ACIS, IGES, DWG, DXF), что позволяет организовывать эффективный обмен данными со смежными организациями и заказчиками, использующими любые CAD / CAM / CAE-системы в работе.

2.2 Описание API

В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. [5]. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда 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.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно при работе под управлением внешнего приложения (контроллера) – после вызова стандартной системной функции. 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 в таблице 2.1 представлены свойства и методы интерфейса KompasObject, которые были использованы при разработке плагина.

Таблица 2.1 – Методы и свойства интерфейса KompasObject, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| ActivateControllerAPI() | bool | Метод для активации API Компас3D |
| Visible | bool | Свойство видимости приложения |

В таблице 2.2 представлены методы интерфейса ksEntity, которые были использованы при разработке плагина.

Таблица 2.2 – Методы интерфейса ksEntity, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |

В таблице 2.3 представлены свойства и методы интерфейса ksDocument2D, которые были использованы при разработке плагина.

Таблица 2.3 – Методы интерфейса ksDocument2D, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| ksLineSeg(double x1, double y1, double x2, double y2, int style) | int | Получить указатель на отрезок на двумерной плоскости либо 0 в случае ошибки |
| ksArcByAngle(double xc, double yc, double rad, double f1, double f2, short direction, int style) | int | Получить указатель на угол на двумерной плоскости либо 0 в случае ошибки |
| ksArcBy3Points(double x1, double y1, double x2, double y2, double x3, double y3, int style) | int | Получить указатель на угол на двумерной плоскости либо 0 в случае ошибки |

В таблице 2.4 представлены свойства и методы интерфейса ksDocument3D, которые были использованы при разработке плагина.

Таблица 2.4 – Методы интерфейса ksDocument3D, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | 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 |

В таблице 2.5 представлены методы интерфейса ksPart, которые были использованы при разработке плагина.

Таблица 2.5 – Свойства и методы интерфейса ksPart, используемые при разработке плагина

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection(short objType) | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| NewEntity(short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

В таблице 2.6 представлены типы объектов документа-модели, которые были использованы при разработке плагина.

Таблица 2.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_planeYOZ | Плоскость YOZ | ksPlaneParam |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_sketch | Эскиз | [ksSketchDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksSketchDefinition.htm) |
| o3d\_bossRotated | Выдавливание вращением | ksBossRotatedDefinition |
| o3d\_axis2Planes | Ось на пересечении плоскостей | ksAxis2PlanesDefinition |
| o3d\_circularCopy | Массив по концентрической сетке | ksCircularCopyDefinition |

2.3 Обзор аналогов

### 2.3.1 Плагин PDF

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат [6]. 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 плагина:

* 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.

### 2.3.2 Экспорт из КОМПАС-3D в формат 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 [7].

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.

# 3 Описание предмета проектирования

Подшипник – сборочный узел, являющийся частью опоры или упора и поддерживающий вал, ось или иную подвижную конструкцию с заданной жёсткостью. Фиксирует положение в пространстве, обеспечивает вращение, качение или линейное перемещение с наименьшим сопротивлением, воспринимает и передаёт нагрузку от подвижного узла на другие части конструкции.

Параметры Подшипника:

* ширина подшипника;
* диаметр внутреннего кольца;
* диаметр внешнего кольца;
* диаметр шарика;
* толщина колец подшипника.

На рисунке 3.1 представлен вид на 3D модель подшипника.

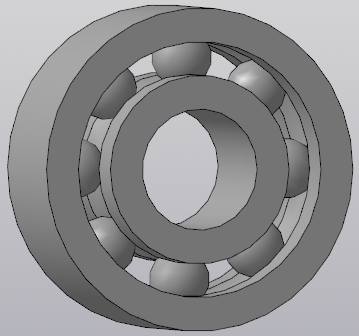


Рисунок 3.1 – Вид на 3D модель подшипника

**4 Описание программы для пользователя**

Чтобы построить подшипник, используя данный плагин, необходимо запустить плагин. В запущенном окне, в поле «Параметры подшипника» необходимо ввести параметры подшипника, а в поле «Форма элемента качения» выбрать «Цилиндр» или «Шарик» (рис. 4.1). Параметры подшипника могут принимать ненатуральные положительные значения. При вводе некорректных значений программа выдаст сообщение об ошибке в котором будут указаны все некорректные значения с пояснениями и возможными диапазонами корректных значений, как представлено на рисунке 4.2.

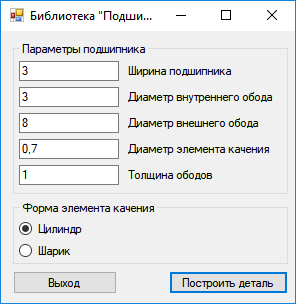


Рисунок 4.1 – Пример ввода параметров

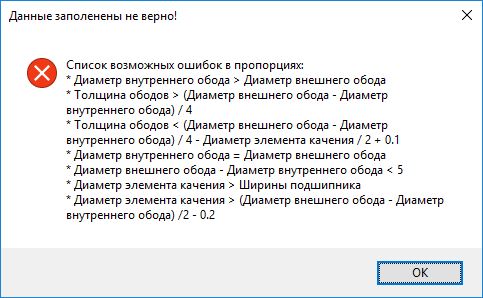


Рисунок 4.2 – Окно ошибки при некорректно введённых данных

В таблице 4.1 представлены некоторые ограничения на размеры параметров подшипника.

Таблица 4.1 – Ограничение на значения некоторых параметров подшипника

|  |  |  |
| --- | --- | --- |
| Параметр | Минимальное значение | максимальное значение |
| Ширина подшипника | 3 | 16 |
| Диаметр внутреннего обода | 3 | 75 |
| Диаметр внешнего обода | 3 | 105 |

После корректного ввода всех значений, нужно нажать на кнопку «Построить деталь», чтобы построить подшипник на рабочей плоскости программы КОМПАС-3D, как показано на рисунке 4.3.

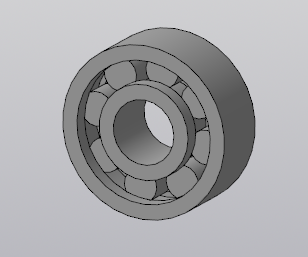


Рисунок 4.3 – Пример построенного подшипника в КОМПАС­-3D

# 5 Проект программы

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценария действий) использован стандарт UML[8].

UML – это 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-моделей возможна генерация кода.

При использовании UML были построены: диаграмма использования и диаграмма классов.

5.1 Диаграмма вариантов использования (Use Cases)

На рисунке 5.1 представлена первоначальная диаграмма вариантов использования.

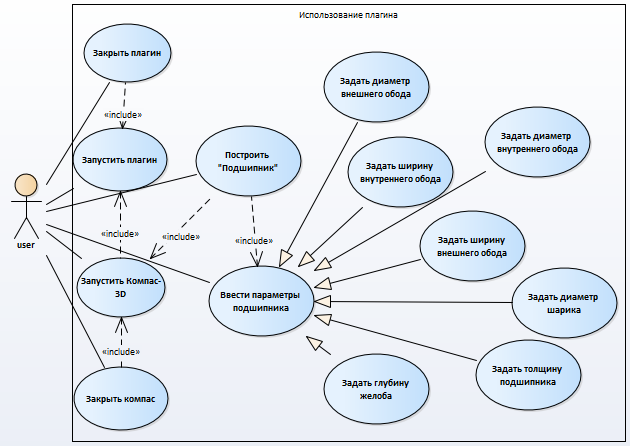


Рисунок 5.1 – Первоначальная диаграмма вариантов использования

В процессе разработки в функциональность плагина были внесены изменения, а именно добавлена возможность выбора формы элемента качения, что отразилось на диаграмме вариантов использования. Измененная диаграмма вариантов использования представлена на рисунке 5.2.

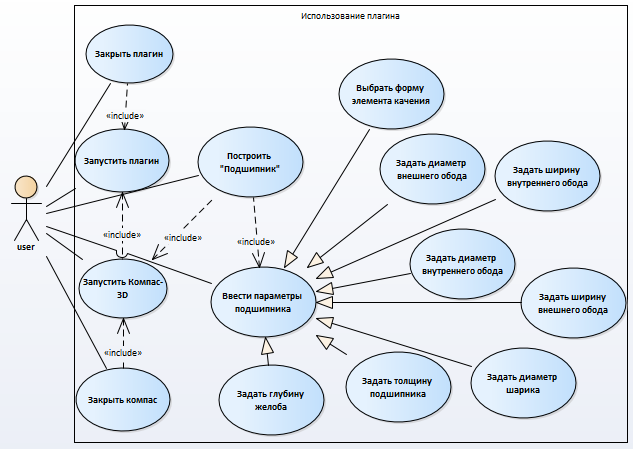


Рисунок 5.2 – Итоговый вид диаграммы вариантов использования

В диаграмму был добавлен случай использования, а именно «Выбрать форму элемента качения».

5.2 Диаграмма классов

На рисунке 5.3 показана изначальная диаграмма классов.

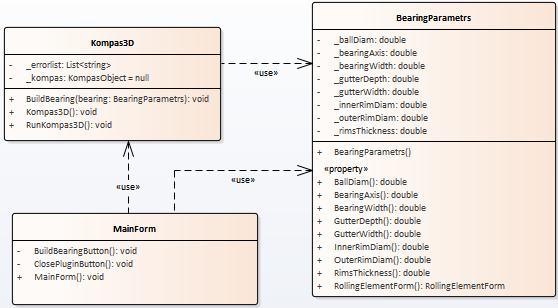


Рисунок 5.3 – Диаграмма классов

После добавления в программу возможности выбора элемента качения, в диаграмму классов добавлен класс перечисления, где содержится список перечислителей, для выбора «Шарика» или «Цилиндра», как элемента качения. Конечный вид диаграммы после изменений представлен на рисунке 5.4.

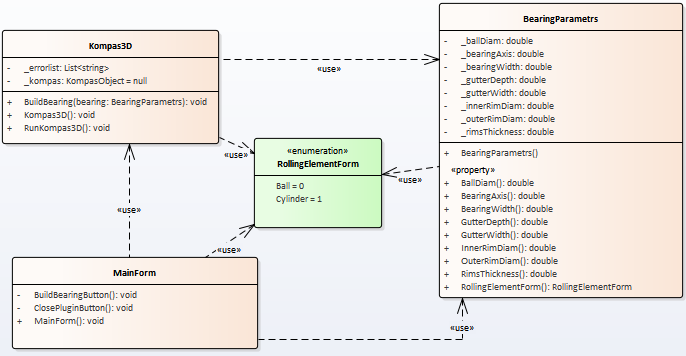


Рисунок 5.4 – Итоговый вид диаграммы классов

5.3 Макет пользовательского интерфейса

Графический интерфейс плагина представляет собой пользовательскую форму с элементами управления значениями параметров объекта. Запуск построения и САПР Компас-3D осуществляется кнопкой «Построить деталь». Первоначальный интерфейс программы представлен на рисунке 5.5.

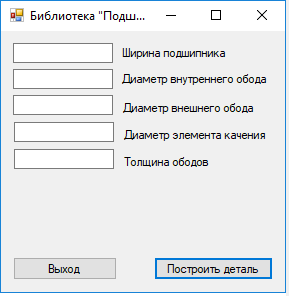


Рисунок 5.5 – Первоначальный пользовательский интерфейс при запуске программы

После добавления возможности выбора элемента качения в плагине пользовательский интерфейс изменился как представлено на рисунке 5.6.

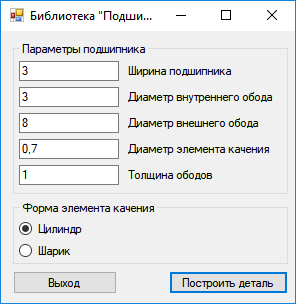


Рисунок 5.6 – Конечный вариант пользовательского интерфейса

# 6 Тестирование программы

Как и было заявлено в ТЗ, плагин тестировался на операционной системе Windows 7 для процессора с разрядностью x32 и Windows 10 для процессора с разрядностью х64. Тестирование на Windows 7 были проведены на виртуальной машине.

На рисунке 6.1 изображены характеристики виртуальной машины с Windows 7 x32.

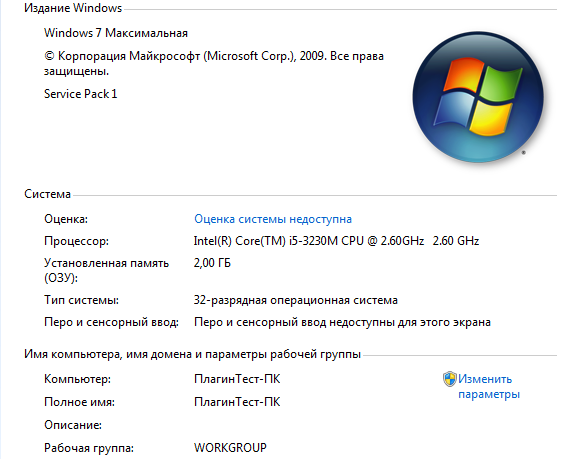


Рисунок 6.1 – Характеристики системы на виртуальной машине

На рисунке 6.2 изображено построение подшипника в КОМПАС-3D на виртуальной машине.

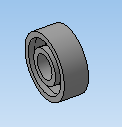


Рисунок 6.2 – Построенная деталь на виртуальной машине с Windows 7 x32

На рисунке 6.3 изображены характеристики компьютера с Windows 10 x64.

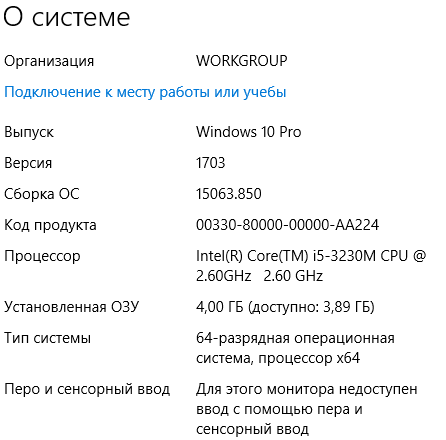


Рисунок 6.3 – Характеристики системы на компьютере

На рисунке 6.4 изображено построение подшипника в КОМПАС-3D на компьютере с Windows 10.

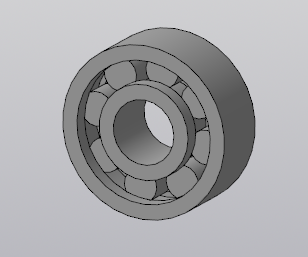


Рисунок 6.4 – Построенная деталь на компьютере с Windows 10 x64

## 6.1 Функциональные тесты

Функциональное тестирование — это тестирование в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения (ПО) в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает. [10]

При запуске плагина открывается окно, показанное на рисунке 6.5.

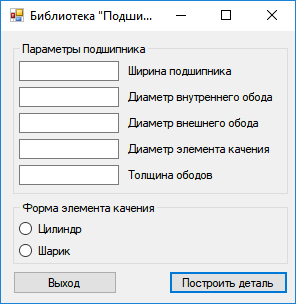


Рисунок 6.5 – Окно плагина

Далее, при попытке ввода неправильных символов в ячейки ввода параметров, а также при пустой ячейке ввода параметров программа выдает ошибку проверки правильности ввода данных (рисунок 6.6).

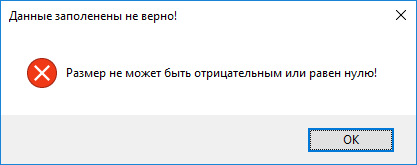


Рисунок 6.6 – Окно ошибки при неправильном вводе данных

Также тестируется ввод некорректных данных (рисунок 6.7).

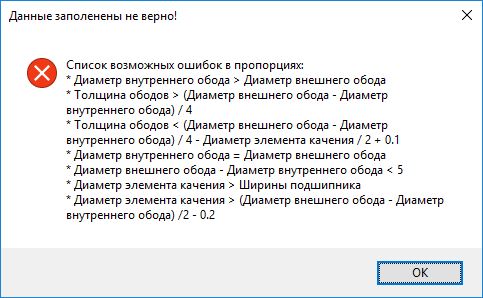


Рисунок 6.7 – Окно ошибки при некорректно введённых данных

Данная ошибка будет выводиться до тех пор, пока пользователь не введет допустимые значения параметров во все ячейки ввода данных (рисунок 6.8).

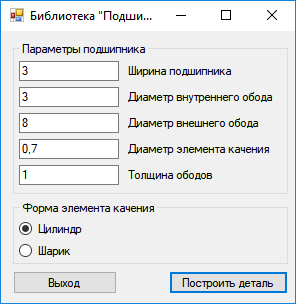


Рисунок 6.8 – Пример допустимых вариантов значений параметров

При вводе корректных данных и нажатии кнопки «Построить деталь», запуститься «КОМПАС-3D» и на рабочей плоскости построится деталь (рисунок 6.9).

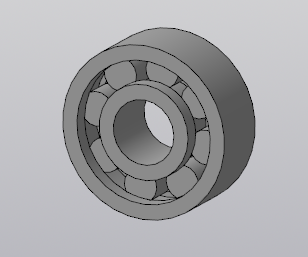


Рисунок 6.9 – Построенная деталь

Ввод минимальных допустимых значений представлен на рисунке 6.10.

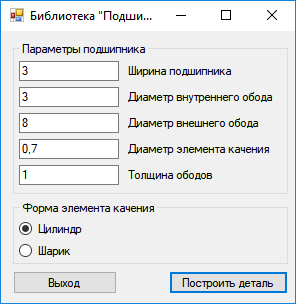


Рисунок 6.10 – Ввод минимально допустимых значений

Результат при вводе минимально допустимых значений представлен на рисунке 6.11.

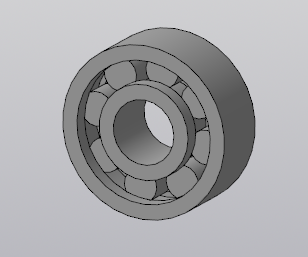


Рисунок 6.11 – Результат ввода минимально допустимых значений

Ввод максимально допустимых значений представлен на рисунке 6.12

.

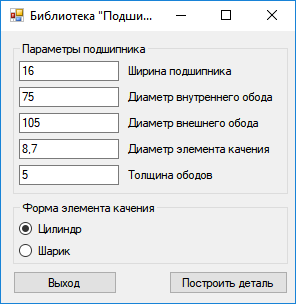


Рисунок 6.12 – Ввод максимально допустимых значений

Результат при вводе максимально допустимых значений показан на рисунке 6.13.

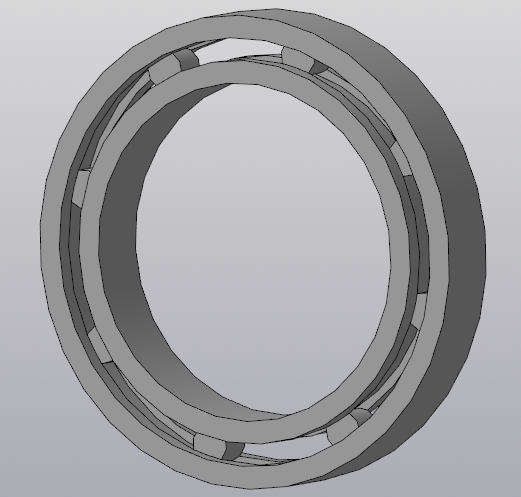


Рисунок 6.13 – Результат ввода максимально допустимых значений

## 6.2 Модульные тесты

В проекте использовалось модульное тестирование. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других элементов, тестируемый элемент должен быть максимально изолирован.

Проверялись только открытые члены класса BearingParametrs, для этого использовались следующие тестовые методы:

– TestBearingParamsRightProps – метод тестирующий создание детали при минимальных, максимальных и усреднённых параметрах;

– TestBearingNegativeParams – метод тестирующий создание детали с неверно заданными пропорциями и отрицательными параметрами. Тест считается пройденным если создание детали завершилось некорректно.

Включая все тестовые случаи, было написано 19 тест-кейсов, подробная таблица с описанием которых представлена в приложении А.

## 6.3 Нагрузочные тесты

Для измерения устойчивости программы было проведено нагрузочное тестирование. Деталь была построена с минимальными параметрами 100 раз. При этом все построенные модели в процессе построения были открыты. Для чистоты эксперимента, чтобы другие процессы минимально влияли на производительность компьютера, программа была запущена на виртуальной машине. На рисунке 6.10 представлен график зависимости времени построения детали от количества построенных деталей.

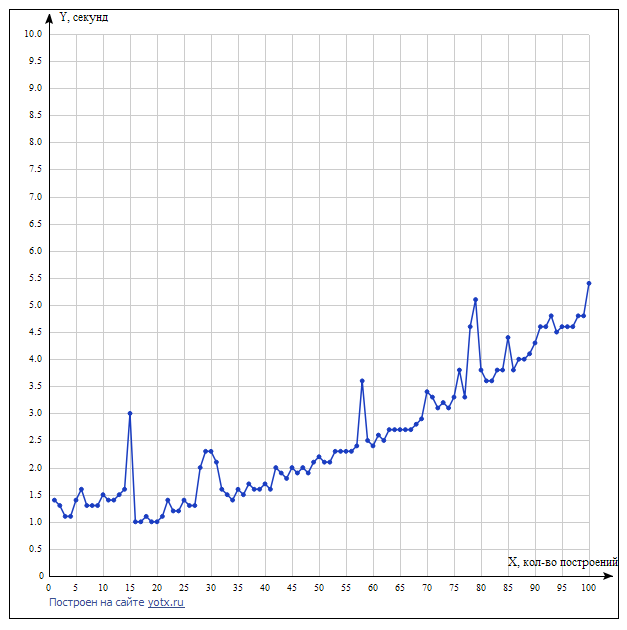


Рисунок 6.14 – График зависимости времени построения детали

от количества построенных деталей

С учетом первого построения детали, которое заняло 54 секунды, общая продолжительность тестирования составила почти 7 минут, среднее время построение одной детали – 3,2 секунды. Из графика видно, с ростом нагрузки на оперативную память время на построение детали повышается.

**Заключение**

В ходе данного проекта были изучены основные этапы проектирования программного продукта, изучена предметная область предмета проектирования, также было изучено API системы автоматизированного проектирования «КОМПАС-3D».

В результате полученные знания были применены для реализации плагина для автоматизации построения модели объекта «Подшипника» в рабочей плоскости программы «КОМПАС-3D».

Реализованный плагин протестирован на платформах Windows 7 и Windows 10. Также проводилось функциональное, нагрузочное и модульное тестирование приложения «КОМПАС-3D».

# Список литературы

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

API – Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения 25.12.2017)

Плагин – Википедия. [Электронный ресурс]. — Режим доступа: <https://ru.wikipedia.org/wiki/Плагин> (дата обращения 25.12.2017)

КОМПАС-3D: О программе. Официальный сайт САПР КОМПАС [Электронный ресурс]. – Режим доступа: <http://kompas.ru/kompas-3d/about/> (дата обращения 25.12.2017)

Кидрук Максим. КОМПАС-3D V10 на 100% / М. Кидрук. – СПб.: Питер, 2009 – 560 с.

Плагин PDF [Электронный ресурс]. — Режим доступа: <http://gkmsoft.ru/allcatalog/pdf2dkompas_plugin/> (дата обращения 25.12.2017)

3D PDF [Электронный ресурс]. — Режим доступа: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения 25.12.2017)

UML. [Электронный ресурс]. – Режим доступа: <http://www.uml.org/> (дата обращения 25.12.2017)

NUnit Documentation – Github [Электронный ресурс] – Режим доступа: <https://github.com/nunit/docs/wiki/NUnit-Documentation> (дата обращения 25.12.2017)

Лайза Криспин, Джанет Грегори. «Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд». Издательство: «Вильямс», 2010. — 464 с.

# Приложение А

Описание тест-кейсов

В таблице А.1 описаны тесты метода TestBearingParamsRightProp

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | Форма элемента качения | Ширина подшипника | Диаметр внутреннего кольца | Диаметр внешнего кольца | Толщина колец | Диаметр элемента качения |
| TestBearingRightProp 1 | 1 | 3 | 3 | 8 | 0,7 | 1,6 |
| TestBearingRightProp 2 | 1 | 7 | 15 | 28 | 2 | 3,18 |
| TestBearingRightProp 3 | 1 | 16 | 75 | 105 | 4 | 8,73 |

Таблица А.1 ­– Тесты на корректные значения

В таблице А.2 описаны тесты метода TestBearingNegativeParams

Таблица А.2 – Тесты на отрицательные значения и неверно заданные пропорции

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | Форма элемента качения | Ширина подшипника | Диаметр внутреннего кольца | Диаметр внешнего кольца | Толщина колец | Диаметр элемента качения |
| TestBearing  NegativeParams 1 | 1 | -3 | 3 | 8 | 0,7 | 1,6 |
| TestBearing  NegativeParams 2 | 1 | 7 | -15 | 28 | 2 | 3,18 |
| TestBearing  NegativeParams 3 | 1 | 16 | 75 | -105 | 4 | 8,73 |
| TestBearing  NegativeParams 4 | 1 | 3 | 3 | 8 | -0,7 | 1,6 |
| TestBearing  NegativeParams 5 | 1 | 7 | 15 | 28 | 2 | -3,18 |
| TestBearingRightProp innerRimDiam > outerRimDiam | 1 | 3 | 8 | 8 | 0,7 | 1,6 |
| TestBearingRightProp innerRimDiam = outerRimDiam | 1 | 7 | 15 | 15 | 2 | 3,18 |
| TestBearingRightProp (outerRimDiam - innerRimDiam)/4 < rimsThickness | 1 | 16 | 75 | 105 | 8 | 8,73 |
| TestBearingRightProp (outerRimDiam - innerRimDiam)/4 - ballDiam / 2 > rimsThickness | 1 | 16 | 75 | 105 | 0,5 | 8,73 |

Окончание таблицы А.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TestBearingRightProp ballDiam > bearingWidth | 1 | 7 | 15 | 28 | 2 | 8 |
| TestBearingRightProp outerRimDiam –  innerRImDiam < 5 | 1 | 3 | 3 | 7 | 0,7 | 1,6 |
| TestBearingRightProp ballDiam > (outerRimDiam - innerRimDiam) / 2 - 0.2 | 1 | 3 | 3 | 8 | 2,5 | 1,6 |
| Enter null parametrs | null | null | null | null | null | null |
| TestBearingNegativeParams double is NaN | 1 | double.NaN | double.NaN | double.NaN | double.NaN | double.NaN |
| TestBearingNegativeParams double is Positive Infinity | 1 | double.  Positive  Infinity | double.  Positive  Infinity | double.  Positive  Infinity | double.  Positive  Infinity | double.  Positive  Infinity |
| TestBearingNegativeParams double is Negative Infinity | 1 | double.  Negative  Infinity | double.  Negative  Infinity | double.  Negative  Infinity | double.  Negative  Infinity | double.  Negative  Infinity |