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)

ПРОЕКТ СИСТЕМЫ

на тему «Разработка программного модуля к САПР КОМПАС 3D для построения подшипника»

по дисциплине «Основы разработки САПР» (ОРСАПР)

Студент гр. 584-2

\_\_\_\_\_\_\_\_ Н. С. Лях

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

дата

Руководитель

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_ А.А. Калентьев

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

дата

# Обзор программы «****КОМПАС 3D**»**

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.

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[1].

## Возможности программы «****КОМПАС 3D»****

Программы данного семейства автоматически генерируют ассоциативные виды трёхмерных моделей (в том числе разрезы, сечения, местные разрезы, местные виды, виды по стрелке, виды с разрывом). Все они ассоциированы с моделью: изменения в модели приводят к изменению изображения на чертеже.

Стандартные виды автоматически строятся в проекционной связи. Данные в основной надписи чертежа (обозначение, наименование, масса) синхронизируются с данными из трёхмерной модели. Имеется возможность связи трёхмерных моделей и чертежей со спецификациями, то есть при «надлежащем» проектировании спецификация может быть получена автоматически; кроме того, изменения в чертеже или модели будут передаваться в спецификацию, и наоборот[1].

## Описание API

API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface, API) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах [2].

Взаимодействие внешнего приложения или подключаемого модуля с системой КОМПАС (с функциями моделирования, математическими функциями ядра системы и пр.) осуществляется посредством программных интерфейсов [3]. В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. В основном, для создания полноценных подключаемых модулей достаточно методов и свойств интерфейсов API 5.

В нижеописанных таблицах представлены методы и объекты, используемые в разработке плагина. В таблицах 1.1 – 1.4 представлены названия, тип и описание методов и свойств используемых интерфейсов.

Таблица 1.1 – Используемые методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| **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 |

Таблица 1. – Используемые методы интерфейса ksEntity

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

Таблица 1.3 – Используемые свойства и методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Create (bool invisible, bool \_typeDoc) | bool | Создать документ-модель (деталь или сборку). Параметры: invisible – признак режима редактирования (видимый, невидимый), typeDoc – тип документа (true – деталь, false – сборка) |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

Таблица 1.4 – Используемые свойства и методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| GetDefaultEntity(short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию. objType – тип объекта |
| NewEntity(short objType) | ksEntity | Создать новый интерфейс объекта и получить  указатель на него. objType – тип объекта |

В таблице 1.5 представлены используемые типы объектов документа-модели.

Таблица 1.5 – Используемые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| **Идентификатор объекта** | **Название объекта** | **Интерфейс параметров** |
| o3d\_planeXOY | Плоскость XOY | ksPlaneParam |
| o3d\_planeXOZ | Плоскость XOZ | 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\_planeOffset | Плоскость с заданным отступом | ksPlaneOffsetDefinition |
| o3d\_bossRotated | Выдавливание вращением | ksBossRotatedDefinition |
| o3d\_axis2Planes | Ось на пересечении плоскостей | ksAxis2PlanesDefinition |
| o3d\_circularCopy | Массив по концентрической сетке | ksCircularCopyDefinition |

**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.

Пример проектируемого изделия приведен на рисунке 1.1.

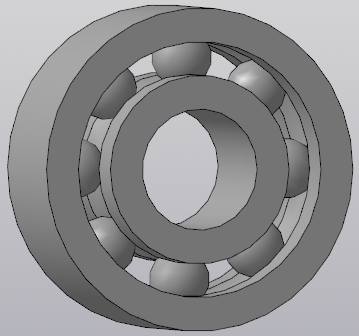


Рисунок 1.1 – Предмет проектирования

# Проектирование приложения

На этапе проектирования приложения составлена диаграмма вариантов использования и диаграмма классов. Основная цель диаграммы вариантов использования – упрощение взаимодействия с будущими пользователями, с клиентами, и особенно пригодится для определения необходимых характеристик системы.

Диаграмма вариантов использования, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне[4].

Диаграмма вариантов использования представлена на рисунке 1.2.

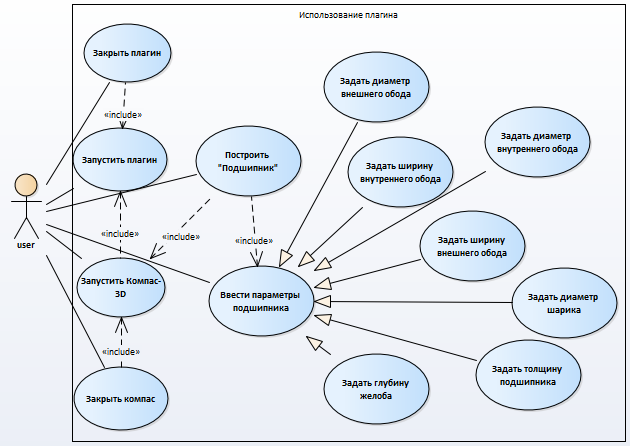


Рисунок 1.2 – Диаграмма вариантов использования «Работа с библиотекой»

UML диаграммы классов представляют собой графическую интерпретацию классов системы, их атрибутов, методов и взаимосвязей между ними. Широко используется при проектировании системы программного продукта, т.к. является наглядным представлением ее организации[4].

Диаграмма классов представлена на рисунке 1.3.

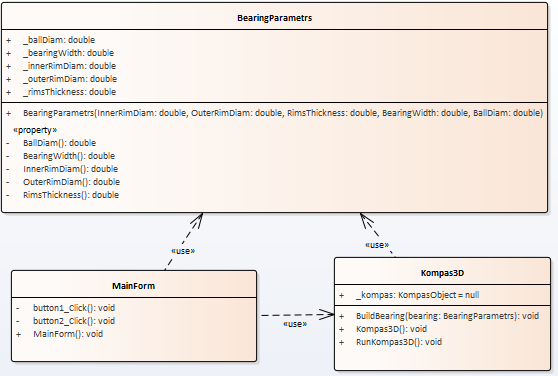


Рисунок 1.3 – Диаграмма классов

Пользовательский интерфейс представлен на рисунке 1.4. В этом окне пользователь может выбрать параметры из вложенного списка, для построения детали нужно нажать на кнопку «Построить деталь».

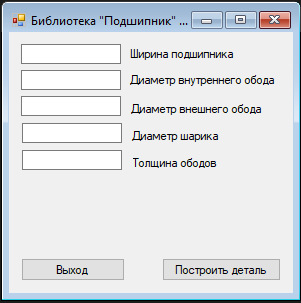


Рисунок 1.4 – Макет пользовательского интерфейса

**Список используемых источников**

1 КОМПАС-3D: О программе [Электронный ресурс]. – Режим доступа: <http://kompas.ru/kompas-3d/about/> (дата обращения: 24.11.2017)

2 API [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/API> (дата обращения: 24.11.2017)

3 Базовые интерфейсы API системы КОМПАС: Максим Кидрук [Электронный ресурс]. – Режим доступа: <http://programming-lang.com/ru/comp_soft/kidruk/1/j190.html> (дата обращения: 24.11.2017)

4 UML. Основы / Фаулер M. – 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.