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)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

на тему: «Проектирование диска для гантели в системе КОМПАС 3D v18»

|  |
| --- |
| Выполнил:  студент гр. 586-2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Е. Дьяченко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г. |
| Руководитель:  к.т.н., доцент каф. КСУП:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Калентьев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020г. |

Томск 2020

**Содержание**

[1 Описание САПР 3](#_Toc527183267)

[1.1 Описание программы 4](#_Toc527183268)

[1.2 Описание API 4](#_Toc527183269)

[1.3 Обзор аналогов 9](#_Toc527183270)

[2 Описание предмета проектирования 10](#_Toc527183271)

[3 Проект программы 10](#_Toc527183272)

[3.1 Диаграмма вариантов использования (Use Cases) 11](#_Toc527183273)

[3.2 Диаграмма классов 13](#_Toc527183274)

[3.3 Макет пользовательского интерфейса 15](#_Toc527183275)

4 Тестирование……………………………………………………………………..17

[4.1 Функциональное тестирование 17](#_Toc527183268)

[4.2 Модульное тестирование 19](#_Toc527183269)

[4.3 Нагрузочное тестирование 21](#_Toc527183270)

5 Заключение ………………………………………………………………...……..24

[Список литературы 25](#_Toc527183276)

# 1 Описание САПР

Система автоматизированного проектирования — автоматизированная

система, реализующая информационную технологию выполнения функций проектирования. Представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования, состоящую из персонала и комплекса технических, программных и других средств автоматизации его деятельности.

Цели создания САПР:

• сокращение трудоёмкости проектирования и планирования;

• сокращение сроков проектирования;

• сокращение себестоимости проектирования и изготовления, уменьшение затрат на эксплуатацию;

• повышение качества и технико-экономического уровня результатов проектирования;

• сокращение затрат на натурное моделирование и испытания.

Задачи САПР:

• автоматизация оформления документации;

• информационная поддержка и автоматизация процесса принятия решений;

• использование технологий параллельного проектирования;

• унификация проектных решений и процессов проектирования;

• повторное использование проектных решений, данных и наработок;

• стратегическое проектирование;

• замена натурных испытаний и макетирования математическим моделированием;

• повышение качества управления проектированием;

• применение методов вариантного проектирования и оптимизации.

# 1.1 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. 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].

# 1.2 Описание API

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. 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 1.1 представлены основные свойства и методы интерфейса KompasObject.

Таблица 1.1 – Методы и свойства интерфейса KompasObject

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип возвращаемых данных | Входные параметры | Описание |
| Document3D() | ksDocument |  | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| GetDynamicArray(long type) | [StructType2D](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/StructType2D.htm) | ext – расширение имени файла,  filter – фильтр поиска (0 – формируется автоматически), prewiew – признакпожключения окна предварительного просмотра: 1 – с подключеине окна, 0 – без подключения окна, typeDir – стартовая папка. | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |
| Visible | bool |  | Свойство видимости приложения |
| Quit() |  |  | Метод для закрытия активного окна приложения КОМПАС |
| GetMathematic2D() | [ksMathematic2D](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksMathematic2D.htm) |  | Метод для получения указателя на интерфейс для работы с математическими функциями |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| GetParamStruct  (short structType) | [StructType2D.](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/StructType2D.htm) | structType – тип интерфейса параметров | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |

В таблице 1.2 представлены методы интерфейса ksEntity, необходимые для разработки плагина.

Таблица 1.2 – Методы интерфейса ksEntity

|  |  |  |
| --- | --- | --- |
| Название | Тип возвращаемых данных | Описание |
| Create() | bool | Создать объект в модели |
| GetDefinition() | IUnkown | Получить указатель на интерфейс параметров объектов и элементов |
| Update() | bool | Изменить свойства объекта (используя ранее установленные свойства) |

В таблице 1.3 представлены свойства и методы интерфейса ksDocument3D, необходимые для разработки плагина.

Таблица 1.3 – Методы интерфейса ksDocument3D

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип возвращаемых данных | Входные параметры | Описание |

Продолжение таблицы 1.3

|  |  |  |  |
| --- | --- | --- | --- |
| Create (bool invisible, bool \_typeDoc) | bool | invisible – признак режима редактирования документа (TRUE – невидимый режим, FALSE – видимый режим),  typeDoc – тип документа (TRUE – деталь, FALSE – сборка). | Создать документ-модель (деталь или сборку) |
| GetPart(int type) | ksPart | type – тип компонента из перечисления Типы компонентов. | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| UpdateDocumentParam() |  |  | Активизировать измененные параметры документа |

В таблице 1.4 представлены методы интерфейса ksPart, необходимые для разработки плагина.

Таблица 1.4 – Свойства и методы интерфейса ksPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип возвращаемых данных | Входные параметры | Описание |
| EntityCollection(short objType) | указатель на интерфейс [ksEntityCollection](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntityCollection.htm) или [IEntityCollection](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntityCollection.htm) | objType – тип объектов, содержащихся в массиве. | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | указатель на интерфейс [ksEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm) или [IEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm) | objType – тип объекта | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |

Продолжение таблицы 1.4

|  |  |  |  |
| --- | --- | --- | --- |
| GetPart(int type) | указатель на интерфейс компонента [ksPart](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksPart.htm) или [IPart](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksPart.htm) | Type – тип компонента | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | указатель на интерфейс [ksEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm) или [IEntity](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/ksEntity.htm). | objType – тип объекта | Создать новый интерфейс объекта и получить указатель на него |

В таблице 1.5 представлены типы объектов документа-модели, необходимые для разработки плагина.

Таблица 1.5 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_unknown | Неизвестный (включает все объекты) |  |
| 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\_face | Грань | [ksFaceDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksEdgeDefinition.htm) |
| o3d\_baseExtrusion | Базовая операция выдавливания | [ksBaseExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutExtrusion | Вырезать выдавливанием | [ksCutExtrusionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |
| o3d\_baseLoft | Создание элемента по сечениям | [ksBaseLoftDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_baseEvolution | Создание кинематического элемента | [ksBaseEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksBaseExtrusionDefinition.htm) |
| o3d\_cutEvolution | Вырезать кинематический элемент | [ksCutEvolutionDefinition](mk:@MSITStore:C:\Program%20Files\ASCON\KOMPAS-3D%20V16\SDK\SDK.chm::/ksCutExtrusionDefinition.htm) |

# 

# 1.3 Обзор аналогов

**Плагин PDF**

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF [3]. 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 Описание предмета проектирования

Предметом проектирования является диск для гантели. Диск для гантели – это съёмная часть гантели, которая позволяет менять вес снаряда для регулировки создаваемой нагрузки на использующего его спортсмена. В основном изготавливается из [стали](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BB%D1%8C).

Параметры диска гантели:

* Внешний диаметр D (от 137 до 251 мм);
* Внутренний диаметр B (от 26 до 31 мм);
* Центральный вырез C (от 80 до E мм);
* Глубина центрального выреза F (от 5 до 10 мм);
* Толщина H (от 26 до 69 мм);
* Скругление внешних рёбер кромки L (от 1 до 10 градусов);
* Скругление внутренних рёбер кромки M (от 1 до 10 градусов);
* Толщина кромки (E) должна быть как минимум меньше 30 мм от внешнего диаметра (D) включительно, (E <= D – 30).
* Глубина выреза внутреннего диаметра (A) равна толщине диска (H),

(A = H).

* Скругления внутренних (M) и внешних (L) рёбер равны (M = L).

На рисунке 2.1 представлена 3D модель диска.

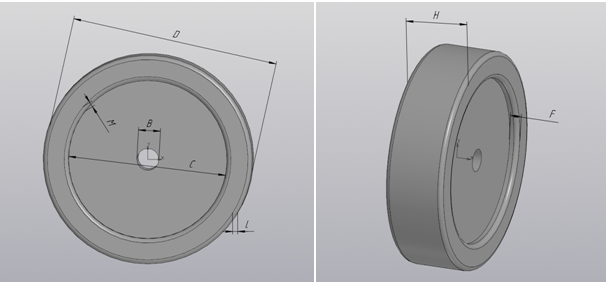


Рисунок 2.1 – 3D модель диска.

# 3 Проект программы

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

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 [4].

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

# 3.1 Диаграмма вариантов использования (Use Cases)

Вариант использования специфицирует это ожидаемое поведение субъекта (системы или её части), — он описывает последовательности действий, включая их варианты, которые субъект осуществляет для достижения действующим лицом определённого результата [5]. Диаграмма вариантов использования представлена на рисунке 3.1

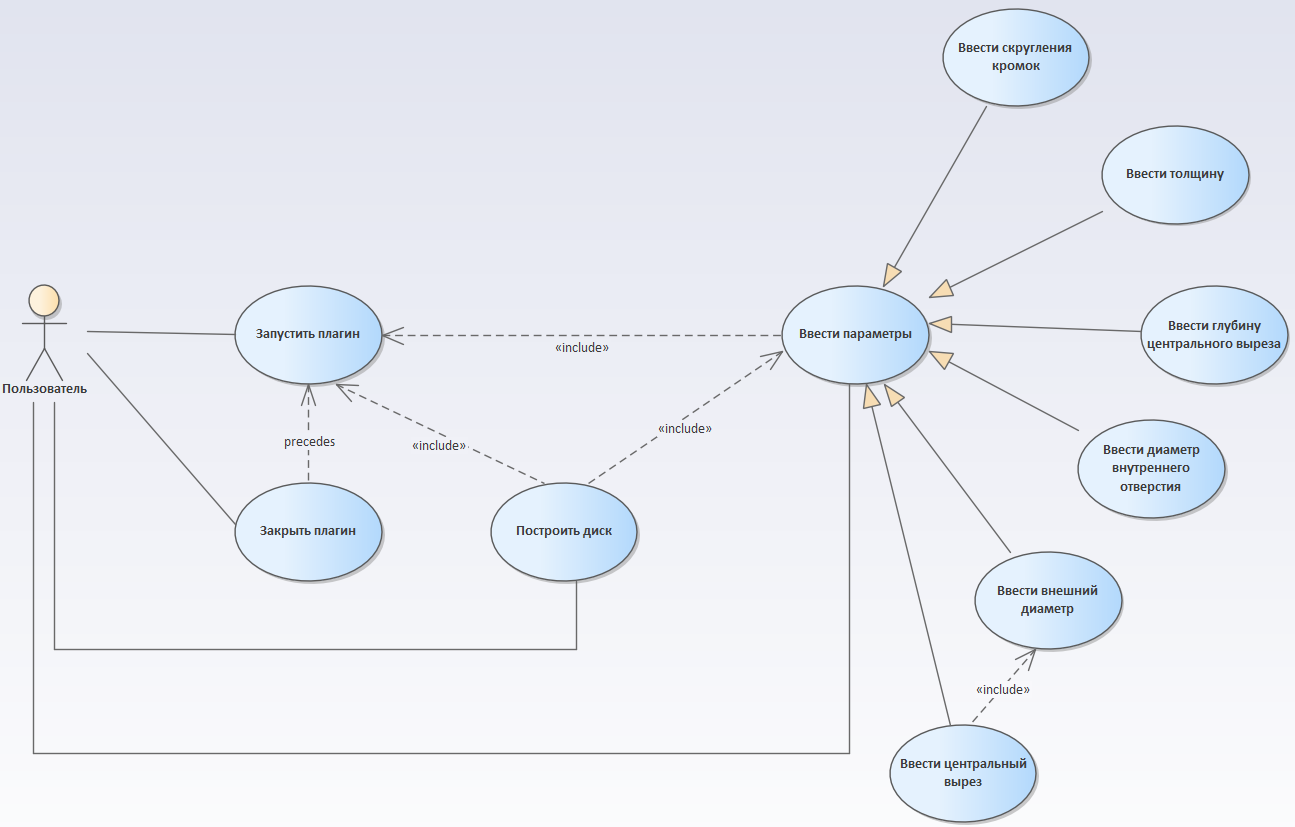


Рисунок 3.1 – Диаграмма вариантов использования.

После добавления дополнительной функциональности схема Use Case поменяла свой вид. Её можно увидеть на рисунке 3.2.

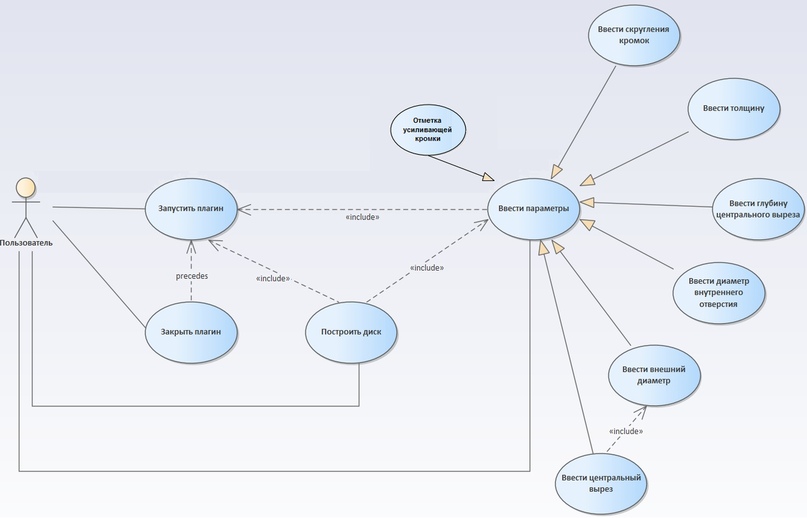


Рисунок 3.2 – Use Case диаграмма проекта после добавления дополнительной функциональности.

# 3.2 Диаграмма классов

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами [3].

На рисунке 3.3. показана диаграмма классов

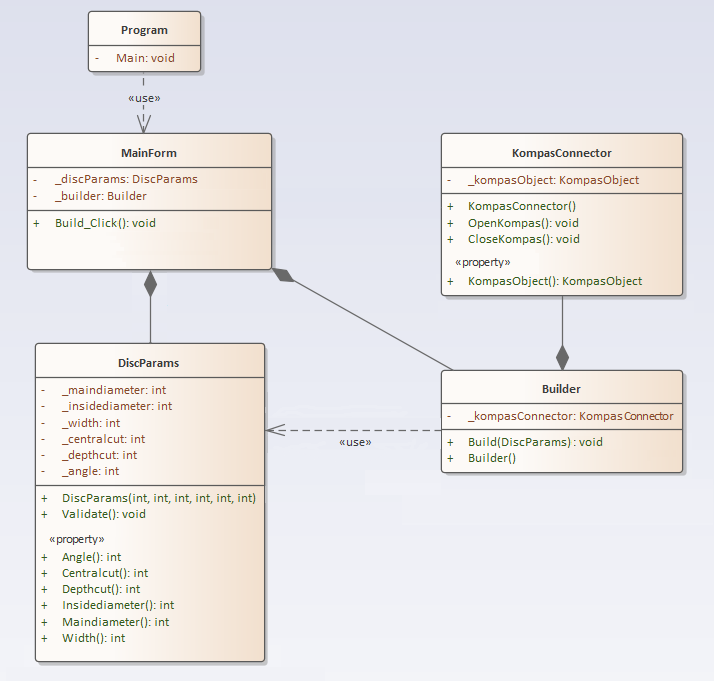


Рисунок 3.3 – Диаграмма классов.

Для реализации подсистемы были спроектированы следующие классы:

* MainForm – класс диалогового окна, обеспечивающий взаимодействие между пользователем и программой через форму;
* DiscParams − класс, хранящий в себе все параметры диска, осуществляет проверку зависимых параметров;
* KompasConnector – класс, отвечающий за работу с API КОМПАС 3D.
* Builder – класс, отвечающий за вызов методов API КОМПАС 3D, необходимых для постройки объекта проектирования.

Диаграмма, полученная после этапа реализации, изображена на рисунке 3.4

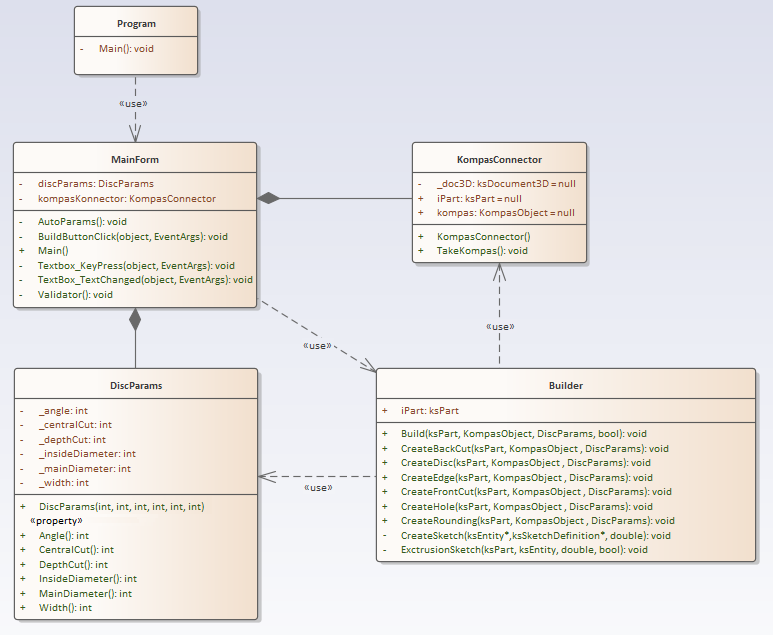


Рисунок 3.4 – Диаграмма классов после этапа реализации.

Для реализации подсистемы были спроектированы следующие классы:

* MainForm – класс диалогового окна, обеспечивающий взаимодействие между пользователем и программой через форму;
* DiscParams − класс, хранящий в себе все параметры модели диска, осуществляет проверку зависимых параметров;
* KompasConnector – класс, отвечающий за работу с API КОМПАС 3D;
* Builder – класс, отвечающий за вызов методов API КОМПАС 3D, необходимых для постройки объекта проектирования.

**3.3 Макет пользовательского интерфейса**

Плагин представляет собой пользовательскую форму с ячейками для ввода параметров.

В блок “Ввод параметров” (обведен рамкой под номером 1) пользователю необходимо ввести желаемые параметры для 3D-модели.

Кнопка “Построить” (обведена рамкой под номером 2) при нажатии на которую произойдёт построение 3D-модели по параметрам, ведённым в поле блока ввода.

Поля ввода, в которых были введены неправильные параметры, будут подсвечиваться красный цветом (отмечены знаком «–»), а также появится предупредительное сообщение (обведено рамкой под номером 3). Если данные в поле введены верно, то оно подсветится зелёным цветом (отмечены знаком «+»). Согласно дополнительному заданию размещен блок для отметки построения усиливающей кромки (отмечено рамкой под номером 4). На рисунке 3.5 представлен макет интерфейса плагина.

Для построения модели «Диск гантели» необходимо:

* Ввести параметры диска;
* При необходимости выбрать построение с усиливающей кромкой;
* Запустить построение модели (нажать кнопку «Построить»).

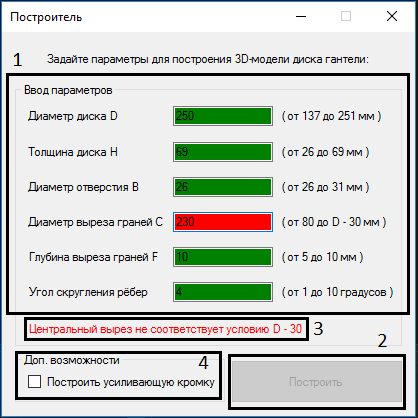


Рисунок 3.5 – Макет пользовательского интерфейса при запуске программы.

# 4 Тестирование

# 4.1 Функциональное тестирование

Функциональное тестирование — тестирование функциональности объекта, т.е. правильно ли объект выполняет свои функции. Фактически, выполняется проверка правильности выходных данных при соответствующих входных [3]. Проведем тестирование при максимальных и минимальных параметрах модели.

Модель с минимальными входными параметрами представлена на рисунке 4.1.

# 

Рисунок 4.1 — Модель диска с минимальными входными параметрами.

Модель с максимальными входными параметрами представлена на рисунке 4.2.

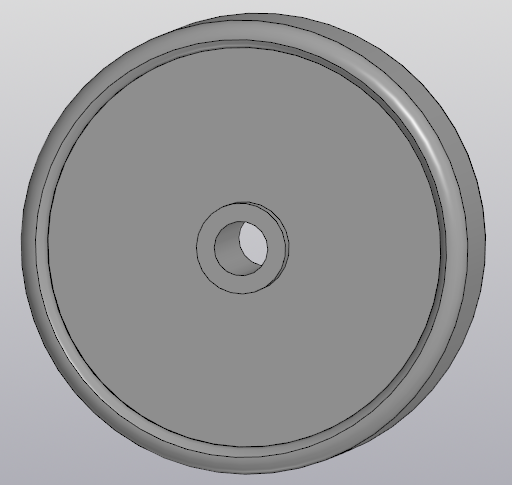


Рисунок 4.2 — Модель диска с максимальными входными параметрами и усиливающей кромкой.

При выходе из зоны допустимых значений параметров, поле подсвечивается красным цветом, появляется сообщение о том, что введены некорректные данные, а также кнопка «Построить» становится недоступна.

Пример приведен на рисунке 4.3. Рамкой выделено сообщение об ошибке и поле, в котором значение задано некорректно.

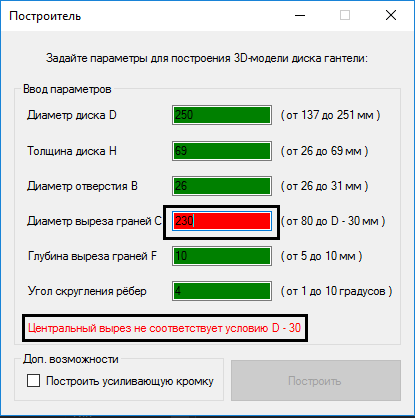


Рисунок 4.3 – Ошибка при вводе некорректных параметров.

# 4.2 Модульное тестирование

Юнит-тестирование (блочное тестирование, «unit-testing») —тестирование отдельного элемента изолированно от остальной системы [3].

Список тестовых сценариев для модульного тестирования граничных значений входных параметров представлен в таблице 4.1.

Таблица 4.1 – Список тестовых сценариев.

|  |  |
| --- | --- |
| TestMainDiameterGet\_CorrectValue() | Позитивный тест проверки диаметра диска |
| TestWidthGet\_CorrectValue() | Позитивный тест проверки толщины диска |
| TestInsideDiameterGet\_CorrectValue() | Позитивный тест проверки диаметра отверстия диска |
| TestCentralCutGet\_CorrectValue() | Позитивный тест проверки диаметра выреза граней диска |
| TestDepthCutGet\_CorrectValue() | Позитивный тест проверки глубины выреза граней диска |
| TestAngleGet\_CorrectValue() | Позитивный тест проверки угла скругления рёбер диска |
| TestBorderMainDiameterSet\_ArgumentException (int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – диаметр диска |

Продолжение таблицы 4.1.

|  |  |
| --- | --- |
| TestBorderWidthSet\_ArgumentException(int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – толщина диска |
| TestBorderInsideDiameterSet\_ArgumentException(int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – диаметр отверстия диска |
| TestBorderCentralCutSet\_ArgumentException(int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – диаметр выреза граней диска |
| TestBorderDepthCutSet\_ArgumentException(int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – глубина выреза граней диска |
| TestBorderAngleSet\_ArgumentException(int wrongParam, string message) | Тестирование при вводе диапазона некорректных значений – угол скругления рёбер диска |

Результаты успешного прохождения всех модульных тестов приведены на рисунке 4.4.

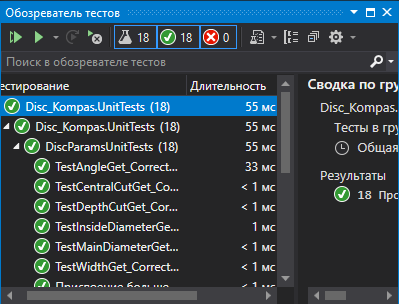


Рисунок 4.4 - Результаты модульных тестов.

# 4.3 Нагрузочное тестирование

Нагрузочное тестирование — подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству) [4].

Нагрузочное тестирование будет проводится на вычислительной машине с 8Гб ОЗУ и операционной системой Windows 10.

После построения 221 детали с минимальными параметрами программа завершилась со сбоем из-за нехватки памяти. На момент сбоя количество потребляемой памяти было около 730 МБ.

После построения 172 детали с максимальными параметрами программа также завершилась со сбоем. Но момент сбоя количество потребляемой памяти было около 886 МБ.

На представленных графиках в текущей главе ось «X» – время в минутах, ось «Y» – количество построенных деталей.

На рисунке 4.5 представлено тестирование зацикленного построения фигуры с минимальными параметрами:

* Диаметр диска 137;
* Толщина диска 26;
* Диаметр отверстия 26;
* Диаметр выреза граней 80;
* Глубина выреза граней 5;
* Угол скругления рёбер 15;

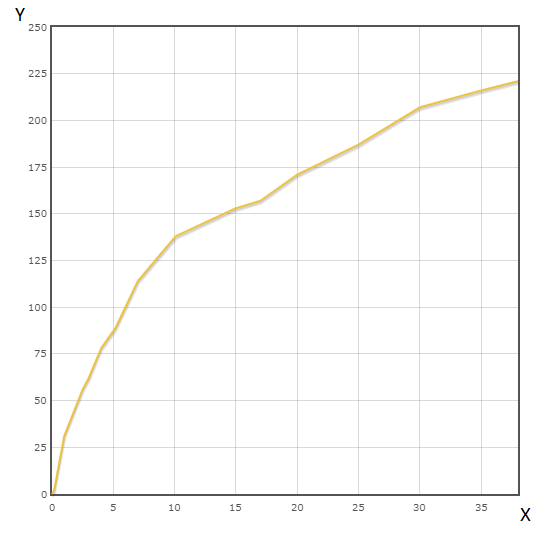


Рисунок 4.5 – График зависимости времени от количества построенных фигур с минимальными параметрами.

На рисунке 4.6 представлено тестирование зацикленного построения фигуры с максимальными параметрами:

* Диаметр диска 251;
* Толщина диска 69;
* Диаметр отверстия 31;
* Диаметр выреза граней 221;
* Глубина выреза граней 10;
* Угол скругления рёбер 10;

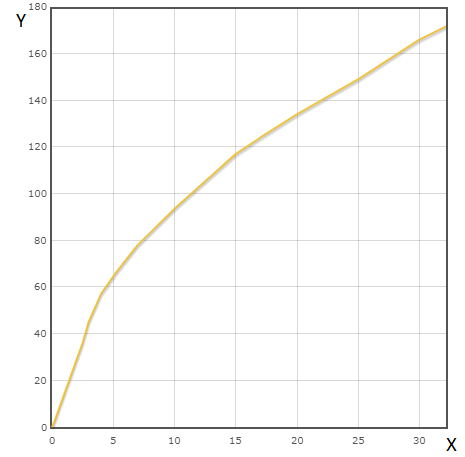


Рисунок 4.6 – График зависимости времени от количества построенных фигур с максимальными параметрами.

По графикам можно определить, что при увеличении количества деталей, постепенно спадает производительность.

# Заключение

При выполнения лабораторных работ были изучены основные этапы проектирования программного продукта, проведена работа с API САПР «КОМПАС-3D».

Было составлено техническое задание, разработан проект системы, составлены UML диаграммы классов и вариантов использования, разработан макет пользовательского интерфейса.

В результате работы был разработан и реализован плагин для САПР «КОМПАС-3D», который производит построение 3D модели диска гантели по заданным параметрам.

Реализованный плагин протестирован функциональным, модульным и нагрузочным тестированием на платформах Windows 10 SP1 «x86» и «x64».

# Список литературы

Система трёхмерного моделирования компании Компас-3D [Электронный ресурс]. — Режим доступа: <https://ascon.ru/products/7/review/> (дата обращения 20.03.2020).

Плагин PDF [Электронный ресурс]. — Режим доступа: <http://gkmsoft.ru/ru/#/pdf3d_kompas> (дата обращения 20.03.2020).

Новые технологии в программировании: учебное пособие / А.А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. – Томск, 2014. − ­ 176 стр.

Виды тестирования [электронный ресурс]. – режим доступа: <https://qa-academy.by/qaacademy/news/klassifikaciya-vidov-testirovaniya/> (дата обращения: 02.03.2020)