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

по дисциплине

«Основы разработки САПР» (ОрСАПР)

Выполнил:   
Студент гр. 586-2

Гензе А.А.

« » 2020 г.

Проверил: к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ Калентьев А.А.

« » 2020 г.

Томск 2020

**РЕФЕРАТ**

Пояснительная записка 54 с., 40 рис., 10 таблиц 19 источников, 1 прил.

КОМПАС-3D V17.1, ПЛАГИН, АВТОМОБИЛЬНЫЙ ДИСК, АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ, САПР, API.

Объектом разработки являются автомобильные штампованные диски, устанавливающиеся на переднюю и заднюю оси легкового автомобиля.

Цель работы – разработка плагина, автоматизирующего построение модели «Автомобильный диск» для системы автоматизированного проектирования КОМПАС-3D V17.1 с помощью интегрированной среды разработки Visual Studio 2019 Сommunity.

В процессе работы изучены основные параметры автомобильного штампованного диска и основные функции API системы автоматизированного проектирования КОМПАС-3D.

В результате работы был создан и протестирован плагин, автоматизирующий построение модели «Автомобильный диск».

Отчет по пояснительной записке выполнен в текстовом редакторе Microsoft Word 2016.

Содержание

[1 Введение 4](#_Toc40551570)

[2 Постановка и анализ задачи 5](#_Toc40551571)

[2.1 Описание САПР 5](#_Toc40551572)

[2.3 Описание предмета проектирования 7](#_Toc40551573)

[2.2 Выбор инструментов и средств реализации 11](#_Toc40551574)

[2.3 Назначение плагина 12](#_Toc40551575)

[2.4 Обзор аналогов 12](#_Toc40551576)

[3 Описание реализации 13](#_Toc40551577)

[3.1 Диаграмма вариантов использования (Use Cases) 13](#_Toc40551578)

[3.2 Диаграмма классов 15](#_Toc40551579)

[4 Описание программы для пользователя 22](#_Toc40551580)

[5 Тестирование программы 29](#_Toc40551581)

[5.1 Функциональное тестирование 29](#_Toc40551582)

[5.2 Модульное тестирование 35](#_Toc40551583)

[5.3 Нагрузочное тестирование 40](#_Toc40551584)

[5 Заключение 46](#_Toc40551585)

[Список используемых источников 47](#_Toc40551586)

[Приложение А 49](#_Toc40551587)

# 1 Введение

В настоящее время каждое предприятие старается максимально автоматизировать процесс производства продукции. Автоматизируется все аспекты производства, от проектирования, до реализации. Это в значительной степени повышает продуктивность, гибкость и качество продукции. Уменьшает количество брака, затраты на производство и риски несчастных случаев.

Использование автоматизации в проектировании позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений и технологий в той или иной области техники. Она позволяет значительно повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением [1].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Автомобильный диск» для системы автоматизированного проектирования КОМПАС-3D V17.1 [2] с помощью интегрированной среды разработки Visual Studio 2019 Сommunity [3].

Интегрированная среда разработки Visual Studio ‒ это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

# 2 Постановка и анализ задачи

В рамках учебной дисциплины ОрСАПР, в соответствии с технически заданием, требовалось разработать плагин, который на основе входных параметров, взаимодействуя с системой КОМПАС-3D, строит модель автомобильного штампованного диска. Также плагин должен позволять изменять входные параметры диска и иметь параметры по умолчанию. Изменяемые параметры: диаметр и ширина диска, количество отверстий под болты и диаметр их расположения, диаметр центрального отверстия, количество вентиляционных отверстий и их диаметр.

## 2.1 Описание САПР

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

«Компас-3D» — семейство систем автоматизированного проектирования с возможностями оформления проектной и конструкторской документации согласно стандартам серии ЕСКД и СПДС. Разрабатывается российской компанией «Аскон». Название линейки является акронимом от фразы «комплекс автоматизированных систем», в торговых марках используется написание заглавными буквами — «КОМПАС».

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

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

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

Система «Компас-3D» предназначена для создания трёхмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология позволяет быстро получать модели типовых изделий на основе однажды спроектированного прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства. Ключевой особенностью «Компас-3D» является использование собственного математического ядра "C3D" и параметрических технологий, разработанных специалистами компании «Аскон».

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

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

## 2.3 Описание предмета проектирования

Предметом проектирования является автомобильный штампованный диск. Штампованные диски распространены широко среди автовладельцев за счет того, что часто в заводской комплектации с конвейера машины выходят именно со штамповкой на колесах. Они ценятся автовладельцами за их дешевизну и простоту в производстве, возможность легко восстановить форму после деформации [4]. Данные диски устанавливаются непосредственно на переднюю и заднюю колесные оси автомобиля.

Диапазон параметров диска:

* ширина диска (B): от 114,3 до 254 мм;
* диаметр диска (D): от 330.2 до 482.6 мм;
* диаметр расположения болтов (PCD): от 98 до 205 мм;
* количество отверстий под болты (LZ): от 3 до 6 шт;
* центральное отверстие диска (DIA): от 54.1 до 161 мм;
* количество вентиляционных отверстий (KN): от 10 до 20 шт;
* диаметр вентиляционных отверстий (DN): 15 до 40 мм.

Изображение предмета проектирования с обозначенными параметрами приведено на рисунке 2.1 ‒ 2.2.

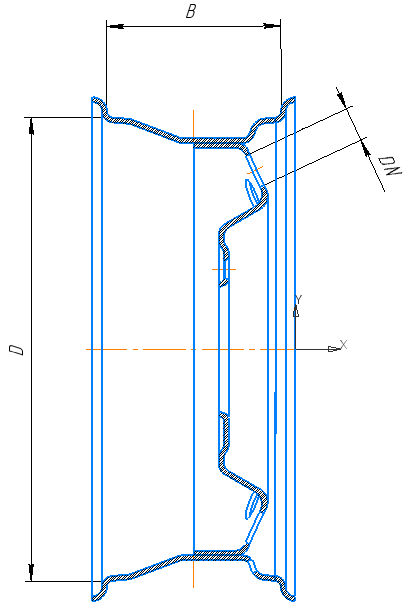


Рисунок 2.1 ‒ Размерные выноски автомобильного штампованного диска в разрезе

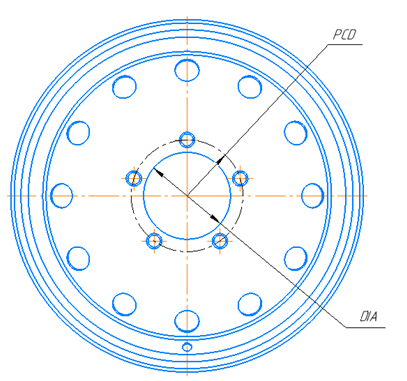


Рисунок 2.2 ‒ Размерные выноски на лицевой стороне диска

Параметры предмета проектирования регламентируются ГОСТ Р 53824-2010 «Колеса неразборные» [5]. Основные параметры диска представлены в таблице 2.1.

Таблица 2.1 ‒ Основные параметры диска

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Диаметр диска (D), дюйм | Ширина диска (В), дюйм | Сверловка | Центральное отверстие (DIA), мм | |
| 13 | 5; 5.5 | 4\*98 | 58.6; 60.1 | |
| 4.5; 5; 5.5 | 4\*100 | 54.1; 56.6; 57.1 | |
| 4.5; 5 | 4\*114.3 | 69.1 | |
| 14 | 5; 5.5; 6 | 4\*98 | 58.1; 58.6 | |
| 4.5; 5; 5.5; 6 | 4\*100 | 54.1; 56.1; 56.6; 57.1; 60.1 | |
| 5.5 | 4\*108 | 63.3; 65.1 | |
| 5.5 | 4\*114.3 | 56.6; 66.1 | |
| 5; 6 | 5\*100 | 57.1 | |
| 5.5 | 5\*120 | 67.1 | |
| 15 | 4.5 | 3\*112 | 57.1 | |
| 5.5; 6 | 4\*98 | 58.1; 58.6 | |
| 5; 5.5; 6 | 4\*100 | 54.1; 56.1; 56.6; 60.1 | |
| 6; 6.5 | 4\*108 | 63.3; 65.1 | |
| 5.5; 6 | 4\*114.3 | 56.6; 64.1; 66.1 | |
| 6 | 5\*98 | 58.1 | |
| 5.5; 6 | 5\*100 | 57.1 | |
| 6 | 5\*105 | 56.6 | |
| 6 | 5\*108 | 60.1; 63.3 | |
| 6 | 5\*110 | 65.1 | |
| 6 | 5\*112 | 57.1; |
| 6; 6.5; 7; 8; 10 | 5\*114.3 | 54.1; 60.1; 66.1; 67.1; 71.6; 84.1 |
| 6 | 5\*118 | 71.1 |
| 6 | 5\*130 | 84.1 |
| 5.5; 6; 7; 8; 10 | 5\*139.7 | 98.6; 110.1; 108.5 |
| 5.5; 6.5 | 5\*160 | 65.1 |
| 8 | 5\*165.1 | 131 |
| 7; 8; 10 | 6\*139.7 | 108.5; 110.1; 110.5 |
| 5.5 | 6\*205 | 161.1 |
| 16 | 6; 6.5; 7 | 4\*108 | 63.3; 65.1 |
| 6 | 5\*98 | 58.1 |
| 6; 6.5 | 5\*100 | 56.1; 57.1 |
| 6.5 | 5\*105 | 56.6 |
| 6.5; 7 | 5\*108 | 63.3; 65.1 |
| 6.5 | 5\*110 | 65.1 |
| 6; 6.5; 7 | 5\*112 | 57.1; 66.6; 70.1 |
| 6; 6.5; 7; 8 | 5\*114.3 | 54.1; 60.1; 64.1; 66.1; 67.1; 71.6; 84.1 |
| 6.5 | 5\*115 | 70.1 |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Диаметр диска (D), дюйм | Ширина диска (В), дюйм | Сверловка | Центральное отверстие (DIA), мм |
|  | 6 | 5\*118 | 71.1 |
| 6.5; 7 | 5\*120 | 65.1; 67.1; 72.6 |
| 8 | 5\*127 | 75.1 |
| 6; 6.5; 8 | 5\*130 | 78.1; 84.1; 89.1 |
| 6.5; 7; 8; 10 | 5\*139.7 | 95.3; 98.1; 98.6; 108.5; 110.1 |
| 8; 10 | 5\*150 | 110.1; 110.5 |
| 5.5; 6.5 | 5\*160 | 65.1 |
| 7 | 5\*165 | 122.5 |
| 5.5; 6.5 | 6\*130 | 84.1 |
| 6.5; 7; 8; 10 | 6\*139.7 | 67.1; 92.5; 106.1; 108.5; 110.1 |
| 5.5 | 6\*170 | 130.1 |
| 5; 6 | 6\*180 | 138.8 |
| 5.5; 6 | 6\*205 | 121.5; 161.1 | |
| 17 | 6.5; 7 | 4\*100 | 60.1 | |
| 7.5 | 4\*108 | 65.1 | |
| 7 | 5\*100 | 56.1; 57.1 | |
| 7 | 5\*105 | 56.6 | |
| 7; 7.5; 8.5 | 5\*108 | 63.3; 65.1 | |
| 6.5; 7 | 5\*112 | 57.1; 66.6; 70.1 | |
| 6.5; 7 | 5\*114.3 | 60.1; 64.1; 67.1 | |
| 7; 7.5 | 5\*120 | 65.1; 72.6 | |
| 8 | 5\*127 | 75.1 | |
| 8 | 5\*139.7 | 110.1 | |
| 7; 7.5 | 6\*114.3 | 66.1 | |
| 7; 8 | 6\*139.7 | 100.1; 106.1; 110.1 | |
| 18 | 8 | 5\*108 | 70.1 | |
| 8 | 5\*112 | 70.1 | |
| 8 | 5\*114.3 | 71.6 | |
| 7.5 | 6\*114.3 | 66.1 | |
| 8 | 6\*139.7 | 106.1 | |
| 19 | 8; 8.5 | 5\*112 | 70.1 | |
| 8.5 | 5\*114.3 | 71.6 | |

На основе данных таблицы 2.1 зависимости параметров следующие:

* от диаметра диска зависит количество отверстий под болты;
* от количества болтов зависит диаметр их расположения;
* от диаметра расположения болтов зависит диаметр центрального отверстия и ширина.

Также от диаметра будет зависеть минимальное и максимальное количество вентиляционных отверстий. Данная зависимость представлена в таблице 2.2.

Таблица 2.2 ‒ Зависимости минимального и максимального количества вентиляционных отверстий от диаметра диска

|  |  |  |
| --- | --- | --- |
| Диаметр диска | Мин. Кол-во | Макс. Кол-во |
| 13 | 10 | 12 |
| 14 | 10 | 14 |
| 15 | 10 | 14 |
| 16 | 12 | 16 |
| 17 | 12 | 16 |
| 18 | 14 | 18 |
| 19 | 16 | 20 |

## 2.2 Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2 [6] для системы Компас 3D V17.1. Инструментом тестирования и создания модульных тестов был выбран обозреватель тестов расширения для Visual Studio с тестовым фреймворком NUnit версии 3.12 [7].

Для реализации пользовательского интерфейса использовалась платформа Window Forms [8].

Взаимодействие плагина с системой Компас 3D осуществляется посредством интерфейсов, называемых API [9]. В КОМПАС 3D на данный момент существует API двух версий API 5 и API 7 [10]. Явных преимуществ между версиями нет, поскольку обе версии реализуют различные функции системы и взаимно дополняют друг друга. Для выполнения лабораторных была выбрана версия API 5, так как для полноценной реализации плагина «Автомобильный диск» достаточно методов и свойств интерфейсов API 5.

## 2.3 Назначение плагина

Назначение, разрабатываемого плагина обусловлено быстрым моделированием автомобильного диска. Благодаря данному расширению авто-дизайнеры могут наглядно рассмотреть спроектированную модель диска и при необходимости поменять его внешний вид или характеристики под необходимые.

## 2.4 Обзор аналогов

Одним из аналогов разрабатываемого модуля является SolidWorks. SolidWorks представляет собой мощный инструмент для 3D моделирования и автоматизированного проектирования сложных изделий различного назначения. Это полноценный набор для конструирования изделий в цифровом виде, который содержит в себе множество дополнительных инструментов, позволяющих производить над моделью виртуальные технические испытания [11]. Пример создания автомобильного диска в программе SolidWorks приведен на рисунке 2.3.

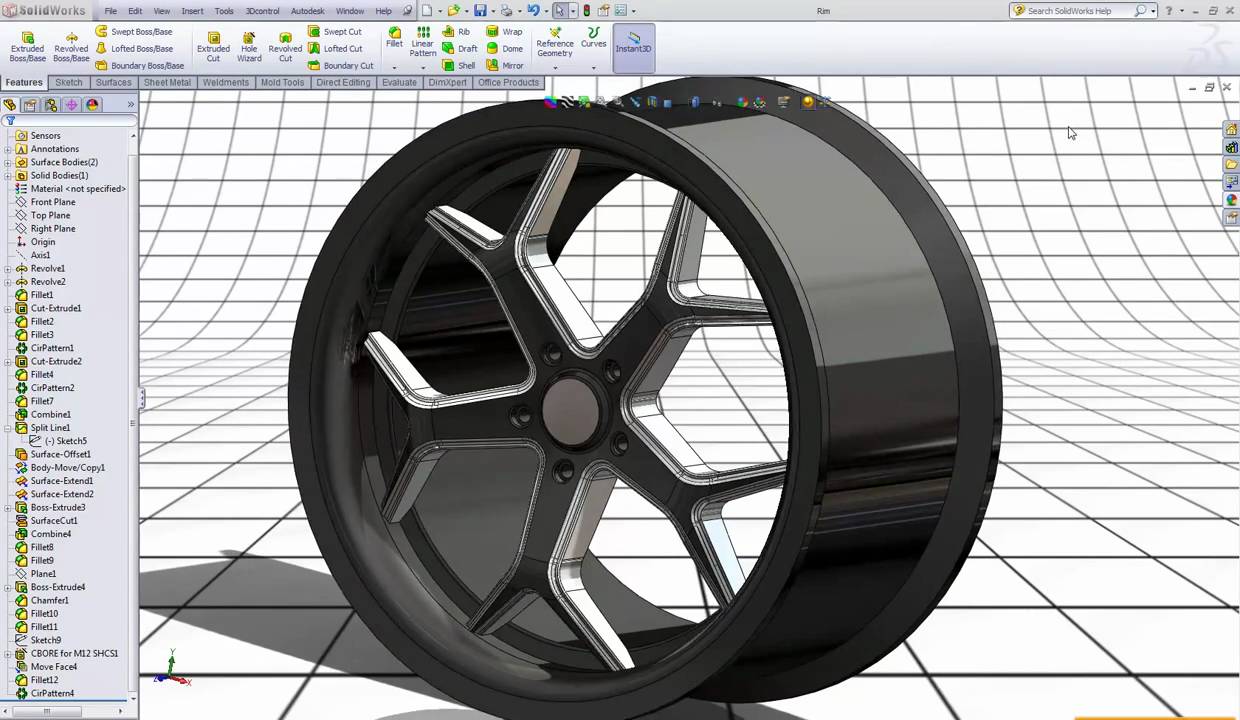


Рисунок 2.3 ‒ Пример создания автомобильного диска в SolidWorks

# 3 Описание реализации

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML [12].

UML язык графического описания для объектного моделирования в обрасти 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.

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

## 3.1 Диаграмма вариантов использования (Use Cases)

Диаграмма вариантов 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 [13].

Изначальная диаграмма использования представлена на рисунке 3.1.

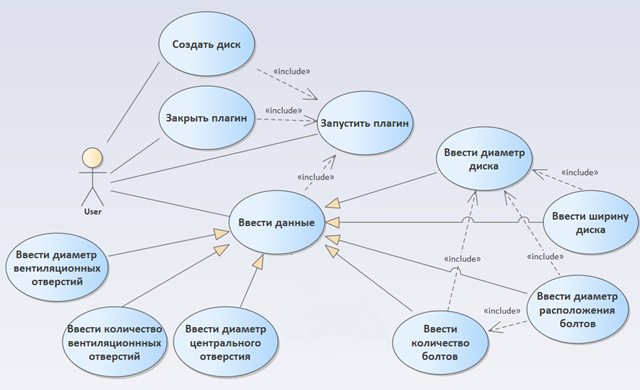


Рисунок 3.1 ‒ Изначальная диаграмма использования

В результате изменений, внесенных заказчиком в функциональность разрабатываемого плагина и недочетами на этапе проектирования, диаграмма использования была изменена. Произошли следующие изменения:

* добавлены сущности открытия и закрытия компаса, т.к. не открыв компаса построить объект невозможно;
* убраны ненужные связи «include» между параметрами диаметра диска, ширины, количества болтов и диаметром расположения болтов, т.к. пользователь может в любой момент выбрать другое значение любого из перечисленных параметров;
* изменено содержание сущностей выбора параметров с «Ввести…» на «Выбрать…»;
* по желанию заказчика добавлена возможность включать и выключать отрисовку хампа типа «Н». Хампами (от англ. hump, «возвышение, бугор») называют кольцевые выступы вдоль закраин колесного диска, предназначенного для бескамерной покрышки. Основное назначение хампов ‒ надежная фиксация борта покрышки в поворотах, чтобы не допустить разгерметизации колеса. В обозначениях дисков, имеющих один хамп вдоль внешней стороны, присутствует одна буква Н [14].

Измененная диаграмма использования представлена на рисунке 3.2.



Рисунок 3.2 ‒ Измененная диаграмма использования

## 3.2 Диаграмма классов

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области (в процессе анализа) или элементы программной системы (в процессах проектирования и реализации) [15].

Изначальная диаграмма классов представлена на рисунке 3.3.

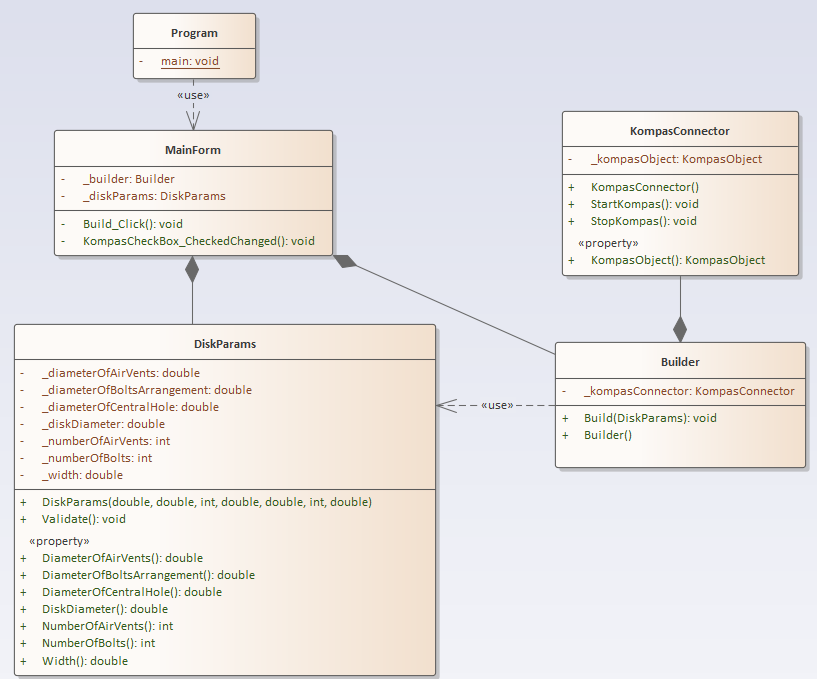


Рисунок 3.3 ‒ Изначальная диаграмма классов

На этапе проектирования подсистемы были следующие классы:

* MainForm – класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой;
* DiskParams − класс, хранящий в себе все параметры проектируемой 3D-модели, осуществляет проверку зависимых параметров;
* KompasConnector – класс, отвечающий за работу с API КОМПАС 3D.
* Builder – класс, отвечающий за вызов методов API, необходимых для постройки проектируемой 3D-модели.

В ходе реализации было выявлено ряд недочетов и непродуманных нюансов. Также, заказчиком, были добавлены изменения в функциональность плагина в виде дополнительного флажка, который отвечает за включение и выключение отрисовки хампа типа «H». В итоге решение было поделено на четыре проекта: KompasAPI, Model, View и UnitTests.

Проект KompasAPI содержит следующие классы:

* Builder – отвечает за вызов методов API, необходимых для постройки проектируемой 3D-модели. Также хранит поля с константными значения координат отрезков, необходимых для отрисовки эскизов;
* KompasConnector ‒ отвечающий за открытие и закрытие программы Компас 3D.

Итоговая диаграмма классов проекта KompasAPI представлена на рисунке 3.4.

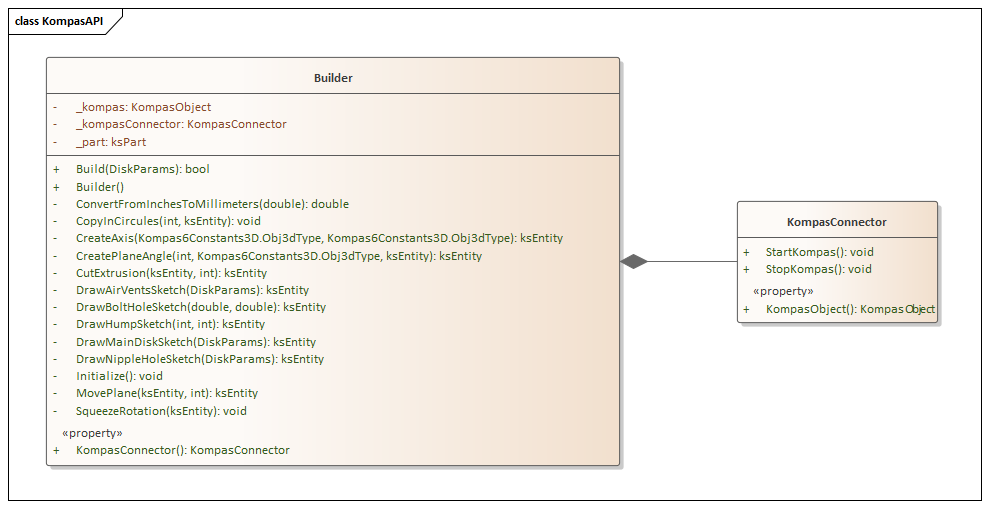


Рисунок 3.4 ‒ Диаграмма классов проекта KompasAPI

Во избежание нагромождения, поля с константными значениями координат не отображены на диаграмме классов.

Итоговые классы проекта Model следующие:

* DiskParams – хранит параметры диска по умолчанию и, в последующем, установленные пользователем параметры проектируемой 3D-модели;
* ValuesRelation ‒ объекты данного класса содержат зависимости доступных значений диска;
* AvailableParameters ‒ инициализирует список объектов класса ValuesRelation. Также, в зависимости от входных значений, осуществляет выборку по этому списку подходящих параметров и формирует из них новый список;
* ParamsValidator ‒ проверяет, допустимо ли значение;
* LineSegment ‒ класс параметров отрезка;
* ArcSegment ‒ класс параметров дуги;
* AvailableValuesChangeType ‒ перечисление типов изменения допустимых значений параметров;
* ValuesChangeEventArgs ‒ класс аргументов события изменения доступных значений параметров;
* DoubleExtensions ‒ расширение для типа Double.

На рисунке 3.5 представленная диаграмма классов проекта Model.

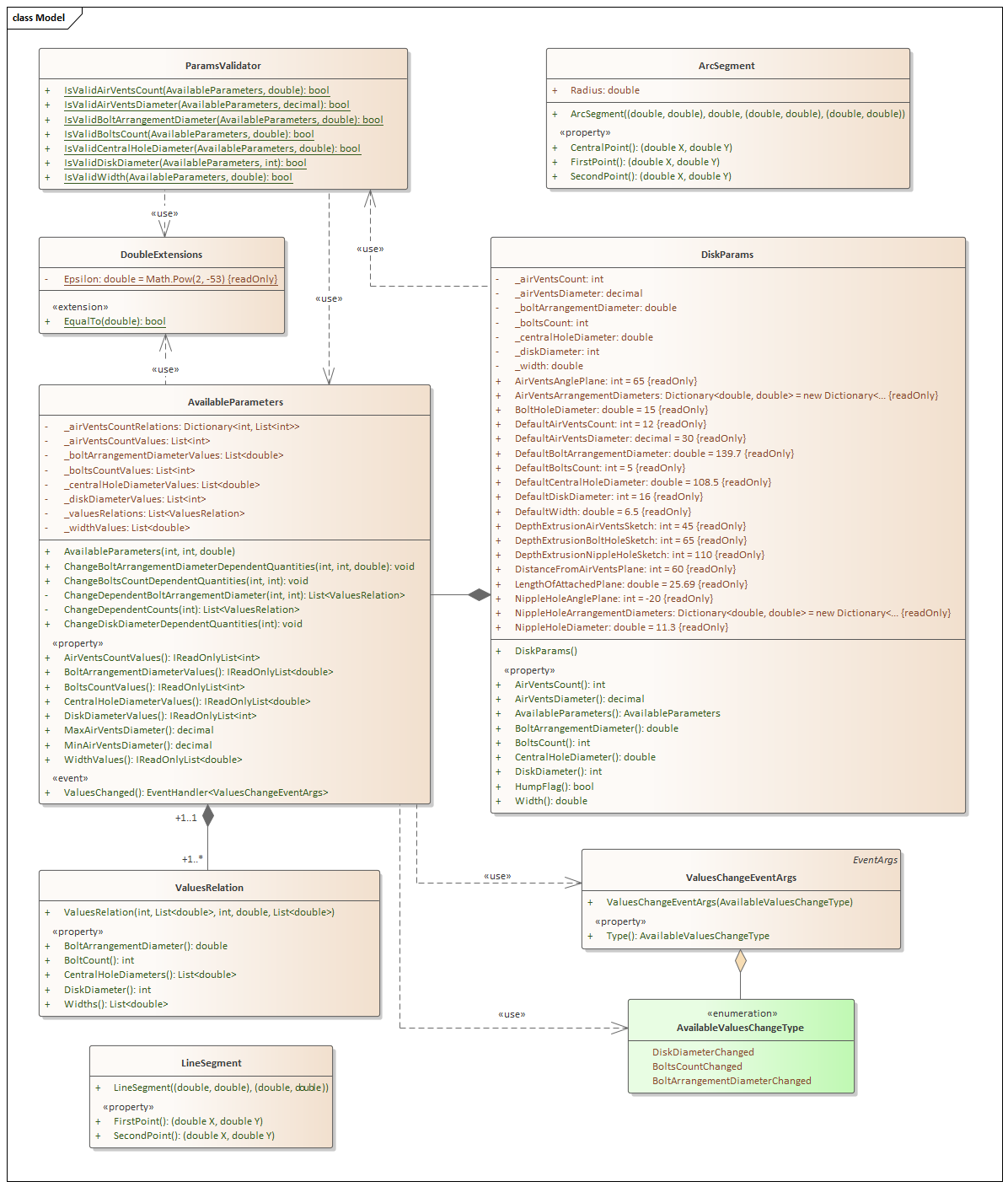


Рисунок 3.5 ‒ Диаграмма классов проекта Model

Классы LineSegment и ArcSegment проекта Model используются в классе Build проекта KompasAPI. Поэтому связей с классами проекта Model у них нет.

Классы проекта View:

* MainForm ‒ класс диалогового окна, который обеспечивает взаимодействие между пользователем и программой;
* Program ‒ главная точка входа для приложения.

На рисунке 3.6 представленная диаграмма классов проекта View.

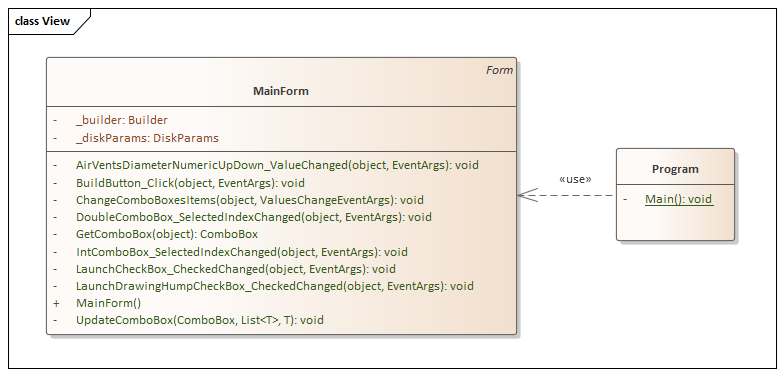


Рисунок 3.6 ‒ Диаграмма классов проекта View

Общая диаграмма классов приложения представлена на рисунке 3.7.

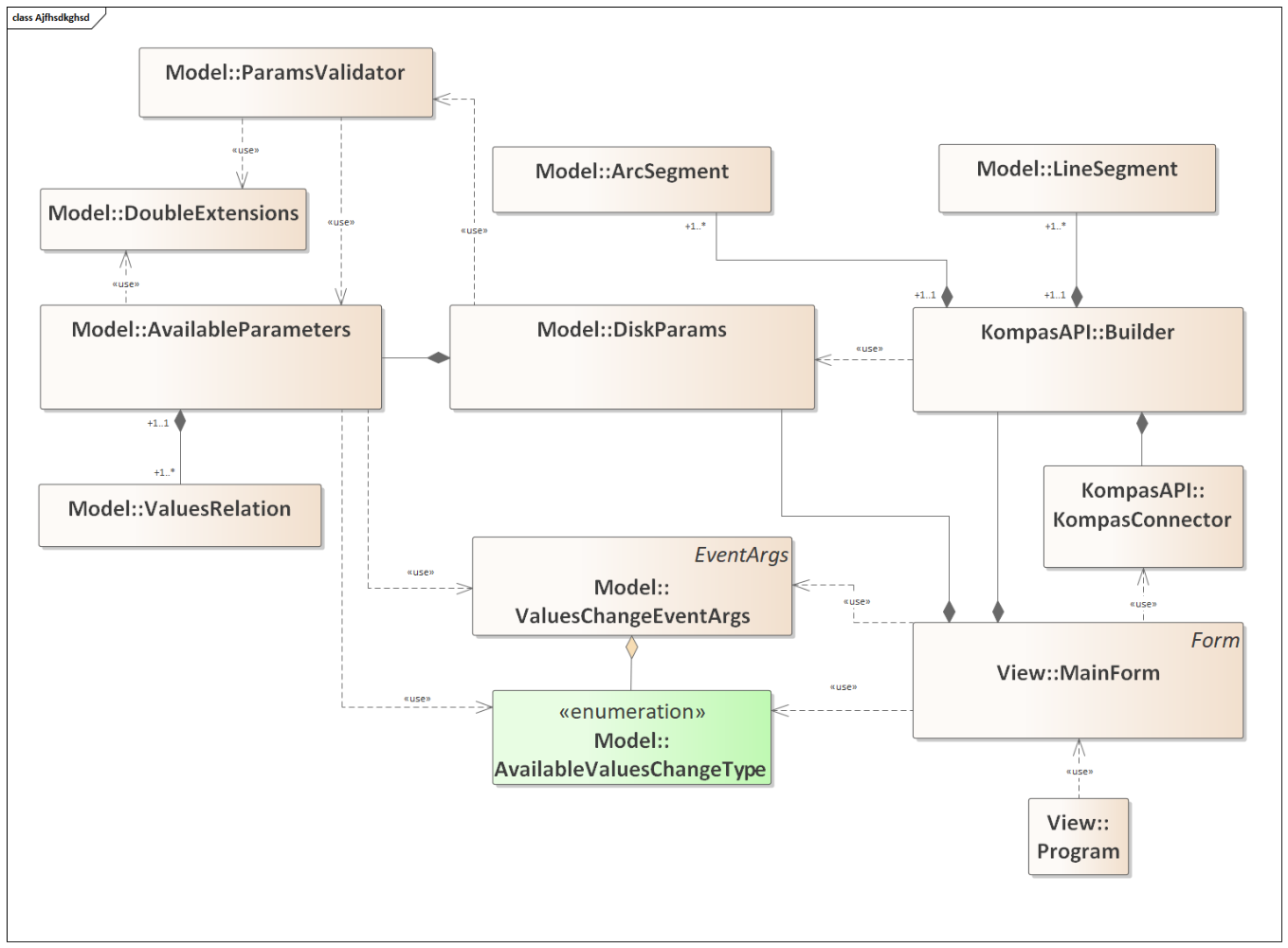


Рисунок 3.7 ‒ Общая диаграмма классов

Такое различие между изначальной и итоговой диаграммой классов обусловлено отсутствием необходимого опыта в разработке. Из-за этого на этапе проектирования отсутствовал класс для проверки значений ParamsValidator, класс зависимостей параметров диска и инициализирующий его класс ValuesRelation и AvailableParameters соответственно. При написании предопределенного делегата EventHandler [16] потребовалось наличие перечисления AvailableValuesChangeType и класс ValuesChangeEventArgs, его использующий. Также на этапе реализации стало ясно, что для построения эскиза диска требуется большое количество констант, отсюда возникла необходимость в создании классов для хранения координат отрезков ‒ ArcSegment и LineSegment.

При реализации дополнительной функциональности в класс DiskParams было добавлено свойство HumpFlag, отвечающее за состояние флажка переключения отрисовки хампа. В класс Builder был добавлен метод DrawHumpSketch, отвечающий за отрисовку эскиза. Также в Builder были добавлены и проинициализированы три поля, в которых хранятся константные параметры отрезков для эскиза.

# 4 Описание программы для пользователя

Разрабатываемый плагин состоит из одного диалогового окна. Всю область окна можно условно разделить на четыре части:

* область флажка включения и отключения системы Компас 3D;
* область с выпадающими списками основных параметров диска;
* область с выпадающими списками дополнительных параметров диска;
* область, в которой расположена кнопка построения диска.

Изначальный внешний вид диалогового окна плагина представлен на рисунке 4.1.

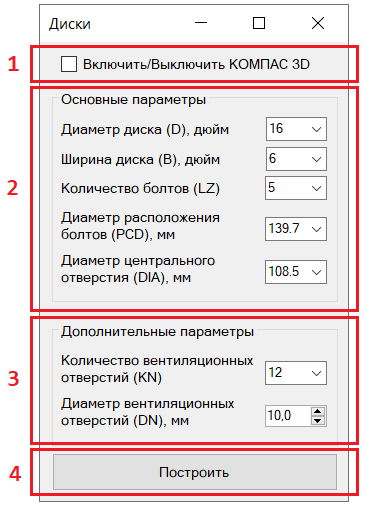


Рисунок 4.1 – Начальный вид диалогового окна

Для реализации дополнительной функциональности необходимо было добавить флажок для включения и отключения отрисовки хампа. Данный параметр считается дополнительным, поэтому флажок был помещен в область 3. Итоговый внешний вид диалогового окна представлен на рисунке 4.2.

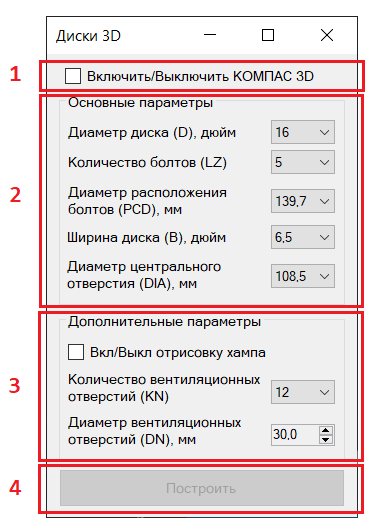


Рисунок 4.2 – Итоговый внешний вид диалогового окна

Для обеспечения безошибочной работы с плагином, пользователю был запрещен ввод собственных значений. Также, при отсутствии галочки отвечающей за включение системы Компас 3D кнопка построения диска будет неактивна, что видно из рисунка 4.2.

При запуске плагина все поля заполнены значениями по умолчанию, это видно на рисунке 4.2. При нажатии на любое поле со значением, кроме поля задания диаметра вентиляционных отверстий, появиться список доступных значений. Демонстрация работы выпадающего списка приведена на рисунке 4.3.

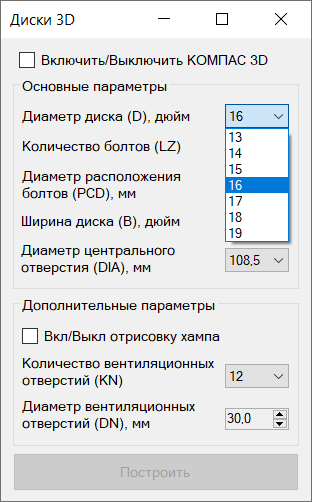


Рисунок 4.3 ‒ Демонстрация работы выпадающего списка

При этом, для определенного значения диаметра существует определенные значения количества болтов и вентиляционных отверстий (рисунок 4.4).

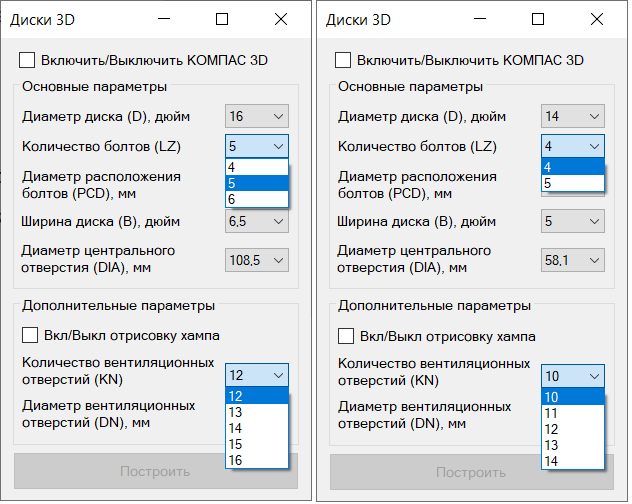


Рисунок 4.4 ‒ Демонстрация зависимости количества болтов и вентиляционных отверстий от диаметра диска

Для определенного значения количества болтов существует определенное значение диаметра их расположения (рисунок 4.5).

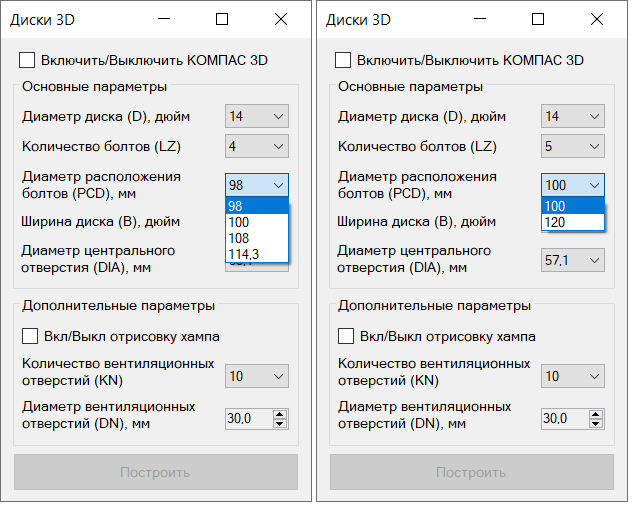


Рисунок 4.5 – Демонстрация зависимости диаметра расположения болтов от их количества

А для определенного значения диаметра расположения болтов существуют определенные значения ширин диска (рисунок 4.6) и диаметров центрального отверстия (рисунок 4.7).

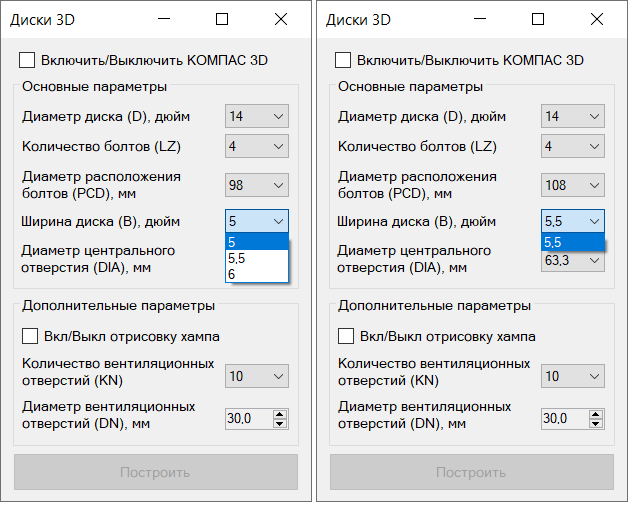


Рисунок 4.6 – Демонстрация зависимости ширин диска от диаметра расположения болтов

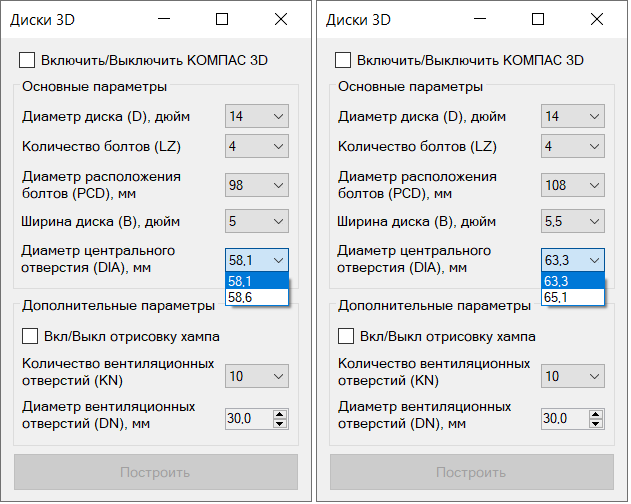


Рисунок 4.7 – Демонстрация зависимости диаметров центрального отверстия от диаметра расположения болтов

Значение единственного поля, которое не является выпадающим списком задается при помощи стрелочек. При единичном нажатии значение увеличивается на 0.1 мм, а при зажатии какой-либо стрелочки значение будет быстро увеличиваться или уменьшаться, в зависимости от того, какая стрелка зажата.

Т.к. система Компас3D открывается с помощью установки флажка, то необходимо было обработать случай, когда пользователь закрывает систему вручную, ведь в таком случае кнопка построения остается активной. Если пользователь в данном случае нажмет на кнопку, выведется окно с сообщением: «Компас закрыт» (рисунок 4.8). Если пользователь при выключенной вручную системе нажмет на флажок выключения системы, то также выведется окно с сообщением: «Компас уже закрыт» (рисунок 4.9). В случае, если система запущена, а флажок не установлен, и пользователь его попытается установить, второй раз система не запустится. В таком случае станет активной кнопка построения и больше никаких видимых изменений не произойдет.

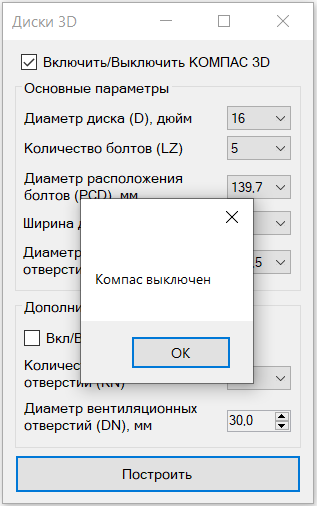


Рисунок 4.8 ‒ Демонстрация нажатия кнопки «Построить» при выключенной вручную системе Компас3D

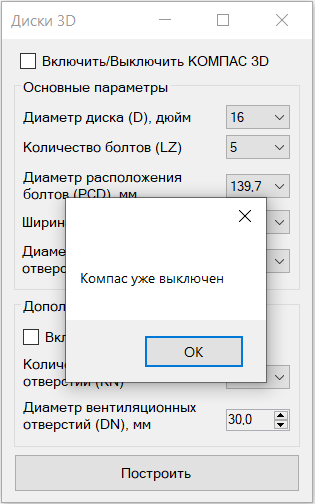


Рисунок 4.9 ‒ Демонстрация нажатия на установленный флажок при выключенной вручную системе Компас3D

Результат нажатия кнопки «Построить» с по умолчанию установленными значениями представлен на рисунке 4.10.

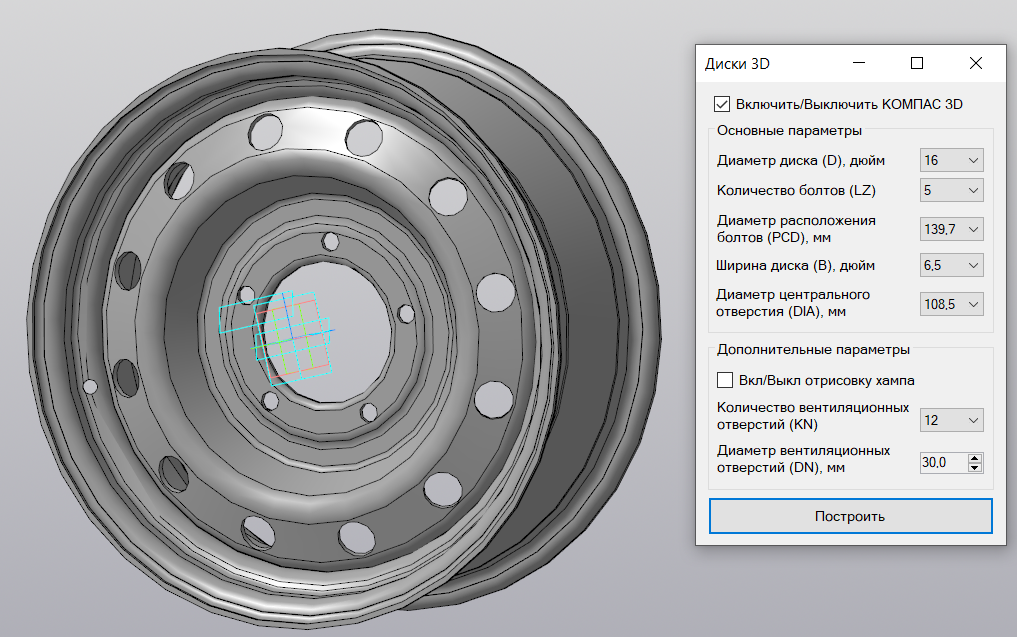


Рисунок 4.10 ‒ Модель, построенная по заданным параметрам по умолчанию

# 5 Тестирование программы

Тестирование ‒ это процесс испытания программы, целью которого является, проверка работоспособности, выявление ошибок. Также проверка позволяет установить, удовлетворяет ли программа всем заявленным требованиям.

## 5.1 Функциональное тестирование

При функциональном тестировании [17] проверялась корректность работы плагина «Автомобильный диск», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами.

Проведено тестирование максимальных и минимальных параметров модели.

На рисунках 5.1 ‒ 5.2 представлена проверка размеров модели с минимальным диаметром диска и минимальными допустимыми параметрами.

Минимальные параметры диска:

* диаметр диска 330.2 мм (13 дюймов);
* количество болтов 4 шт;
* диаметр расположения болтов 98 мм;
* ширина диска 127 мм (5 дюймов);
* диаметр центрального отверстия 58.6 мм;
* количество вентиляционных отверстий 10 шт;
* диаметр вентиляционных отверстий 15 мм.

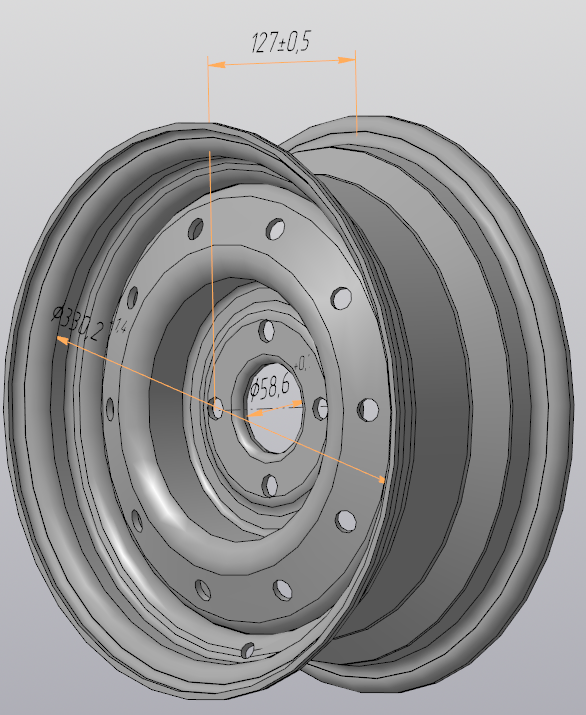


Рисунок 5.1 – Модель с минимальным диаметром диска и минимальными допустимыми параметрами

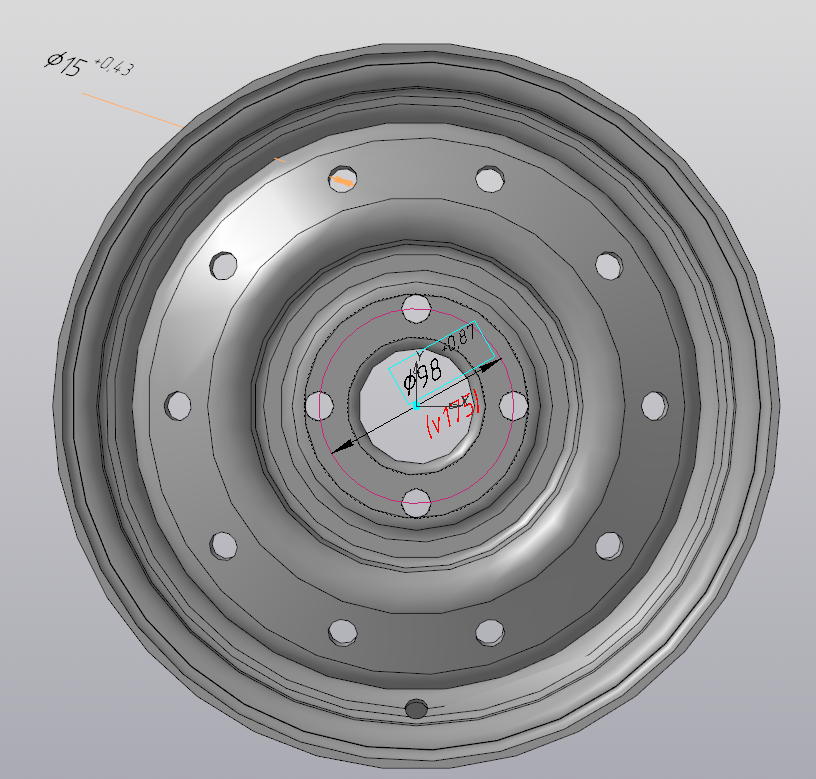


Рисунок 5.2 – Модель с минимальным диаметром диска и минимальными допустимыми параметрами

На рисунках 5.3 ‒ 5.4 представлена проверка размеров модели с максимальным диаметром диска и максимальными допустимыми параметрами.

Максимальные параметры диска:

* диаметр диска 482.6 мм (19 дюймов);
* количество болтов 5 шт;
* диаметр расположения болтов 114.3 мм;
* ширина диска 215.9 мм (8.5 дюймов);
* диаметр центрального отверстия 71.6 мм;
* количество вентиляционных отверстий 20 шт;
* диаметр вентиляционных отверстий 40 мм.

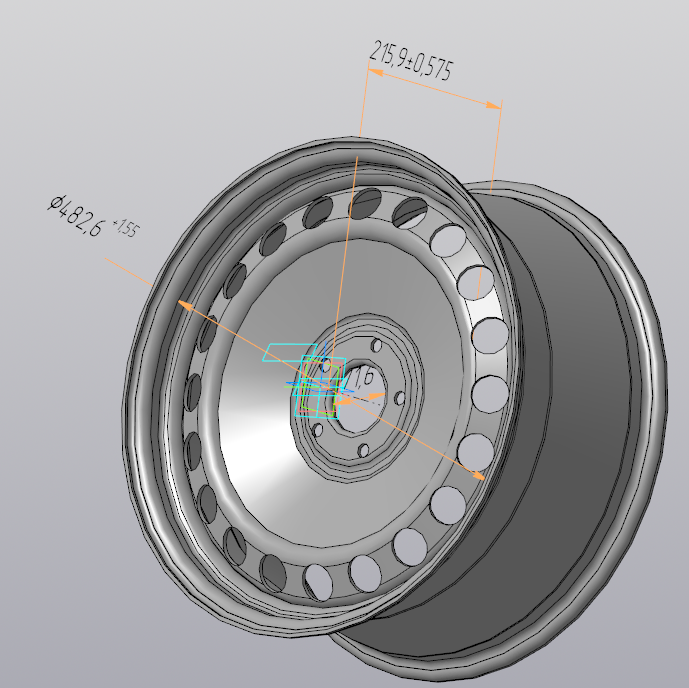


Рисунок 5.3 – Модель с максимальным диаметром диска и максимальными допустимыми параметрами



Рисунок 5.4 – Модель с максимальным диаметром диска и максимальными допустимыми параметрами

На рисунках 5.5 ‒ 5.6 представлена проверка размеров модели с максимальным центральным отверстием диска и минимальными допустимыми параметрами:

* диаметр диска 381 мм (15 дюймов);
* количество болтов 6 шт;
* диаметр расположения болтов 205 мм;
* ширина диска 139.7 мм (5.5 дюймов);
* диаметр центрального отверстия 161.1 мм;
* количество вентиляционных отверстий 10 шт;
* диаметр вентиляционных отверстий 15 мм.

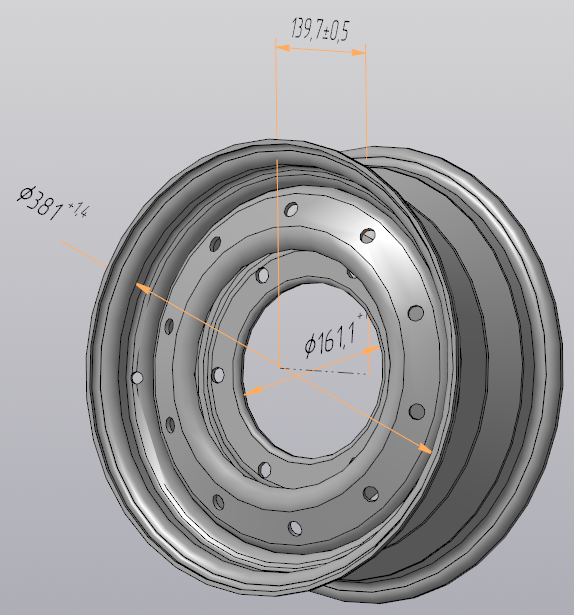


Рисунок 5.5 – Модель с максимальным центральным отверстием диска и минимальными допустимыми параметрами

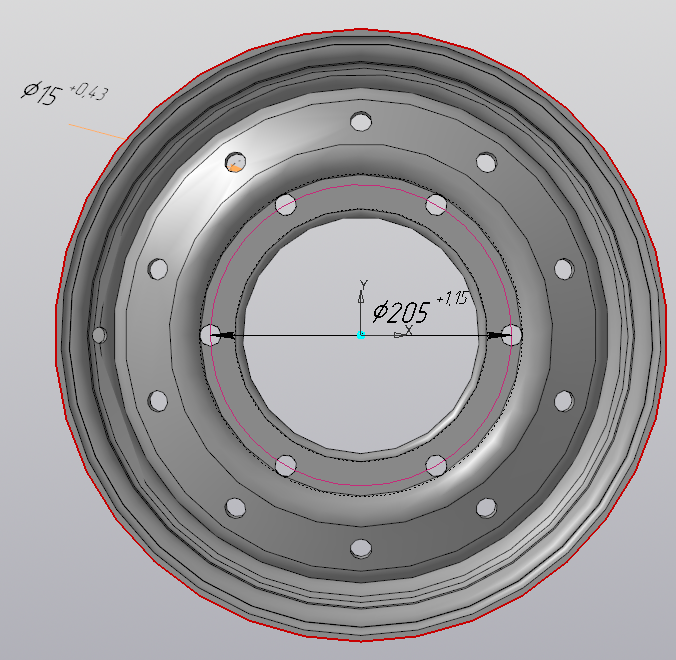


Рисунок 5.6 – Модель с максимальным центральным отверстием диска и минимальными допустимыми параметрами

На рисунке 5.7 представлена проверка размеров модели с максимальной шириной диска и остальными параметрами, установленными в значения по умолчанию:

* диаметр диска 406.4 мм (16 дюймов);
* количество болтов 5 шт;
* диаметр расположения болтов 139.7 мм;
* ширина диска 254 мм (10 дюймов);
* диаметр центрального отверстия 108.5 мм;
* количество вентиляционных отверстий 12 шт;
* диаметр вентиляционных отверстий 30 мм.

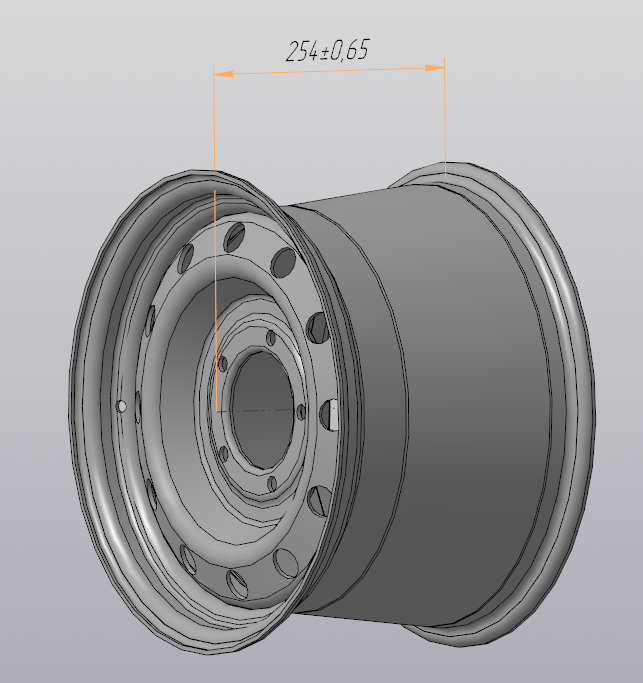


Рисунок 5.7 – Модель с максимальной шириной диска

На рисунке 5.8 продемонстрирован обод с хампом типа Н и без него.

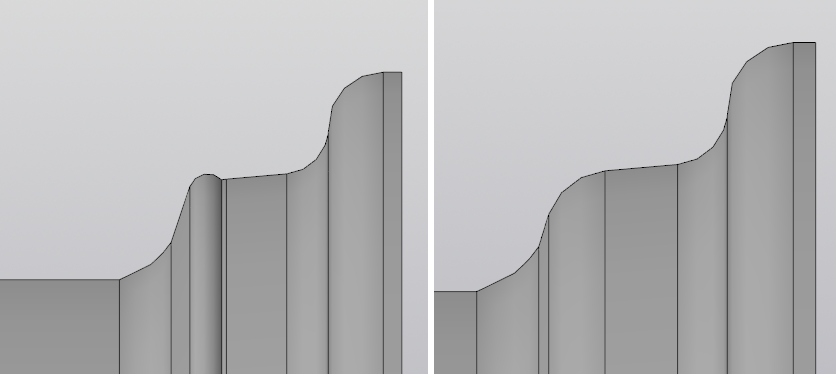


Рисунок 5.8 ‒ Демонстрация отрисовки хампа типа H

## 5.2 Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.12 проведено модульное тестирование [18], проверялись открытые поля и методы.

На рисунке 5.9 представлен результат тестирования класса ArcSegment с помощью вспомогательного класса ArcSegmentTests. Описание тестовых случаев в приложения А (таблица А.1).

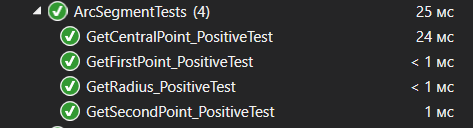


Рисунок 5.9 – Результат тестирования класса ArcSegment

На рисунке 5.10 представлен результат тестирования класса LineSegment с помощью вспомогательного класса LineSegmentTests. Описание тестовых случаев в приложения А (таблица А.2).

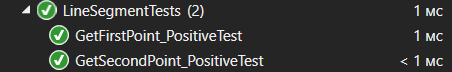


Рисунок 5.10 – Результат тестирования класса LineSegment

На рисунке 5.11 представлен результат тестирования класса AvailableParameters с помощью вспомогательного класса AvailableParametersTests. Описание тестовых случаев в приложения А (таблица А.3).

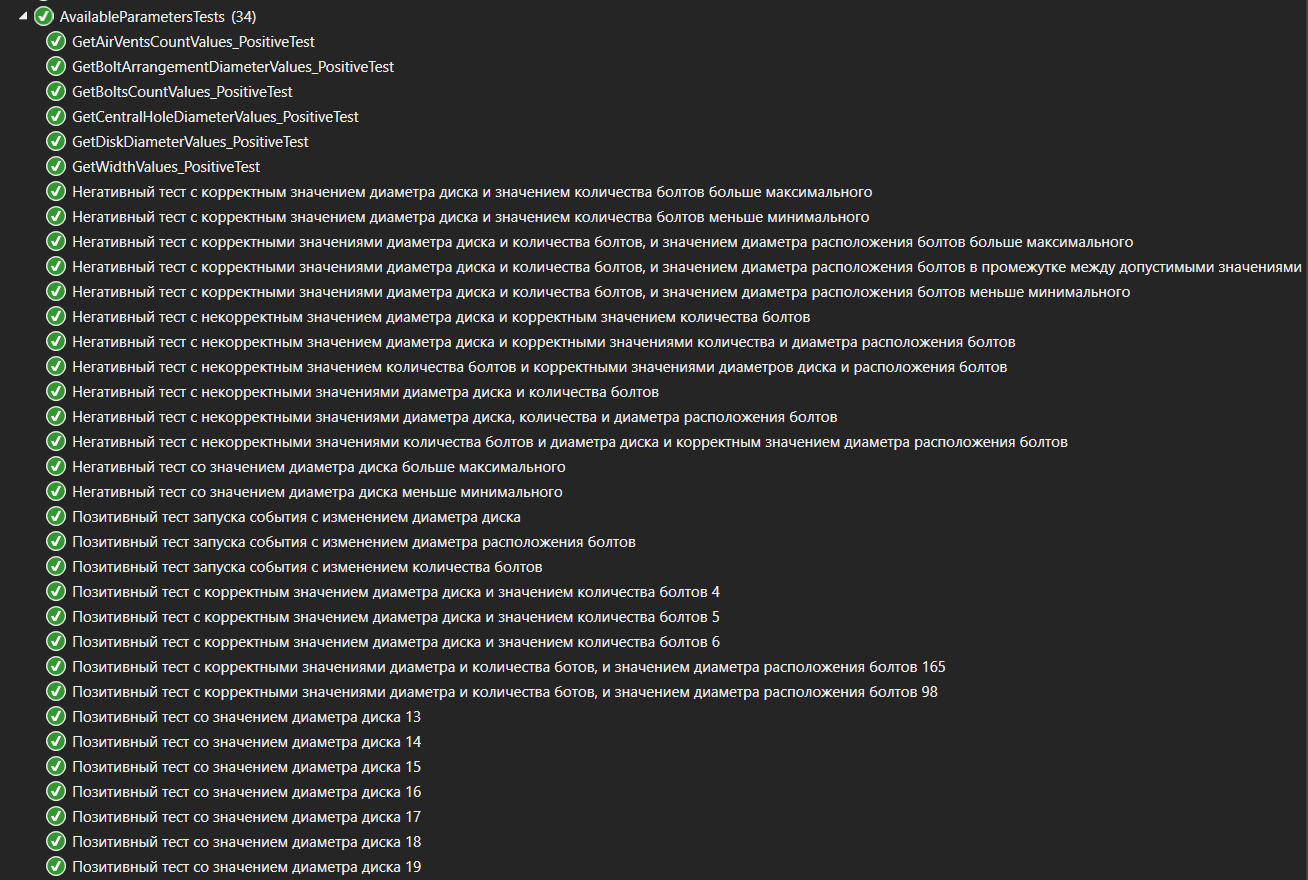


Рисунок 5.11 – Результат тестирования класса AvailableParameters

На рисунке 5.12 представлен результат тестирования класса DiskParams с помощью вспомогательного класса DiskParamsTests. Описание тестовых случаев в приложения А (таблица А.4).

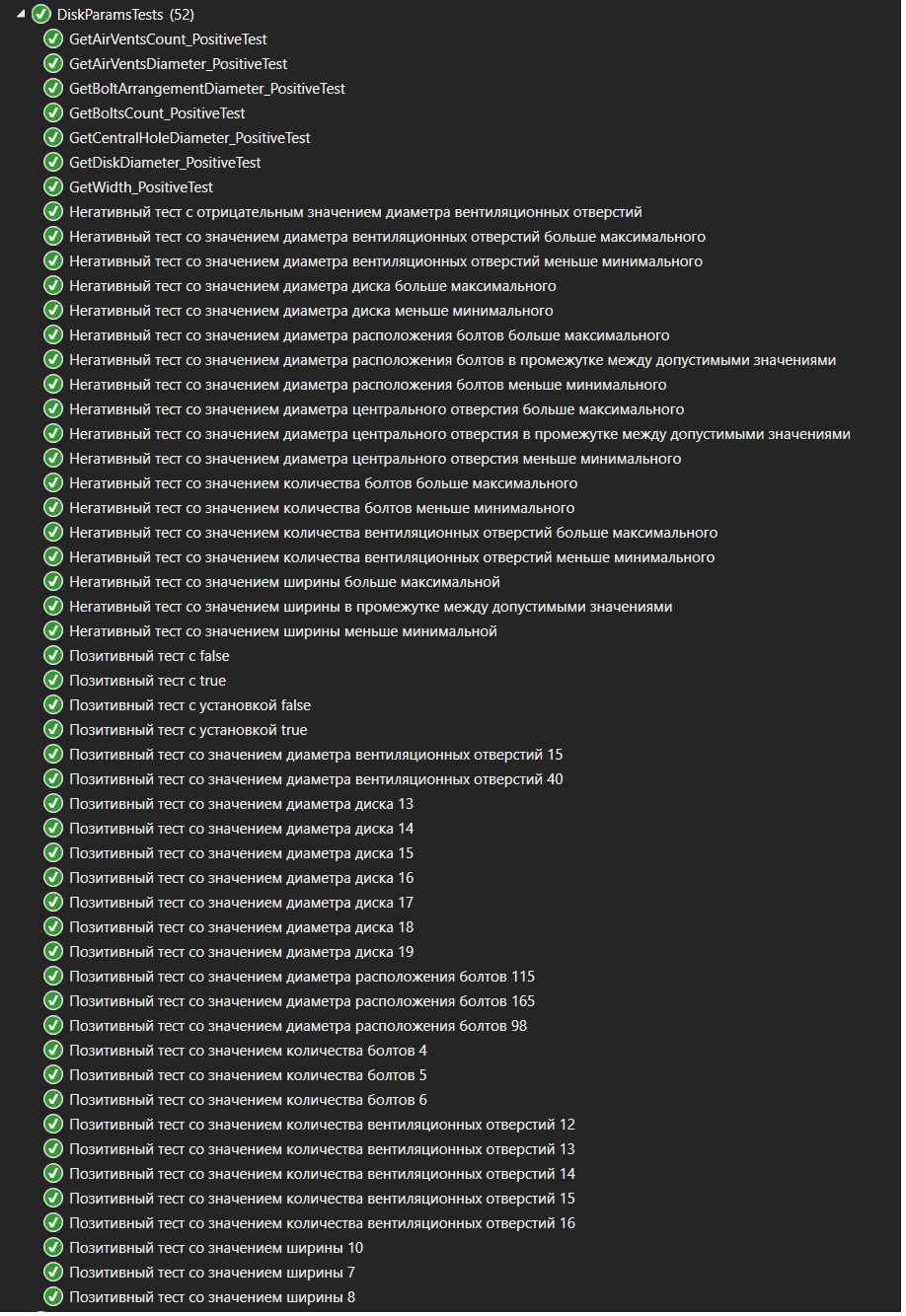


Рисунок 5.12 – Результат тестирования класса DiskParams

На рисунке 5.13 представлен результат тестирования класса DoubleExtensions с помощью вспомогательного класса DoubleExtensionsTests. Описание тестовых случаев в приложения А (таблица А.5).

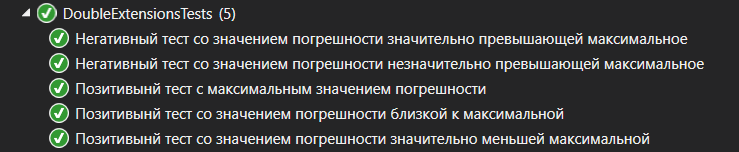


Рисунок 5.13 – Результат тестирования класса DoubleExtensions

На рисунке 5.14 представлен результат тестирования класса ValuesChangeEventArgs с помощью вспомогательного класса ValuesChangeEventArgsTests. Описание тестовых случаев в приложения А (таблица А.6).

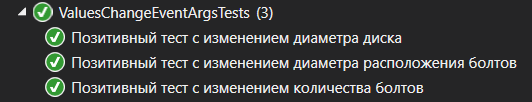


Рисунок 5.14 – Результат тестирования класса ValuesChangeEventArgs

На рисунке 5.15 представлен результат тестирования класса ValuesRelation с помощью вспомогательного класса ValuesRelationTests. Описание тестовых случаев в приложения А (таблица А.7).

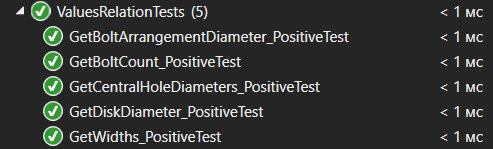


Рисунок 5.15 – Результат тестирования класса ValuesRelation

На рисунке 5.16 представлен результат тестирования класса ParamsValidator с помощью вспомогательного класса ParamsValidatorTests. Описание тестовых случаев в приложения А (таблица А.8).

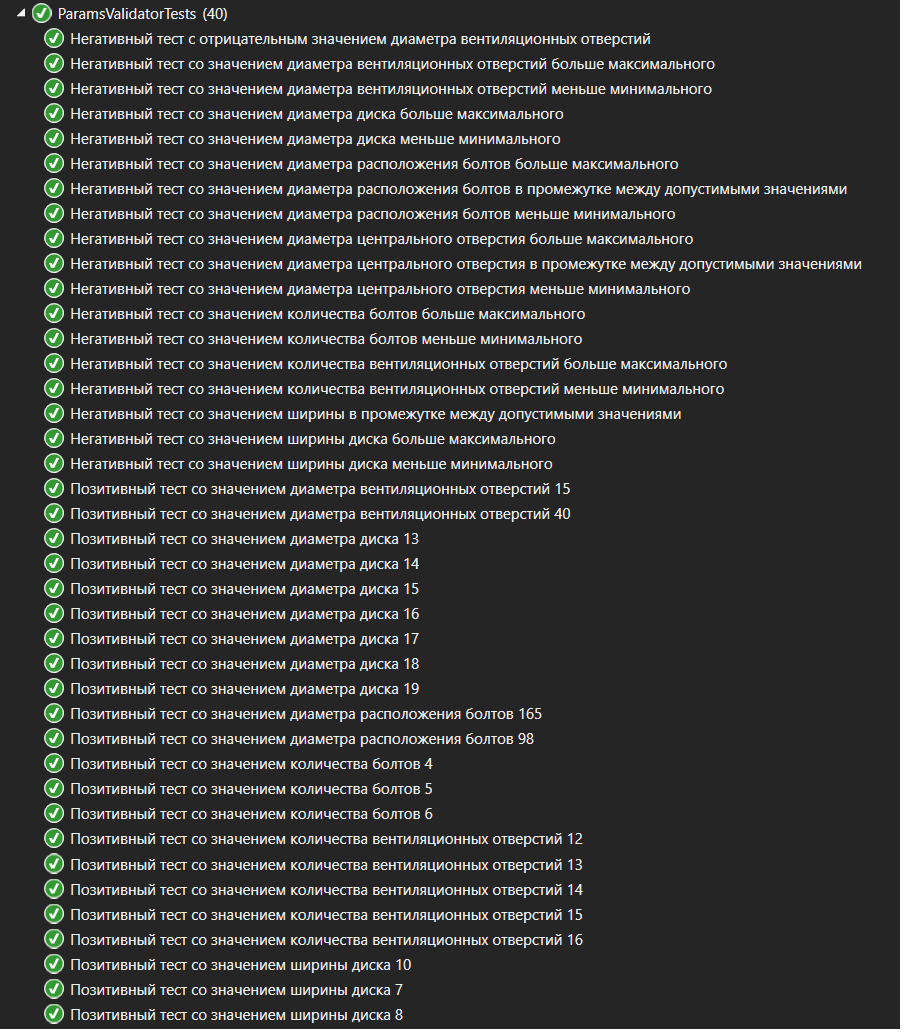


Рисунок 5.16 – Результат тестирования класса ParamsValidator

## 5.3 Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование [19]. Тестирование производилось на ПК со следующей конфигурацией:

* Операционная система Microsoft Windows 10, 64-битная версия;
* Процессор: Intel Core i5-7300HQ 2.5GHz
* ОЗУ: 6 GB
* Графическая система: NVIDIA GeForce 1050

На рисунке 5.17 для проведения нагрузочного тестирования был добавлен секундомер («Stopwatch»), который засекал время от начала построения, с каждым успешным построением фигуры («\_builder.Build(\_diskParams)») производилась запись результатов в текстовый файл «log.txt».

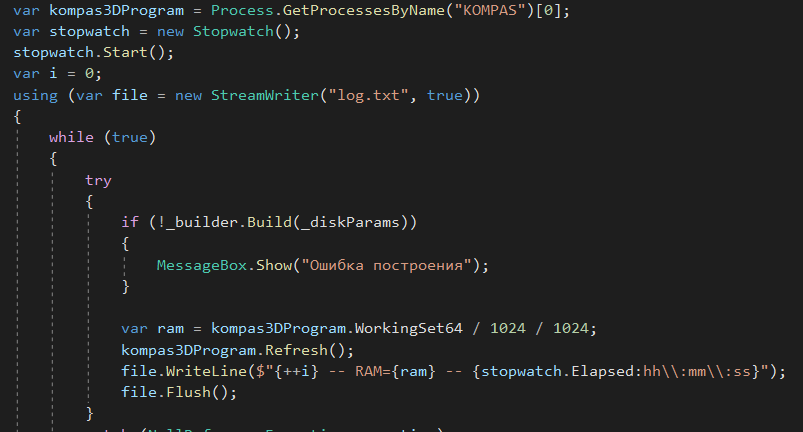


Рисунок 5.17 – Зацикливание перестроения фигуры

Максимальные параметры тестируемой модели диска:

* диаметр диска 482.6 мм (19 дюймов);
* количество болтов 5 шт;
* диаметр расположения болтов 114.3 мм;
* ширина диска 215.9 мм (8.5 дюймов);
* диаметр центрального отверстия 71.6 мм;
* количество вентиляционных отверстий 20 шт;
* диаметр вентиляционных отверстий 40 мм.

Минимальные параметры тестируемой модели диска:

* диаметр диска 330.2 мм (13 дюймов);
* количество болтов 4 шт;
* диаметр расположения болтов 98 мм;
* ширина диска 127 мм (5 дюймов);
* диаметр центрального отверстия 58.6 мм;
* количество вентиляционных отверстий 10 шт;
* диаметр вентиляционных отверстий 15 мм.

Продолжительность нагрузочных тестов с минимальными и максимальными параметрами модели составила 21 мин. 47 сек. и 8 мин. 29 сек. соответственно. Окончанием тестирования считалось аварийное завершение работы программы Компас3D.

Графики количества построенных деталей относительно времени работы плагина изображены на рисунке 5.18. Где Y ‒ количество построенных деталей, а Х ‒ время в минутах. Голубой график ‒ зацикленное построение фигуры с максимальным диаметром диска и максимальными допустимыми параметрами. Желтый график ‒ зацикленное построение фигуры с минимальными диаметром и с минимальными допустимыми параметрами.

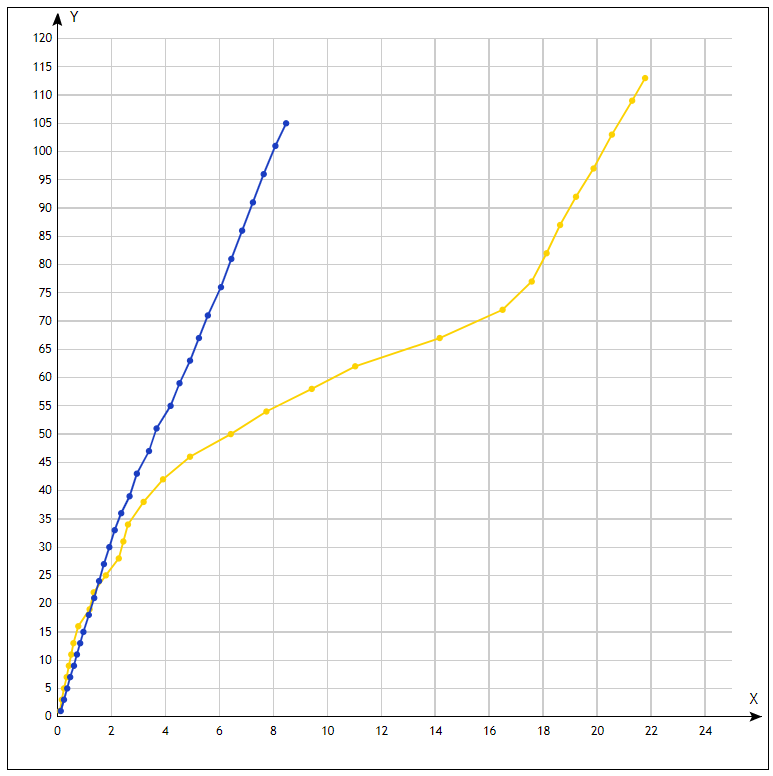


Рисунок 5.18 – График зависимости количества построенных фигур   
от времени (в минутах)

Потребление ОЗУ резко росло в обоих случаях, примерно, до двух минут. Пиковое значение для минимальных параметров модели составило 3486 МБ, а для максимальных параметров 3243 МБ. После достижении двух минут, потребление ОЗУ периодически уменьшалось и увеличивалось. При этом, в диспетчере задач было видно, что программа Компас3D периодически не отвечала, но по прошествии непродолжительного времени снова начинала работать. В этот момент и происходило снижение потребляемой ОЗУ.

Графики потребляемой оперативной памяти программы Компас3D относительно времени работы плагина изображены на рисунке 5.19. Где Y ‒ количество используемой оперативной памяти, в мегабайтах, а Х ‒ время в минутах. Голубой график ‒ зацикленное построение фигуры с максимальным диаметром диска и максимальными допустимыми параметрами. Желтый график ‒ зацикленное построение фигуры с минимальными диаметром и с минимальными допустимыми параметрами.

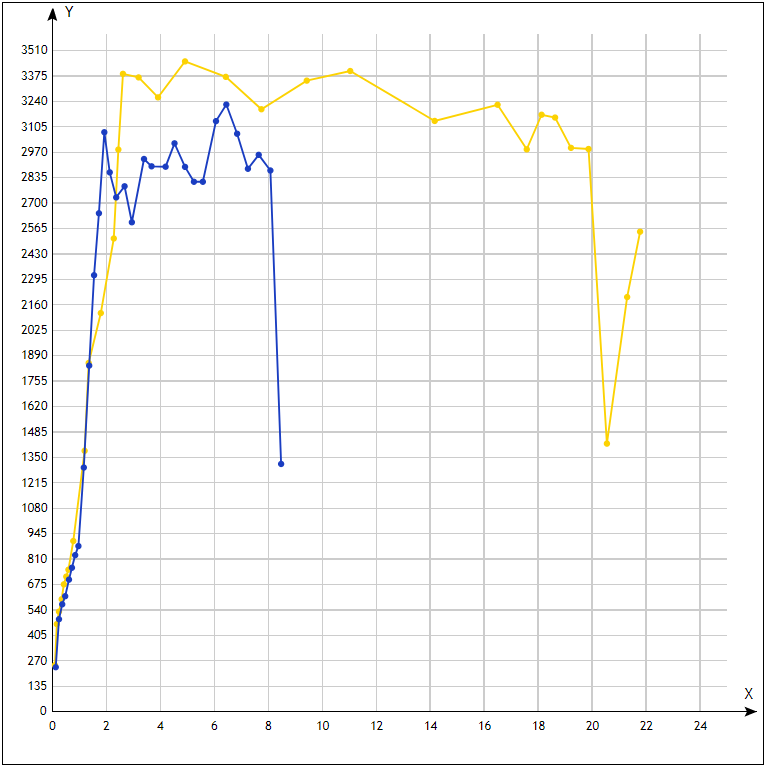


Рисунок 5.19 – График зависимости потребляемой оперативной памяти программой Компас3D (в мегабайтах) от времени (в минутах)

Значения загрузки ЦП программой Компас3D на всем временном интервале, в обоих тестовых случаях, колебались незначительно. Данные колебания связаны с тем, что программа Компас3D периодически не отвечала.

Графики загрузки ЦП программой Компас3D относительно времени работы плагина изображены на рисунке 5.20. Где Y ‒ загрузка ЦП в процентах, а Х ‒ время в минутах. Голубой график ‒ зацикленное построение фигуры с максимальным диаметром диска и максимальными допустимыми параметрами. Желтый график ‒ зацикленное построение фигуры с минимальными диаметром и с минимальными допустимыми параметрами.

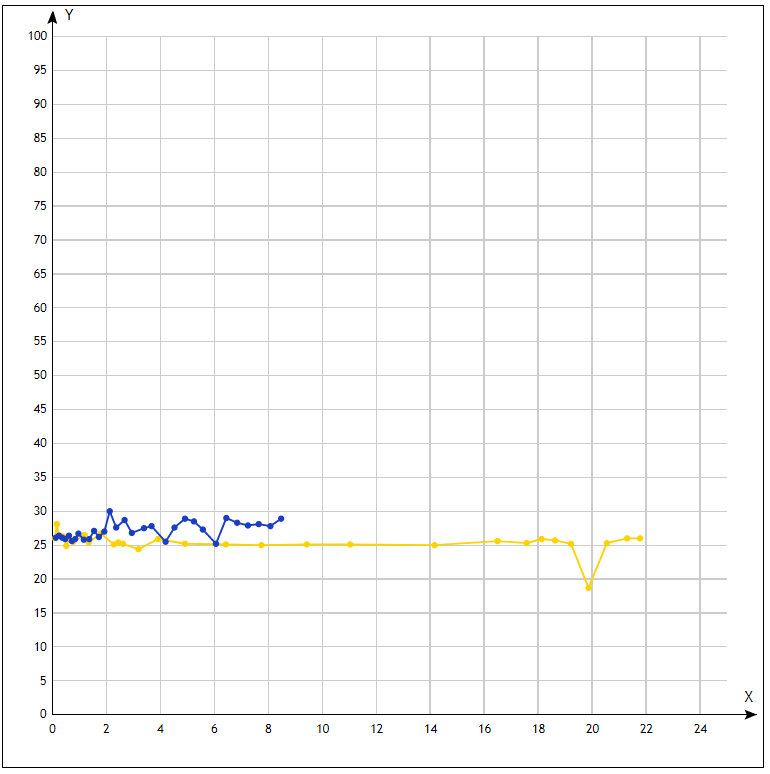


Рисунок 5.20 – График зависимости загрузки ЦП программой Компас3D (в процентах) от времени (в минутах)

Из рисунка 5.20 также видно, что тестирование плагина с максимальными параметрами модели требует больше ресурсов ЦП.

ПК, на котором проводились нагрузочное тестирование имеет небольшое количество оперативной памяти. А количество операций, необходимых для построения модели довольно большое. Из-за этого, при увеличении количества деталей, быстро наступает аварийное завершение работы программы Компас3D. Количество построенных деталей при минимальных параметрах равняется 113, при максимальных 105.

# 5 Заключение

В ходе выполнения лабораторных работ были изучены основные этапы проектирования программного продукта, изучена предметная область объекта проектирования, также было изучено API системы автоматизированного проектирования Компас3D.

В результате полученные знания были применены для реализации плагина для автоматизации построения модели «Автомобильный диск» в системе Компас3D, проведены функциональные, модульные и нагрузочные тесты.

# Список используемых источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования. [Электронный ресурс]. – Режим доступа: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 24.02.2020).
2. КОМПАС (САПР). [Электронный ресурс]. − Режим доступа: https://kompas.ru/ (дата обращения: 24.02.2020).
3. Visual Studio: Microsoft. [Электронный ресурс]. – Режим доступа: <https://visualstudio.microsoft.com/ru/> (дата обращения: 24.02.2020).
4. О штампованных колесных дисках.[Электронный ресурс]. – Режим доступа: <https://kolesa.guru/diski/stampovannye.html> (дата обращения: 18.03.2020).
5. Колеса неразборные: ГОСТ Р 53824-2010.[Электронный ресурс]. – Режим доступа: http://gostrf.com/normadata/1/4293814/4293814005.htm (дата обращения: 18.03.2020).
6. .NET Framework 4.7.2: Поддержка Microsoft. [Электронный ресурс] ‒ Режим доступа: <https://support.microsoft.com/ru-ru/help/4054530/microsoft-net-framework-4-7-2-offline-installer-for-windows> (дата обращения 18.03.2020).
7. NUnit. [Электронный ресурс]. – Режим доступа: <https://nunit.org/> (дата обращения: 24.03.2020).
8. Обзор Windows Forms: документация Microsoft. [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/windows-forms-overview (дата обращения: 24.03.2020).
9. Что такое API. [Электронный ресурс]. – Режим доступа: <https://ru.stackoverflow.com/questions/942216/Что-такое-api-понятным-языком> (дата обращения: 24.03.2020).
10. КОМПАС-3D V10 на 100 %. [Электронный ресурс] ‒ Режим доступа: https://it.wikireading.ru/23741 (дата обращения 24.03.2020).
11. SolidWorks. [Электронный ресурс]. ‒ Режим доступа: https://www.solidworks.com/ru/product/solidworks-3d-cad (дата обращения 01.03.2020).
12. UML. [Электронный ресурс]. – Режим доступа: <http://www.uml.org/> (дата обращения: 13.04.2020).
13. Новые технологии в программировании: учебное пособие / А.А. Калентьев, Д. В. Гарайс, А. Е. Горяинов. – Томск, 2014. − ­ 176 стр.
14. Маркировка автомобильных дисков. [Электронный ресурс]. – Режим доступа: http://www.uazbuka.ru/disc\_marker.html (дата обращения: 13.04.2020).
15. Теоретические сведения о диаграммах классов. [Электронный ресурс]. – Режим доступа: https://flexberry.github.io/ru/gpg\_class-diagram.html (дата обращения: 5.05.2020).
16. EventHandler Делегат. [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/system.eventhandler?view=netframework-4.8 (дата обращения: 13.04.2020).
17. Функциональное тестирование. [Электронный ресурс]. – Режим доступа: <https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 5.05.2020).
18. Модульное тестирование. [Электронный ресурс]. – Режим доступа: <http://espressocode.top/unit-testing-software-testing/> (дата обращения: 13.05.2020).
19. Нагрузочное тестирование. [Электронный ресурс]. – Режим доступа: <https://daglab.ru/nagruzochnoe-testirovanie-programmnogo-obespechenija/> (дата обращения: 15.05.2020).

# Приложение А

(справочное)

Все таблицы, представленные ниже, являются описанием вспомогательных классов для проведения модульного тестирования. Каждый метод в классе является тестовым случаем. В каждой таблице описаны поля, методы класса (тестовые случаи) и их описание.

Таблица А.1 – Класс ArcSegmentTests

|  |  |
| --- | --- |
| Наименование | Описание |
| \_arcSegment: ArcSegment | Параметры дуги |
| SetArcSegment() | Инициализировать дугу |
| GetCentralPoint\_PositiveTest() | Позитивный тест геттера CentralPoint |
| GetRadius\_PositiveTest() | Позитивный тест геттера Radius |
| GetFirstPoint\_PositiveTest() | Позитивный тест геттера FirstPoint |
| GetSecondPoint\_PositiveTest() | Позитивный тест геттера SecondPoint |

Таблица А.2 ‒ Класс LineSegmentTests

|  |  |
| --- | --- |
| Наименование | Описание |
| \_lineSegment: LineSegment | Параметры отрезка |
| SetLineSegment() | Инициализировать отрезок |
| GetFirstPoint\_PositiveTest() | Позитивный тест геттера FirstPoint |
| GetSecondPoint\_PositiveTest() | Позитивный тест геттера SecondPoint |

Таблица А.3 – Класс AvailableParametersTests

|  |  |
| --- | --- |
| Наименование | Описание |
| \_availableParameters: AvailableParameters | Доступные параметры |
| \_diskParams: DiskParams | Параметры диска |
| SetAvailableParameters() | Создать экземпляр класса DiskParams и инициализировать доступные параметры диска |
| GetDiskDiameterValues\_PositiveTest() | Позитивный тест геттера DiskDiameterValues |
| GetWidthValues\_PositiveTest() | Позитивный тест геттера WidthValues |
| GetBoltsCountValues\_PositiveTest() | Позитивный тест геттера BoltsCountValues |

Продолжение таблицы А.3

|  |  |
| --- | --- |
| Наименование | Описание |
| GetBoltArrangementDiameter- Values\_PositiveTest() | Позитивный тест геттера BoltArrangementDiameterValues |
| GetCentralHoleDiameter-Values\_PositiveTest() | Позитивный тест геттера CentralHoleDiameterValues |
| GetAirVentsCount- Values\_PositiveTest() | Позитивный тест геттера AirVentsCountValues |
| ChangeDiskDiameterDependent-Quantities\_NegativeTest(int diskDiameter) | Негативный тест метода ChangeDiskDiameterDependentQuantities |
| ChangeDiskDiameterDependent-Quantities\_PositiveTest(int diskDiameter) | Позитивный тест метода ChangeDiskDiameterDependentQuantities |
| ChangeBoltsCountDependent-Quantities\_NegativeTest(int diskDiameter, int boltsCount) | Негативный тест метода ChangeBoltsCountDependentQuantities |
| ChangeBoltsCountDependent-Quantities\_PositiveTes(int diskDiameter, int boltsCount) | Позитивный тест метода ChangeBoltsCountDependentQuantities |
| ChangeBoltArrangementDiameter-DependentQuantities\_NegativeTest  (int diskDiameter, int boltsCount, double boltArrangementDiameter) | Негативный тест метода ChangeBoltArrangementDiameter-DependentQuantities |
| ChangeBoltArrangementDiameter-DependentQuantities\_PositiveTest  (int diskDiameter, int boltsCount, double boltArrangementDiameter) | Позитивный тест метода ChangeBoltArrangementDiameter-DependentQuantities |
| ValuesChanged\_WhenCalled-\_IsSuccessfull  (AvailableValuesChangeType expectedChangeType) | Позитивный тест запуска события ValuesChanged |

Таблица А.4 – Класс DiskParamsTests

|  |  |
| --- | --- |
| Наименование | Описание |
| \_diskParams: DiskParams | Параметры диска |
| SetParams() | Создать экземпляр класса DiskParams |
| SetHumpFlag\_PositiveTest(bool flag) | Позитивный тест сеттера свойства HumpFlag |
| GetHumpFlag\_PositiveTest(bool flag) | Позитивный тест геттера свойства HumpFlag |
| SetDiskDiameter\_NegativeTest(int diskDiameter) | Негативный тест сеттера свойства DiskDiameter |
| SetDiskDiameter\_PositiveTest(int diskDiameter) | Позитивный тест сеттера свойства DiskDiameter |
| GetDiskDiameter\_PositiveTest() | Позитивный тест геттера DiskDiameter |
| SetWidth\_NegativeTest(double width) | Негативный тест сеттера свойства Width |
| SetWidth\_PositiveTest(double width) | Позитивный тест сеттера свойства Width |
| GetWidth\_PositiveTest() | Позитивный тест геттера Width |
| SetBoltsCount\_NegativeTest(int boltsCount) | Негативный тест сеттера свойства BoltsCount |
| SetBoltsCount\_PositiveTest(int boltsCount) | Позитивный тест сеттера свойства BoltsCount |
| GetBoltsCount\_PositiveTest() | Позитивный тест геттера BoltsCount |
| SetBoltArrangementDiameter\_NegativeTest  (double boltArrangementDiameter) | Негативный тест сеттера свойства BoltArrangementDiameter |
| SetBoltArrangementDiameter\_PositiveTest  (double boltArrangementDiameter) | Позитивный тест сеттера свойства BoltArrangementDiameter |
| GetBoltArrangementDiameter\_PositiveTest() | Позитивный тест геттера BoltArrangementDiameter |
| SetCentralHoleDiameter\_NegativeTest  (double centralHoleDiameter) | Негативный тест сеттера свойства CentralHoleDiameter |
| SetCentralHoleDiameter\_PositiveTest  (double centralHoleDiameter) | Позитивный тест сеттера свойства CentralHoleDiameter |
| GetCentralHoleDiameter\_PositiveTest() | Позитивный тест геттера CentralHoleDiameter |
| SetAirVentsCount\_NegativeTest(int airVentsCount) | Негативный тест сеттера свойства AirVentsCount |
| SetAirVentsCount\_PositiveTest(int airVentsCount) | Позитивный тест сеттера свойства AirVentsCount |

Продолжение таблицы А.4

|  |  |
| --- | --- |
| Наименование | Описание |
| GetAirVentsCount\_PositiveTest() | Позитивный тест геттера AirVentsCount |
| SetAirVentsDiameter\_NegativeTest  (decimal airVentsDiameter) | Негативный тест сеттера свойства AirVentsDiameter |
| SetAirVentsDiameter\_PositiveTest(decimal airVentsDiameter) | Позитивный тест сеттера свойства AirVentsDiameter |
| GetAirVentsDiameter\_PositiveTest() | Позитивный тест геттера AirVentsDiameter |

Таблица А.5 – Класс DoubleExtensions

|  |  |
| --- | --- |
| Наименование | Описание |
| EqualTor\_NegativeTest(double epsilon) | Негативный тест метода EqualTo |
| EqualTor\_PositiveTest(double epsilon) | Позитивный тест метода EqualTo |

Таблица А.6 – Класс ValuesChangeEventArgs

|  |  |
| --- | --- |
| Наименование | Описание |
| GetType\_PositiveTest | Позитивный тест геттера Type |

Таблица А.7 – Класс ValuesRelation

|  |  |
| --- | --- |
| Наименование | Описание |
| \_valuesRelation: ValuesRelation | Зависимость доступных значений |
| SetValuesRelation() | Инициализировать зависимость значений |
| GetDiskDiameter\_PositiveTest() | Позитивный тест геттера DiskDiameter |
| GetWidths\_PositiveTest() | Позитивный тест геттера Widths |
| GetBoltCount\_PositiveTest() | Позитивный тест геттера BoltCount |
| GetBoltArrangementDiameter-\_PositiveTest() | Позитивный тест геттера BoltArrangementDiameter |
| GetCentralHoleDiameters \_PositiveTest() | Позитивный тест геттера CentralHoleDiameters |

Таблица А.8 – Класс ParamsValidator

|  |  |
| --- | --- |
| Наименование | Описание |
| \_availableParameters: AvailableParameters | Доступные параметры |
| SetAvailableParameters() | Создать экземпляр класса DiskParams и инициализировать доступные парматетры |
| IsValidDiskDiameter\_NegativeTest(int diskDiameter) | Негативный тест метода IsValidDiskDiameter |
| IsValidDiskDiameter\_PositiveTest(int diskDiameter) | Позитивный тест метода IsValidDiskDiameter |
| IsValidWidth\_NegativeTest(double width) | Негативный тест метода IsValidWidth |
| IsValidWidth\_PositiveTest(double width) | Позитивный тест метода IsValidWidth |
| IsValidBoltsCount\_NegativeTest(int boltsCount) | Негативный тест метода IsValidBoltsCount |
| IsValidBoltsCount\_PositiveTest(int boltsCount) | Позитивный тест метода IsValidBoltsCount |
| IsValidBoltArrangementDiameter-\_NegativeTest  (double boltArrangementDiameter) | Негативный тест метода IsValidBoltArrangementDiameter |
| IsValidBoltArrangementDiameter-\_PositiveTest  (double boltArrangementDiameter) | Позитивный тест метода IsValidBoltArrangementDiameter |
| IsValidCentralHoleDiameter-\_NegativeTest  (double centralHoleDiameter) | Негативный тест метода IsValidCentralHoleDiameter |
| IsValidCentralHoleDiameter-\_PositiveTest  (double centralHoleDiameter) | Позитивный тест метода IsValidCentralHoleDiameter |
| IsValidAirVentsCount-\_NegativeTest  (double airVentsCount) | Негативный тест метода IsValidAirVentsCount |
| IsValidAirVentsCount-\_PositiveTest(double airVentsCount) | Позитивный тест метода IsValidAirVentsCount |
| IsValidAirVentsDiameter-\_NegativeTest  (decimal airVentsDiameter) | Негативный тест метода IsValidAirVentsDiameter |

Продолжение таблицы А.8

|  |  |
| --- | --- |
| Наименование | Описание |
| IsValidAirVentsDiameter\_PositiveTest  (decimal airVentsDiameter) | Позитивный тест метода IsValidAirVentsDiameter |