Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Выполнил:   
Студент гр. 586-2

Селиванов П.А.

Проверил: к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ Калентьев А.А.

« » 2020 г.

Томск 2020

**Реферат**

Пояснительная записка 26 страницы, 14 иллюстраций, 5 таблиц, 5 использованных источников.

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

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

Цель работы – создание плагина для построения трехмерной детали (короб сабвуфера) по введенным параметрам в «КОМПАС-3D».

В процессе выполнения курсового проекта разработан плагин для построения шестерни в рабочей плоскости САПР КОМПАС-3D, а также проектная документация к ней.

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

Содержание

[**Введение** 4](#_Toc40099436)

[**1 Описание САПР** 5](#_Toc40099437)

[**1.1 Описание программы** 5](#_Toc40099438)

[**1.2. Описание API** 6](#_Toc40099439)

[**1.3 Обзор аналогов** 9](#_Toc40099440)

[**1.3.1 Сайт subbox.pro** 9](#_Toc40099441)

[**1.3.2 Программа Bassbox 6 Pro** 9](#_Toc40099442)

[**2. Описание предмета проектирования** 11](#_Toc40099443)

[**3. Проект программы** 13](#_Toc40099444)

[**3.1 Диаграмма USECASE** 13](#_Toc40099445)

[**3.2 Диаграммы классов** 13](#_Toc40099446)

[**3.3 Макет пользовательского интерфейса.** 15](#_Toc40099447)

[**4 Тестирование** 17](#_Toc40099448)

[**4.1 Функциональное тестирование** 17](#_Toc40099449)

[**4.2 Модульное тестирование** 18](#_Toc40099450)

[**4.4 Нагрузочные тесты** 20](#_Toc40099451)

[**5 Заключение** 22](#_Toc40099452)

[**Список используемых источников** 23](#_Toc40099453)

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

## **1 Описание САПР**

## **1.1 Описание программы**

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей (в том числе, деталей, формируемых из листового материала путём его гибки) и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология позволяет быстро получать модели типовых изделий на основе проектированного ранее прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства [1].

Система «Компас-3D» включает следующие компоненты: система трёхмерного твердотельного моделирования, универсальная система автоматизированного проектирования «Компас-График» и модуль формирования спецификаций.

Ключевой особенностью продукта является обеспечение сквозного процесса проектирования от реализации идеи в 3D до подготовки полного комплекта документации. В основе КОМПАС-3D лежат собственное математическое ядро и параметрические технологии, разработанные специалистами АСКОН. Продукт содержит инструменты для коллективного проектирования изделий и объектов строительного проектирования любой степени сложности и позволяет подготовить полноценную электронную модель изделия, здания и сооружения.

Базовая функциональность продукта легко расширяется за счёт различных приложений, дополняющих функционал КОМПАС-3D эффективным инструментарием для решения прикладных инженерных задач. Например, приложения для проектирования трубопроводов, металлоконструкций, различных деталей машин позволяют большую часть действий выполнять автоматически, сокращая общее время разработки проекта в несколько раз.

## **1.2. Описание API**

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс можно с помощью экспортной функции CreateKompasObject(). Методы этого интерфейса, главные из которых представлены в таблицах 1.1-1.4, 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).

Таблица 1.1. Методы интерфейса KompasObject.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| Document3D() |  | |  |  | | --- | --- | |  | Указатель на интерфейс до­кумента трех­мерной моде­ли ksDocument3D. | | Дает возможность получить указатель на интерфейс трехмерного документа(детали или сборки) |
| GetDynamicArray(long type) | |  |  | | --- | --- | | ext | - расширение имени файла, | | filter | - фильтр пои­ска (0 - фильтр фор­мируется авто­матически), | | preview | - признак под­ключения окна предваритель­ного просмо­тра:  1 - с подклю­чением окна, 0 - без под­ключения ок­на, | | typeDir | - стартовая папка. | | строка с име­нем файла | Возвращает указатель на интерфейс динамического массива |

Окончание таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Visible |  |  | Свойство видимости приложения |
| GetMathematic2D() |  | указатель на интерфейс [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 |
| GetParamStruct(short structType) | |  |  | | --- | --- | | structType | [- тип интерфейса параметров](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/StructType2D.htm) | | указатель на интерфейс указанного ти­па из [StructType2D.](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/StructType2D.htm) | Метод для получения указателя на интерфейс графического документа (чертежа или фрагмента) |

Таблица 1.2. Методы интерфейса ksDocument3D.

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Входные параметры | Возвращаемое значение | Описание |
| Create (bool invisible, bool \_typeDoc) | |  |  | | --- | --- | | invisible | - признак ре­жима редакти­рования доку­мента  (TRUE - неви­димый режим,  FALSE - види­мый режим), | | typeDoc | - тип докумен­та  (TRUE - де­таль,  FALSE - сбор­ка). | | |  |  | | --- | --- | | TRUE | - в случае успешного за­вершения. | | Дает возможность создать пустой документ (деталь или сборку) |
| UpdateDocumentParam() |  | |  |  | | --- | --- | | TRUE | - в случае успешного за­вершения. | | Активизировать измененные параметры документа |
| GetPart(int type) | |  |  | | --- | --- | | type | - тип компо­нента из пере­числения [Типы компонентов](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/PartType.htm). | |  | Получить указатель на интерфейс компонента в соответствии с заданным типом |

Таблица 1.3 – Методы интерфейса IPart.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Возвращаемое значение | Описание |
| EntityCollection  (short objType) | |  |  | | --- | --- | | objType | - тип объектов, содержащихся в массиве. | | 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) | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity  (short objType) | |  |  | | --- | --- | | objType | - тип объекта. | | |  | | --- | | 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). |   Типы объектов (objType):   |  |  |  | | --- | --- | --- | | o3d\_planeXOY | 1 | - плоскость XOY | | o3d\_planeXOZ | 2 | - плоскость XOZ | | o3d\_planeYOZ | 3 | - плоскость YOZ | | o3d\_pointCS | 4 | - точка начала системы координат | | o3d\_axisOX | 71 | - ось OX | | o3d\_axisOY | 72 | - ось OY | | o3d\_axisOZ | 73 | - ось OZ | | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | |  |  | | --- | --- | | type | - тип компонента. | | 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). | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | |  |  | | --- | --- | | objType | - [тип объекта](mk:@MSITStore:D:\INSTAL\KOMPAS-3D%20V17.1\KOMPAS\SDK\SDK.chm::/Obj3dType_NewEntil_Part.htm). | | 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). | Создать новый интерфейс объекта и получить указатель на него |

## **1.3 Обзор аналогов**

В наше время существует множество различных аналогов построения 3D модели сабвуфера с различными параметрами. В данных программах присутствует множество других модулей, такие как построение чертежей, различных графиков, портов и сабвуфера. В связи с этим, библиотека для построения короба сабвуфера в КОМПАС 3D будет разработана для учебных целей.

## **1.3.1 Сайт subbox.pro**

Существует веб-сайт [2] по построению и отрисовки 3D модели короба сабвуфера по заданным параметрам. Также, помимо построения 3D модели, на данном сайте присутствует возможность отрисовки чертежа. Интерфейс сайта приведен на рисунке 1.1

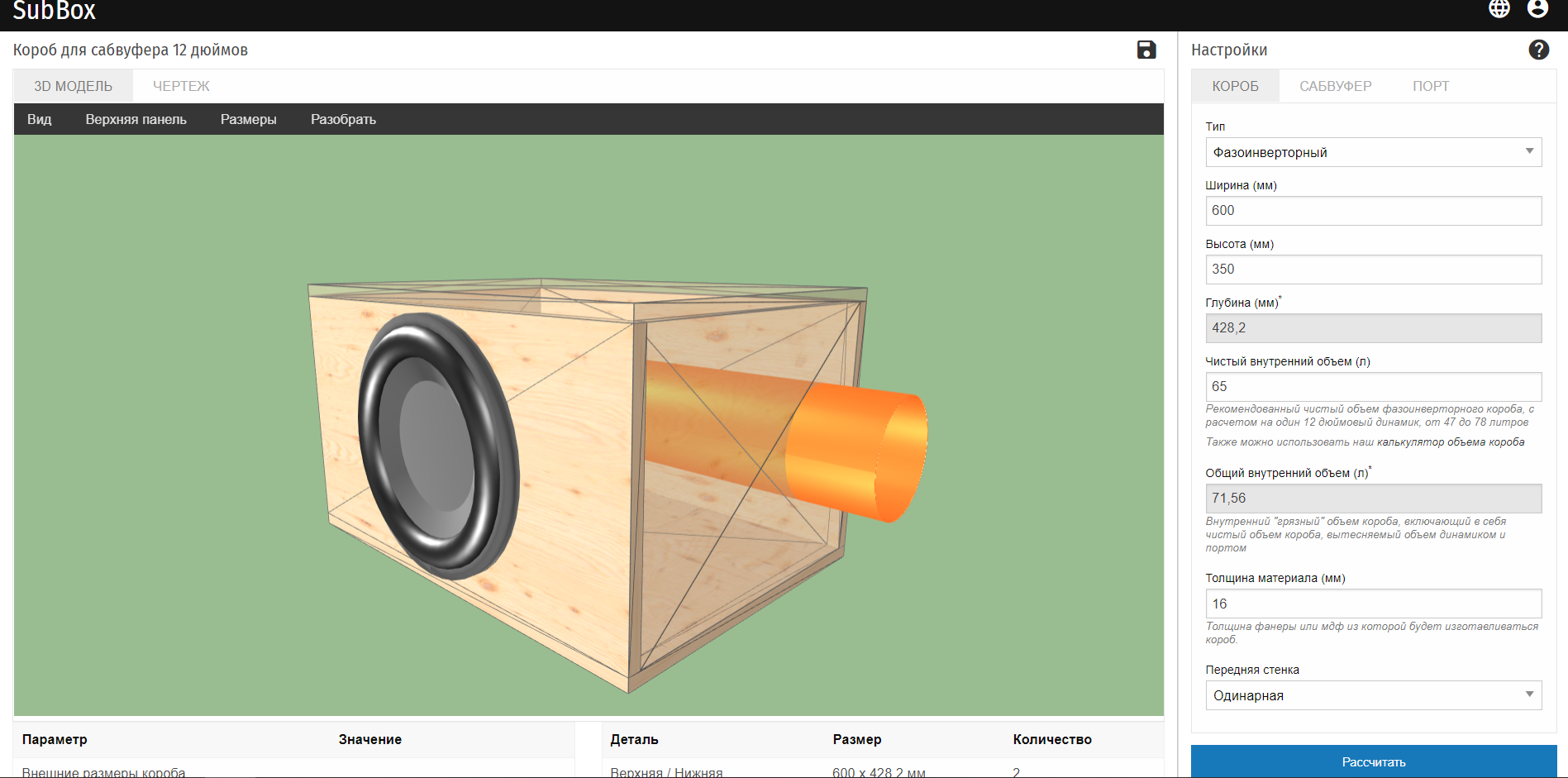


Рисунок 1.1 — Сайт subbox.pro

## **1.3.2 Программа Bassbox 6 Pro**

BassBox 6 Pro — это программа для расчета акустического оформления для сабвуфера(закрытый ящик, фазоивертор, бандпасс)[3]. Имеет большую базу готовых параметров для динамиков большинства известных производителей. Данная программа ведет расчеты на основе [параметров Тиля-Смолла](https://doctorbass.ru/baza-znaniy/parametry-tilya-smolla-fs-qts-vas/), строит графики АЧХ и учитывает передаточную частоту салона. Интерфейс данной программы приведен на рисунке 1.2.

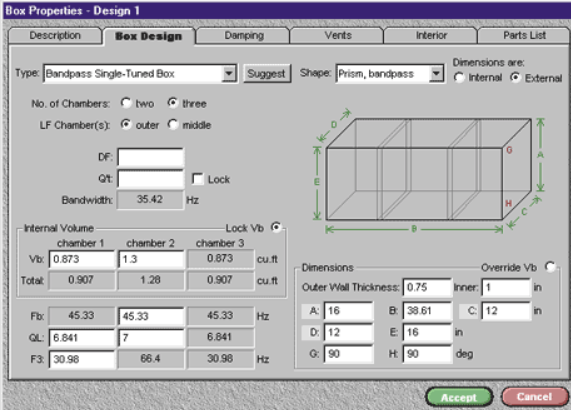


Рисунок 1.2 — Программа BassBox 6 Pro.

## **2. Описание предмета проектирования**

Предметом проектирования является короб сабвуфера. Короб – это корпус, в который устанавливаются динамики для воспроизведения низких частот.

На этапе разработки и построения модели в KOMPAS 3D было выявлено, что диапазон максимальных и минимальных значений параметров в техническом задании были заданы некорректно.  
Предыдущие параметры короба сабвуфера.

* Длина короба L: от Lmin = Nc\*Dc+2S+20 мм до Lmax = 2.4Dc+2S+20 мм;
* Ширина короба: W = 2Dп+2S мм;
* Высота короба: H = Dc+2S мм;
* Количество отверстий для сабвуфера Nс: от одного до двух(на передней панели);
* Диаметр отверстия сабвуфера Dс: от 200 мм до 480 мм;
* Диаметр отверстия порта(трубы) Dп: от 50 мм до 150 мм;
* Толщина материала S: от 20 мм до 40 мм.

Текущие параметры короба сабвуфера:

* Длина короба L: от Lmin = Nc\*Dc + 2S см до Lmax = 106 см;
* Ширина короба W: от Wmin = Dс + 2S см до Wmax = L см;
* Высота короба H: от Hmin = Dс + 2S см до Hmax = 55 см;
* Количество отверстий для сабвуфера Nс: от одного до двух(на передней панели);
* Диаметр отверстия сабвуфера Dс: от 15 см до 45 см;
* Диаметр отверстия порта(трубы) Dп: от 4 см до Dс/2 см;
* Толщина материала S: от 2 см до 4 см.

Изображение предмета проектирования с обозначенными параметрами приведено на рисунке 2.1

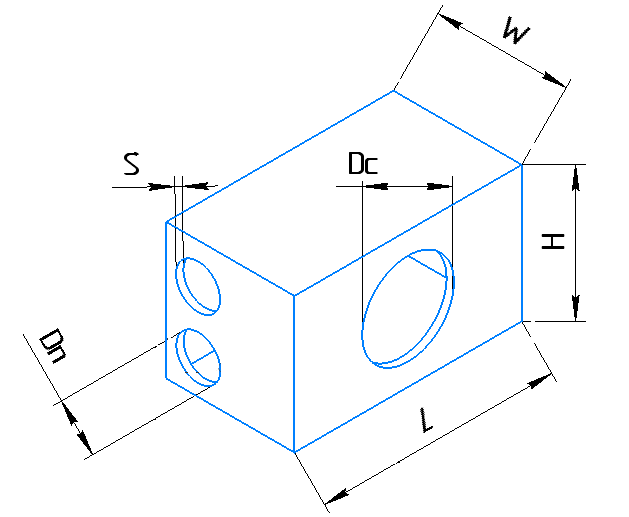


Рисунок 2.1 —­­ Геометрические параметры короба сабвуфера

В связи со сложностью постройки корпуса сабвуфера с щелевым портом, для моделирования будет использоваться корпус с круглым портом. Значения параметров предмета проектирования взяты с открытого источника [5] и будут использованы в процессе моделирования в качестве опорных. Данные значения приведены в таблице 2.1.

Таблица 2.1 – Параметры и размены корпуса для сабвуфера.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Калибр, дюйм | Длинна, см | Ширина, см | Высота, см | Диаметр трубы, см | Длина порта, см |
| 6,5 | 45 | 32,6 | 20 | 8 | 60 |
| 8 | 50 | 36,1 | 27 | 10 | 60 |
| 10 | 65 | 37,4 | 30 | 12,5 | 60 |
| 12 | 80 | 38,2 | 36 | 15 | 60 |
| 15 | 85 | 44,3 | 46 | 20 | 60 |
| 18 | 90 | 47,2 | 60 | 22 | 60 |
| 2 x 8 | 85 | 37,1 | 27 | 10 | 48 |
| 2 x 10 | 85 | 43,2 | 35 | 12,5 | 46 |
| 2 x 12 | 90 | 42,3 | 55 | 15 | 44 |

## **3. Проект программы**

## **3.1 Диаграмма USECASE**

Вариант использования (ВИ) 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 [4]. Диаграмма вариантов использования представлена на рисунке 3.1

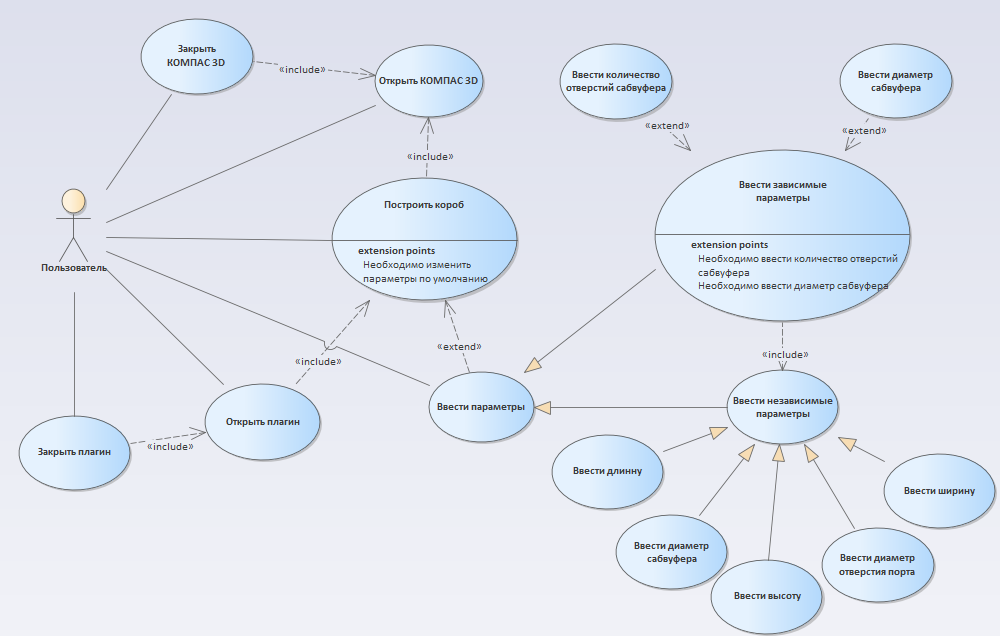


Рисунок 3.1 — Диаграмма вариантов использования

Однако, в связи с тем, порядок ввода зависимых параметров оказался не важным, в процессе разработки диаграмма вариантов использования была изменена. Новая диаграмма представлена на рисунке 3.2

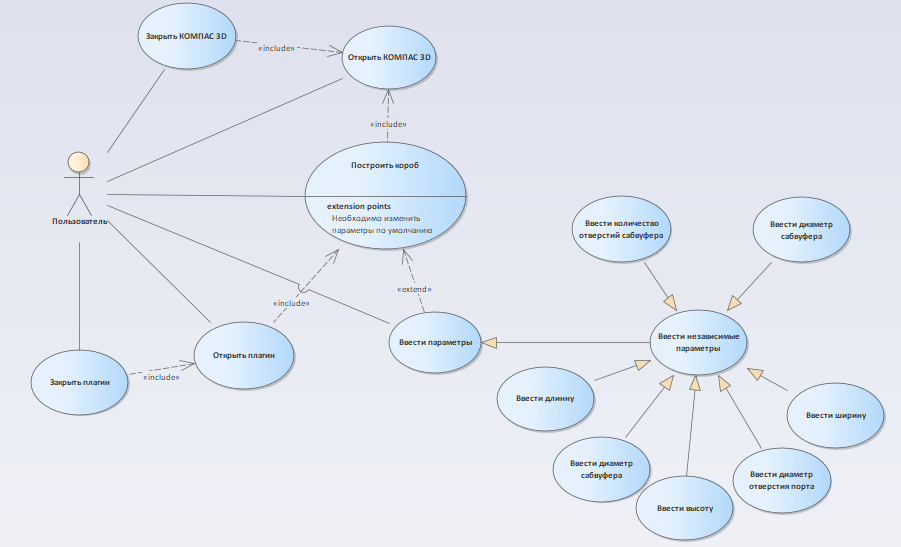
****

Рисунок 3.2 — Финальная версия диаграммы вариантов использования

## **3.2 Диаграммы классов**

Диаграмма классов — один из видов UML-диаграмм, позволяющий описать статический аспект программной системы за счёт описания классов и их взаимосвязей в системе. Изначальная диаграмма классов представлена на рисунке 3.2.

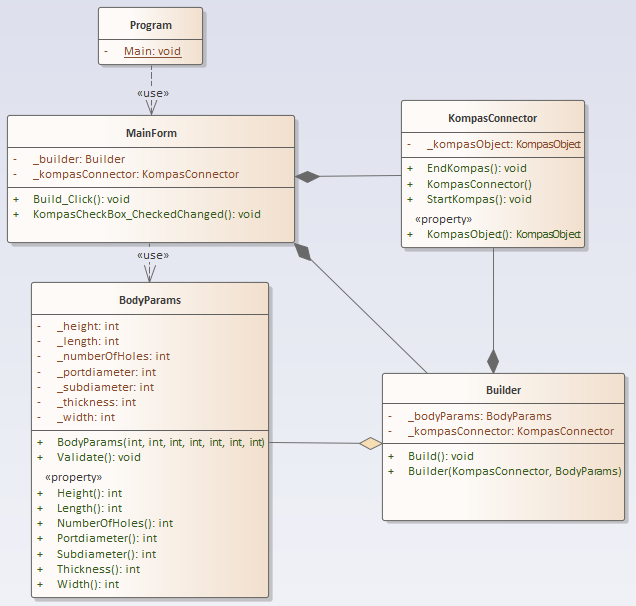
****

Рисунок 3.3 — Изначальная диаграмма вариантов использования

Поля и методы диаграммы, представленной в техническом задании, были исправлены для правильной работы плагина. Добавлены методы для работы дополнительной функциональности. Количество классов осталось неизменным. Однако, в связи с тем, что объект класса Builder не передается в приватные поля класса KompasConnector, связь между ними была изменена на агрегацию. Также измененена связь между BodyParams и MainForm на композицию, так как объект BodyParams хранится в MainForm. Финальная диаграмма классов приведена на рисунке 3.3.

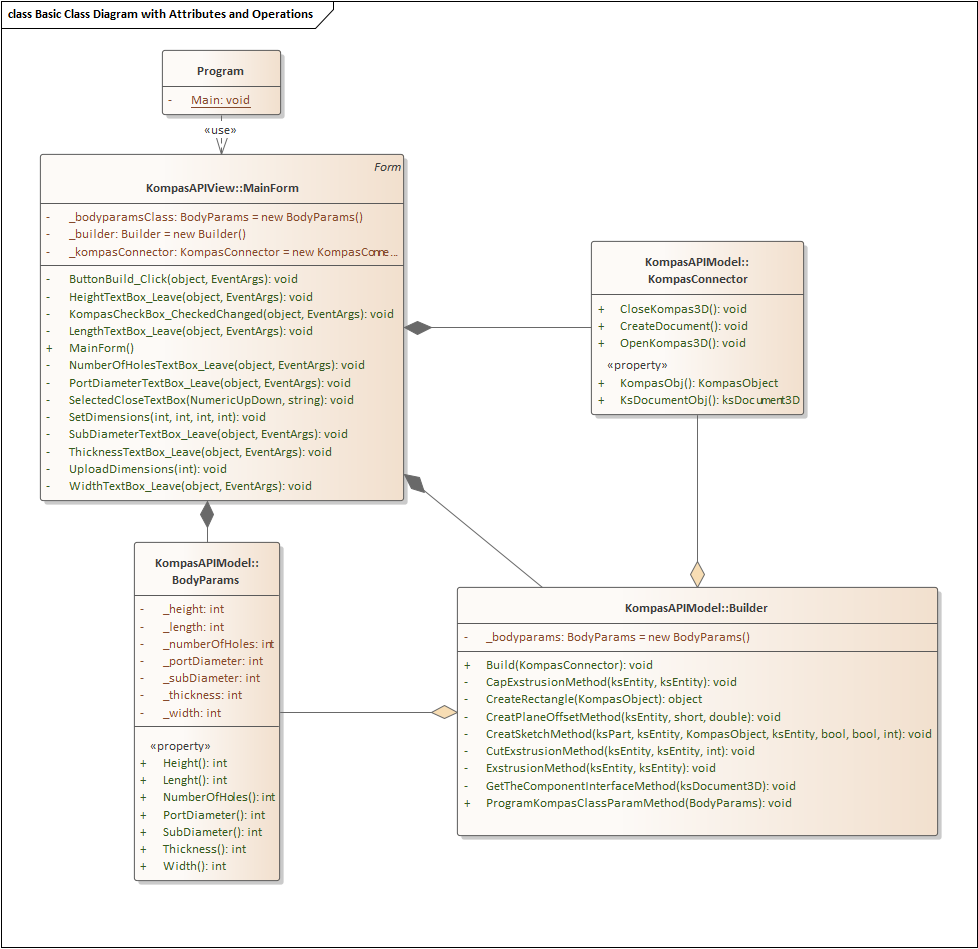


Рисунок 3.4 – Финальная версия диаграммы классов.

Для реализации подсистемы были спроектированы следующие классы:

* MainForm – класс диалогового окна, обеспечивающий взаимодействие между пользователем и программой через форму;
* BodyParams − класс, хранящий в себе все параметры модели, 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.**

Пользовательский интерфейс состоит из отдельных элементов и форм, которые собираются в единое целое. Проектирование интерфейса заставляет думать не только о расположении элементов, но и о динамике перехода пользователя от одного подобного элемента к другому таким образом, чтобы это было максимально удобно и эффективно. Это нетривиальная задача, и для её решения необходимо понимать, как именно пользователь будет действовать при работе с программой [4].

Плагин представляет собой пользовательскую форму с ячейками для ввода параметров. При запуске плагина поля заполнены значениями по умолчанию. Также в макете присутствует переключатель открытия/закрытия компаса. Запуск построения объекта осуществляется кнопкой «Построить». Данная кнопка не доступна, пока не будет включен КОМПАС 3D.

В качестве дополнительного задания была разработана функицональность, задающая в полях габаритов короба (длинна, ширина, высота) рекомендуемые значения. Данные значения задаются при изменении диаметра отверстия сабвуфера на шаблонные(15, 20, 25, 30, 40, 45).

Макет пользовательского интерфейса изображен на рисунке 3.4

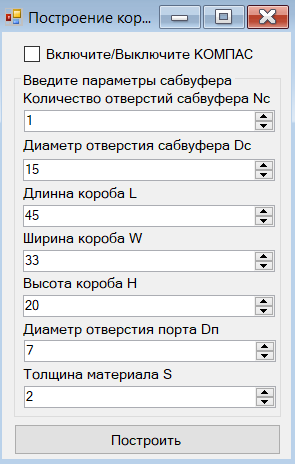


Рисунок 3.5 — Макет пользовательского интерфейса

## **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[4]. Проведем тестирование при максимальных и минимальных параметрах модели.

Минимальные параметры модели:

* Длина короба L: 19 см;
* Ширина короба W: 19 см;
* Высота короба H: 19 см;
* Количество отверстий для сабвуфера Nс: 1
* Диаметр отверстия сабвуфера Dс: 15 см;
* Диаметр отверстия порта(трубы) Dп: 4 см;
* Толщина материала S: 2 см.

Максимальные параметры модели:

* Длина короба L: 106 см;
* Ширина короба W: 106 см;
* Высота короба H: 55 см;
* Количество отверстий для сабвуфера Nс: 2
* Диаметр отверстия сабвуфера Dс: 45 см;
* Диаметр отверстия порта(трубы) Dп: 23;
* Толщина материала S: 4 см.

Модель с минимальными входными параметрами представлена на рисунке 4.1.

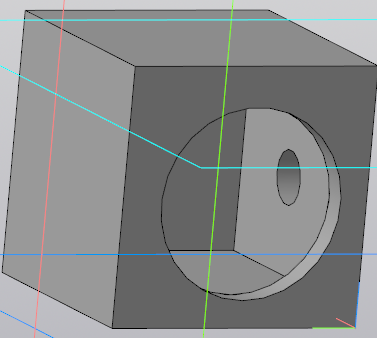


Рисунок 4.1 — Модель короба сабвуфера с минимальными входными параметрами

Модель с максимальными входными параметрами представлена на рисунке 4.2.

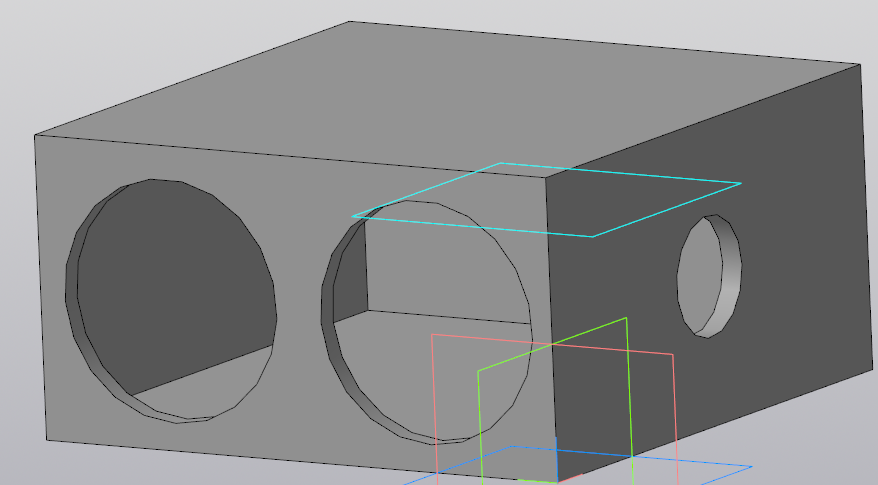


Рисунок 4.2 — Модель короба сабвуфера с максимальными входными параметрами

При выходе из зоны допустимых значений параметров выдается ошибка, приведенная на рисунке 4.3.

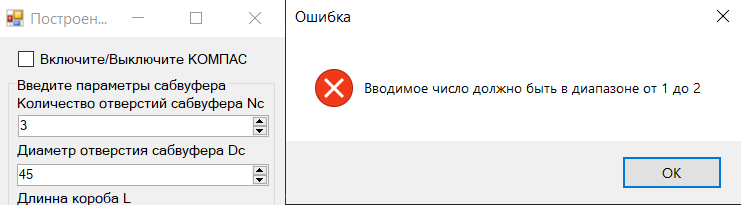


Рисунок 4.3 – Ошибка при вводе неподходящих параметров.

## **4.2 Модульное тестирование**

Юнит-тестирование (блочное тестирование, «unit-testing») —тестирование отдельного элемента изолированно от остальной системы [4]. Список тестовых сценариев для модульного тестирования граничных значений входных параметров представлен в таблице 4.1.

Таблица 4.1 – Список тестовых сценариев.

|  |  |
| --- | --- |
| Название тестового метода | Описание |
| SetSubDiameter\_NegativeTest(int SubDiameter) | Тестирование при вводе некорректных значений диаметра сабвуфера |
| SetSubDiameter\_PositiveTest(int SubDiameter) | Тестирование при вводе коррекнтых граничных значений диаметра сабвуфера |
| SetNumberOfHoles\_NegativeTest (int NumberOfHoles) | Тестирование при вводе некорректных значений количества отверстий сабвуфера |
| SetNumberOfHoles\_PositiveTest(int NumberOfHoles) | Тестирование при вводе коррекнтых граничных количества отверстий сабвуфера |
| SetLenght\_NegativeTest (int Lenght, int SubDiameter, int NumberOfHoles) | Тестирование при вводе некорректных значений длинны сабвуфера |

Окончание таблицы 4.1

|  |  |
| --- | --- |
| SetLenght\_PositiveTest(int Lenght, int SubDiameter, int NumberOfHoles) | Тестирование при вводе коррекнтых граничных длинны сабвуфера |
| SetHeight\_NegativeTest(int Height, int SubDiameter) | Тестирование при вводе некорректных значений высоты сабвуфера |
| SetHeigh\_PositiveTest(int Height, int SubDiameter) | Тестирование при вводе коррекнтых граничных значений высоты сабвуфера |
| SetWidth\_NegativeTest(int Width, int SubDiameter, int Lenght) | Тестирование при вводе некорректных значений ширины сабвуфера |
| SetWidth\_PositiveTest(int Height, int SubDiameter, int Lenght) | Тестирование при вводе коррекнтых граничных значений ширины сабвуфера |
| SetThickness\_NegativeTest(int Thickness) | Тестирование при вводе некорректных значений толщины сабвуфера |
| SetThickness\_PositiveTest(int Thickness) | Тестирование при вводе коррекнтых граничных значений толщины сабвуфера |
| SetPortDiameter\_NegativeTest(int PortDiameter, int SubDiameter) | Тестирование при вводе некорректных значений диаметра порта |
| SetPortDiameter\_PositiveTest(int PortDiameter, int SubDiameter) | Тестирование при вводе коррекнтых граничных значений диаметра порта |

Результаты успешного прохождения всех модульных тестов приведен на рисунке 4.4.

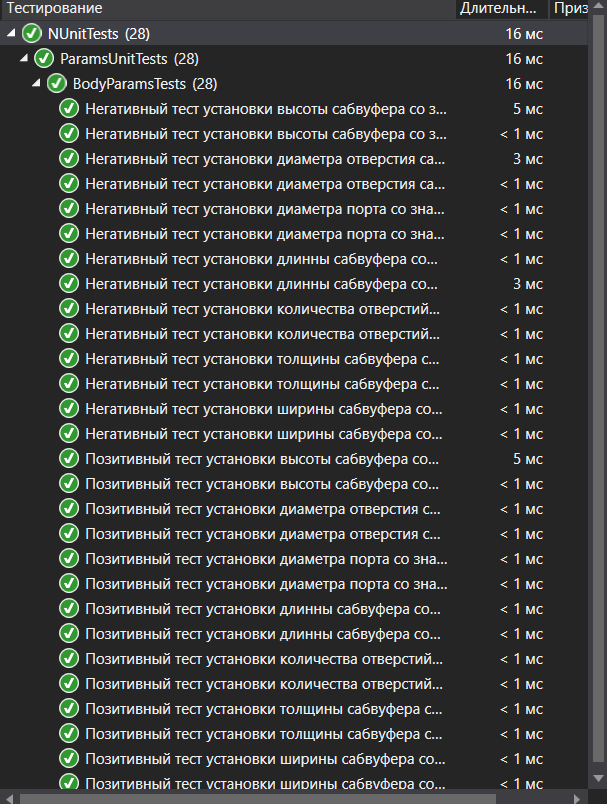


Рисунок 4.4 - Результаты модульных тестов

## **4.4 Нагрузочные тесты**

Построение сборки из деталей осуществляется множество раз последовательно с помощью одного экземпляра приложения «КОМПАС-3D».

Тестирование производилось на ПК со следующей аппаратной конфигурацией:

Процессор: Intel Pentium Pentium P6100

Графическая система: Radeon HD 5145M

ОЗУ: 3 GB DDR 3

При задании максимального количества последовательно строящихся сборок равным 1000, приложение «КОМПАС-3D» завершилось со сбоем на этапе построения 92 сборки по счёту из-за нехватки памяти; при этом на момент сбоя количество потребляемой приложением памяти было в районе 732 мегабайта.

На рисунке 4.5 изображен график зависимости количества непрерывно-строящихся деталей (ось OX) от объема памяти в MB (ось OY), используемой «КОМПАС-3D».

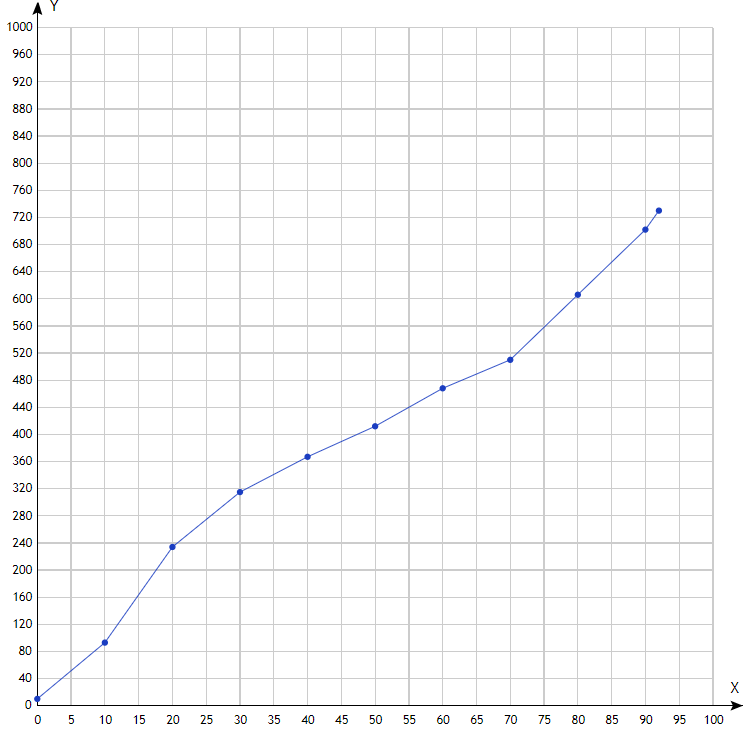


Рисунок 4.5 – График зависимости загруженности памяти от количества деталей.

По графикам можно определить, что зависимость прямая. То есть используемая памяти и загрузка на ЦП, при увеличении количества деталей, увеличивается линейно.

На рисунке 4.6 показан уровень загрузки ЦП (ось OY) в зависимости от количества построеных деталей (ось OX).

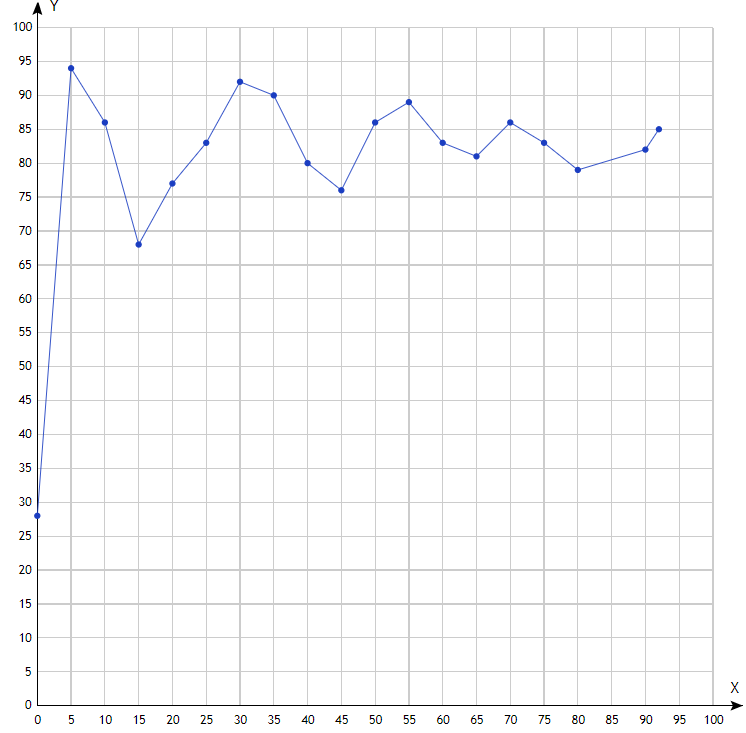


Рисунок 4.6 – График зависимости загруженности процессора от количества деталей.

## **5 Заключение**

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

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

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

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

## **Список используемых источников**

1. КОМПАС(САПР) [Электронный ресурс]. − Режим доступа:

<https://ascon.ru/> (дата обращения: 24.02.2020).

1. subbox.pro [Электронный ресурс]. – Режим доступа: <https://subbox.pro/ru/> (дата обращения: 24.02.2020).
2. Bassbox 6 Pro [Электронный ресурс]. – Режим доступа: <https://www.ht-audio.com/index.html> (дата обращения: 24.02.2020).
3. Новые технологии в программировании: учебное пособие / А.А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. – Томск, 2014. − ­ 176 стр.
4. ФИ для DD [Электронный ресурс]. – Режим доступа: <https://www.drive2.ru/b/411978/> (дата обращения: 20.02.2020).