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)

РАЗРАБОТКА БИБЛИОТЕКИ «ВТУЛКА АМОРТИЗАТОРА» ДЛЯ САПР «КОМПАС-3D»

Пояснительная записка к проекту

по дисциплине «Основы разработки САПР»

Выполнил:

студент гр. 586-1

\_\_\_\_\_\_\_С.Д. Дубинин

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Руководитель:

к.т.н, доцент каф. КСУП

\_\_\_\_\_\_\_ А.А. Калентьев

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Томск 2020

**Реферат**

Реферат: 27 страниц, 18 рисунков, 1 таблица, 11 использованных источников.

КОМПАС-3D, БИБЛИОТЕКА, ВТУЛКА АМОРТИЗАТОРА, РАЗРАБОТКА, ПРОЕКТ, ТЕСТИРОВАНИЕ

В ходе работы были изучены основы разработки программного обеспечения на примере создания библиотеки для построения детали «Втулка амортизатора» для САПР «КОМПАС-3D V 16».

В работе приведен обзор аналогов разрабатываемой библиотеки, рассмотрена предметная область и предмет исследования. После утверждения технического задания был составлен проект системы, включающий в себя диаграммы классов, диаграммы вариантов использования и рассмотрен макет графического интерфейса пользователя. То же проделано и после добавления новых возможностей в программу. Тестирование производилось посредством разработанных к проекту модульных тестов.

Отчет выполнен в программе Microsoft Word. Библиотека написана на языке программирования C# для платформы .NET Framework 4.6.2 в IDE Visual Studio 2019 Сommunity. В разработке использовалась система контроля версий Git.

**Содержание**

[Введение 4](#_Toc39487829)

[2 Постановка и анализ задачи 5](#_Toc39487830)

[2.1 Описание САПР 5](#_Toc39487831)

[2.2 Обзор аналогов 6](#_Toc39487832)

[2.2.1 Программа автоматического построения 3D моделей и разверток по заданным значениям в AutoCAD «Лекало». Расчет и построение механических передач 6](#_Toc39487833)

[2.2.2 Плагин «Fasteners» для программы FreeCAD 7](#_Toc39487834)

[2.3 Описание предмета проектирования 7](#_Toc39487835)

[3 Описание реализации 9](#_Toc39487836)

[3.1 Диаграмма вариантов использования 9](#_Toc39487837)

[3.2 Диаграмма классов 10](#_Toc39487838)

[3.3 Макет пользовательского интерфейса 12](#_Toc39487839)

[4 Описание программы для пользователя 14](#_Toc39487840)

[5 Тестирование библиотеки 17](#_Toc39487841)

[5.1 Функциональное тестирование 17](#_Toc39487842)

[5.2 Модульное тестирование 19](#_Toc39487843)

[5.3 Нагрузочное тестирование 22](#_Toc39487844)

[Заключение 25](#_Toc39487845)

[Список использованных источников 26](#_Toc39487846)

# **Введение**

Ни один набор библиотек к графической системе не может охватить всё множество различных направлений и отраслей промышленности и в полной мере удовлетворить требования всех категорий пользователей. Спектр задач конструкторско-технологической подготовки производства ста настолько широким, что решение некоторых из них как стандартными, так и прикладными средствами, предоставленными компанией-разработчиком программного обеспечения (ПО), может отнимать слишком много времени и быть нерациональным, а в редких случаях – даже невозможным.

Огромный перечень областей применения систем автоматизированного проектирования (САПР) – это первая причина, по которой любая из современных CAD-систем должна быть максимально открытой и обязательно содержать инструменты для создания сторонних библиотек.

В каждой крупной САПР есть свои средства для разработки, которые предоставляются с целью дать возможность разработчикам расширить возможности данной системы под свои конкретные нужды. Данным средством является API – программируемый интерфейс приложения. Это набор готовых средств: классов, процедур, функций, структур и т.д. API позволяет определить возможности, которые предоставляет приложение, при этом абстрагируясь от того, как она реализована.

Расширение возможностей, в основном, подразумевает разработку библиотеки на основе предоставленного API. Библиотекой является независимо компилируемый программный модель, динамически подключаемый к основной программе, предназначенный для расширения или использования ее возможностей [1]. В данном проекте стоит задача разработки библиотеки для построения 3D-модели втулки амортизатора в САПР «КОМПАС-3D».

# **2 Постановка и анализ задачи**

В рамках учебной дисциплины «Основы разработки САПР» требовалось разработать библиотеку в соответствии с техническим заданием. На основе заданных параметров библиотека, взаимодействуя с САПР «КОМПАС-3D», должна строить 3D-модель втулки амортизатора. Также библиотека должна позволять изменять входные параметры втулки. Изменяемые параметры: длина всей втулки, длина верхней части втулки, наружный диаметр втулки, внутренний диаметр втулки, диаметр верхней части втулки, количество отверстий, диаметр отверстий, диаметр расположения отверстий, текст гравировки.

## **2.1 Описание САПР**

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].

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.

## **2.2 Обзор аналогов**

2.2.1 Программа автоматического построения 3D моделей и разверток по заданным значениям в AutoCAD «Лекало». Расчет и построение механических передач

Данная программа позволяет создавать следующие 3D модели в AutoCAD посредством ввода размеров с клавиатуры:

* металлопрокат;
* механические соединения;
* механические передачи;
* элементы гидро- и пнемвоприводов;
* построение конструктивных элементов [2].

На рисунке 2.1 представлен пользовательский интерфейс программы «Лекало» для построения втулки.



Рисунок 2.1 – Пользовательский интерфейс программы «Лекало» для построения втулки

2.2.2 Плагин «Fasteners» для программы FreeCAD

Данный плагин предназначен для моделирования метрических болтов и гаек по стандартам ISO [3].

На рисунке 2.2 представлен пользовательский интерфейс плагина «Fasteners».



Рисунок 2.2 – Пользовательский интерфейс плагина «Fasteners»

## **2.3 Описание предмета проектирования**

Предметом проектирования является втулка амортизатора.

Втулка – это один из элементов механизированной системы, имеющий форму конуса или цилиндра с отверстием для вставки вала или другой детали. Втулки используются для защиты и уменьшения износа промышленных изделий и подвижных узлов. Втулки дают гарантию, что важные части механизмов будут надежно работать. Немаловажно, что использование втулок уменьшает возможность возникновения аварийный ситуаций [4].

На рисунке 2.3 представлен чертеж втулки амортизатора.

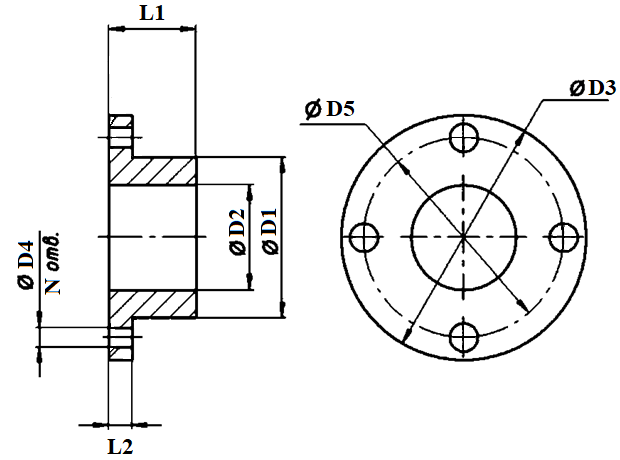


Рисунок 2.3 – Чертеж втулки амортизатора

На рисунке 2.4 представлена 3D-модель втулки амортизатора.

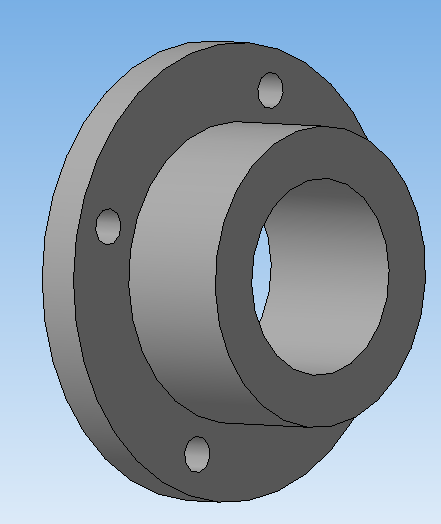


Рисунок 2.4 – 3D-модель втулки амортизатора

Для подобной детали существует ГОСТ 15019-69 – Втулки центрирующие и направляющие для литейных опок. Конструкция и размеры [5].

# **3 Описание реализации**

Для описания архитектуры пользовательского сценария системы был выбран унифицированный язык моделирования (UML) [6]. На основе UML построены диаграммы вариантов использования и диаграммы классов.

Также одним из важных моментов при проектировании программы является внешний вид программы. Внешний вид программы должен максимально отражать функциональные особенности программы, чтобы пользователь мог за минимальное время освоить программу и пользоваться ею.

## **3.1 Диаграмма вариантов использования**

Диаграмма вариантов использования отражает возможный выбор действий (выбора состояния) пользователя внутри системы [6].

На рисунке 3.1 представлена диаграмма вариантов использования, принятая на этапе проектирования.

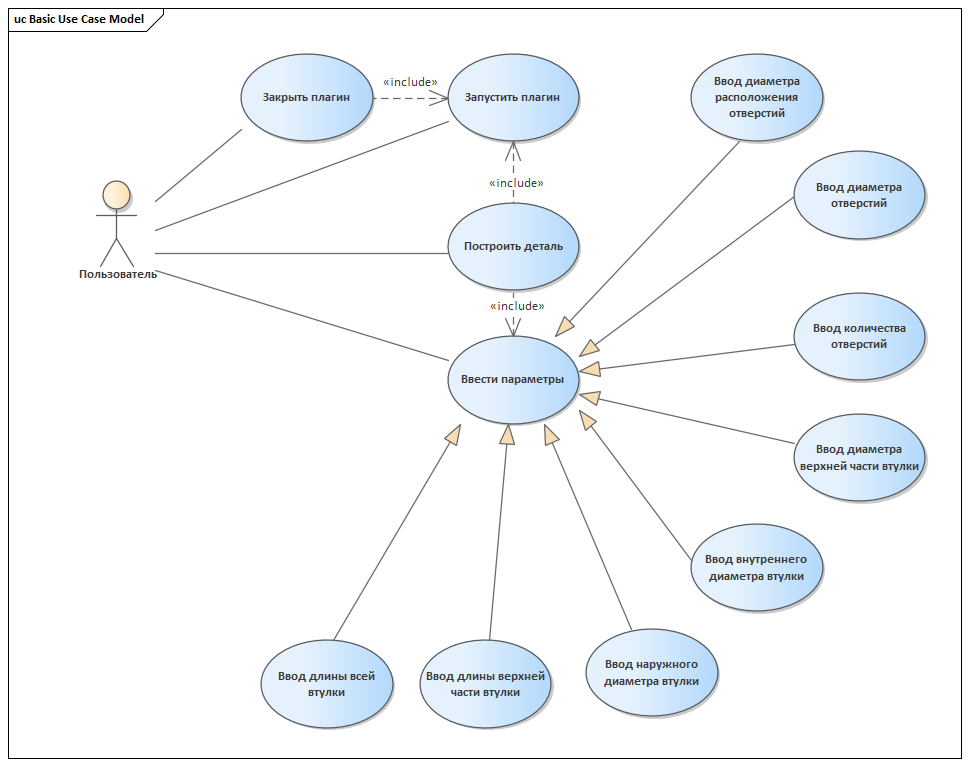


Рисунок 3.1 – Начальная диаграмма вариантов использования

В итоге диаграмма вариантов использования приобрела вид, представленный на рисунке 3.2. Это обусловлено тем, что в программу была добавлена дополнительная функциональность.

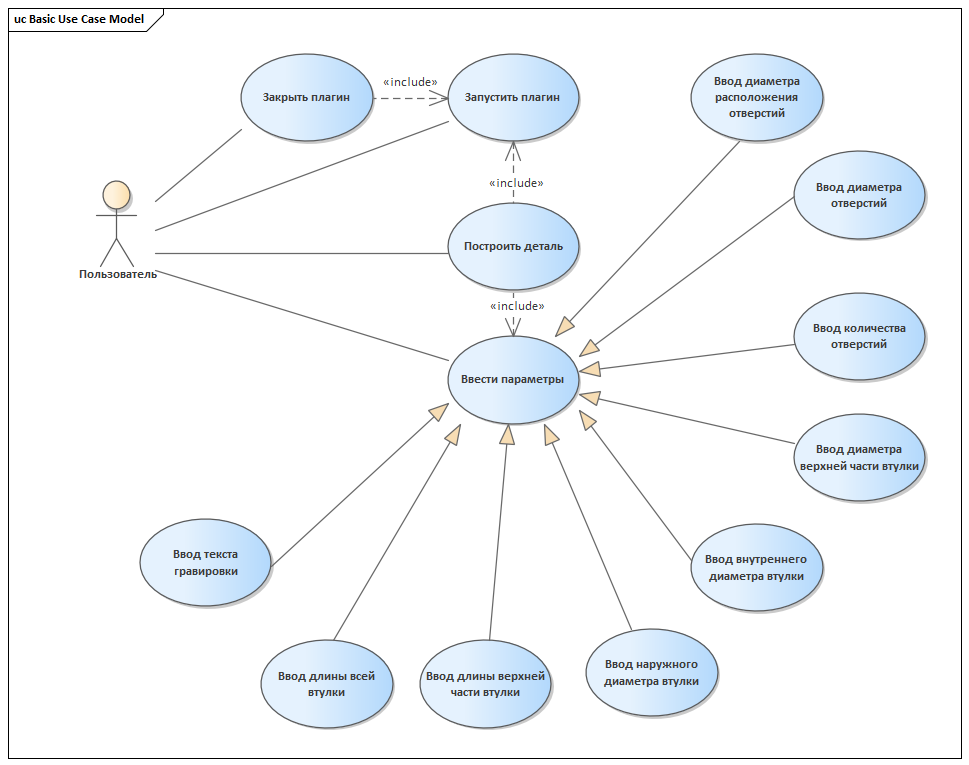


Рисунок 3.2 – Конечная диаграмма вариантов использования

## **3.2 Диаграмма классов**

UML диаграмма классов представляет собой графическую интерпретацию классов системы, их атрибутов, методов и взаимосвязей между ними [6]. На рисунке 3.3 представлена диаграмма классов разрабатываемой библиотеки, принятая на этапе проектирования.

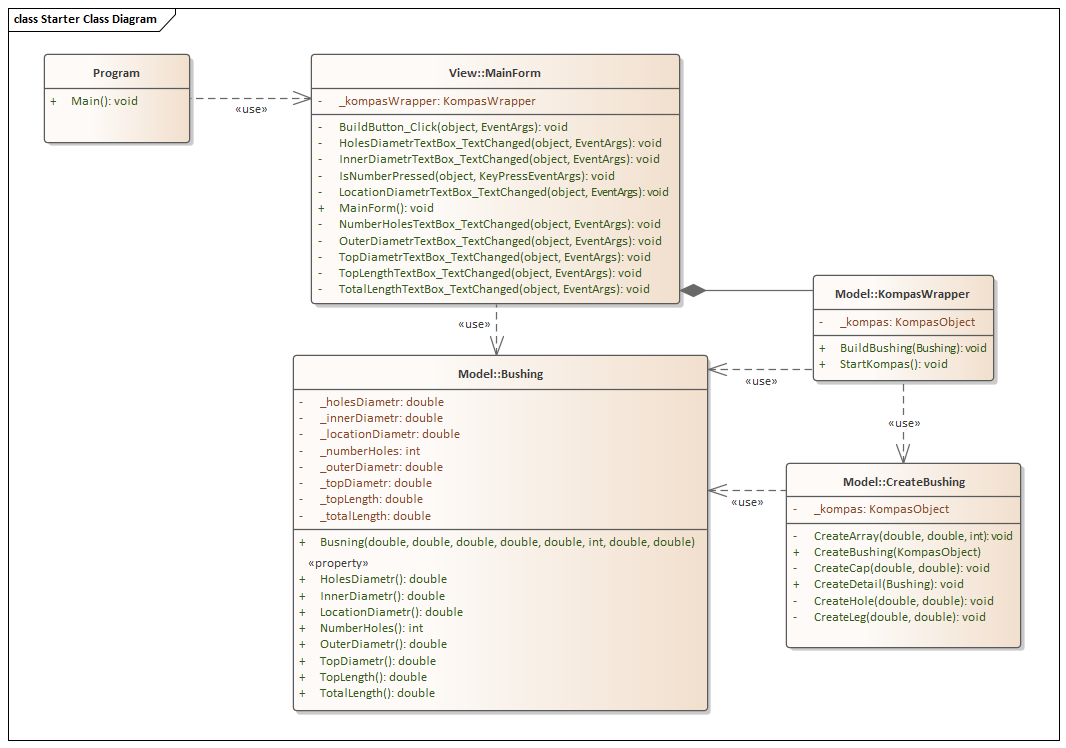
****

Рисунок 3.3 – Начальная диаграмма классов

Ввиду внесенных в программу изменений, диаграмма классов также претерпела некоторые изменения и приобрела вид, представленный на рисунке 3.4.

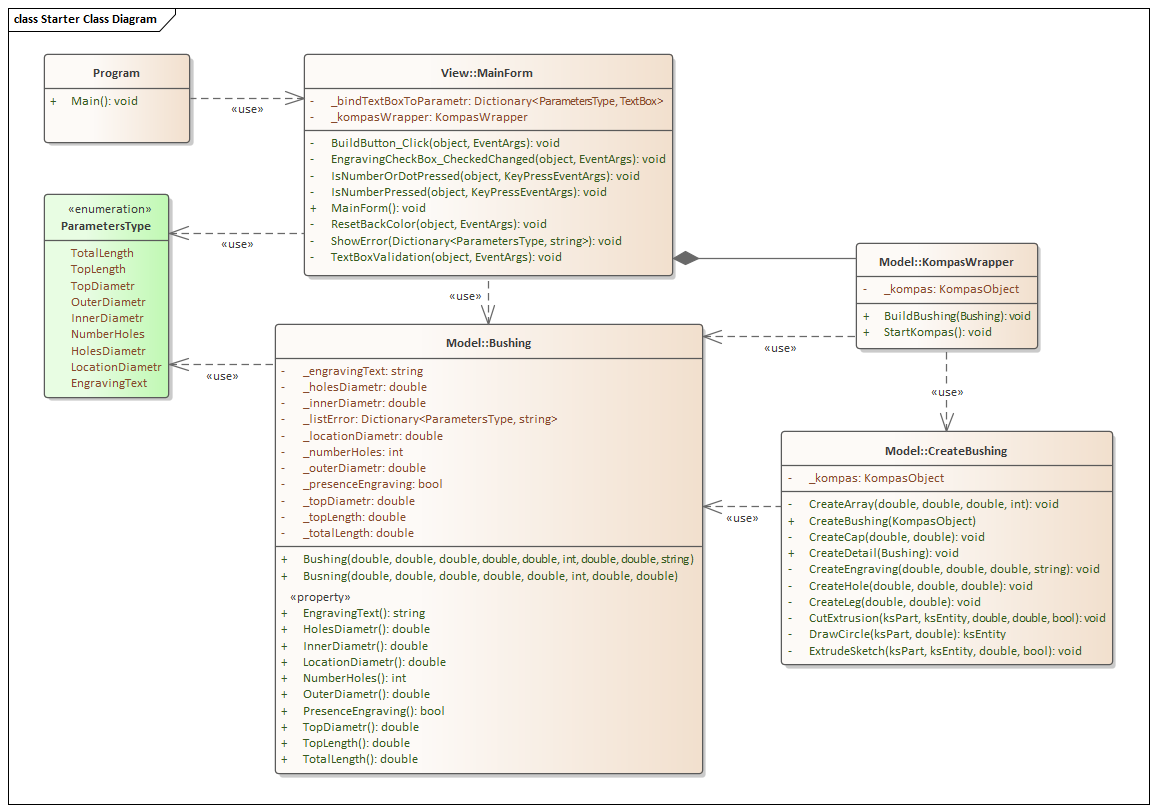


Рисунок 3.4 – Конечная диаграмма классов

Класс Bushing предназначен для описания параметров втулки и их валидации, класс KompasWrapper предназначен для взаимодействия с САПР «Компас-3D», класс CreateBushing предназначен для построения втулки, класс MainForm предназначен для описания пользовательского интерфейса. Было добавлено перечисление ParametersType, хранящее в себе названия параметров втулки, для более удобного отображения ошибок при некорректном вводе параметров пользователем. Также были добавлены дополнительные поля, свойства и методы в ранее созданные классы.

## **3.3 Макет пользовательского интерфейса**

Пользовательский интерфейс – это набор программных и аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером. [7].

На рисунке 3.5 представлен начальный макет интерфейса библиотеки, принятый на этапе проектирования.

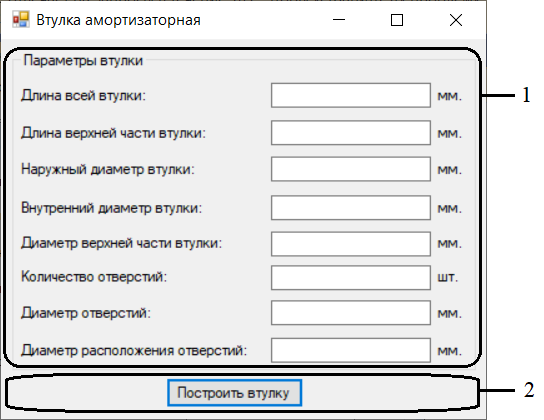
****

Рисунок 3.5 – Начальный макет пользовательского интерфейса библиотеки (1 – область для ввода параметров втулки, 2 – область управления построением)

На рисунке 3.6 представлен конечный макет пользовательского интерфейса библиотеки. Помимо прежних возможностей добавлена возможность добавления текстовой гравировки на втулке.

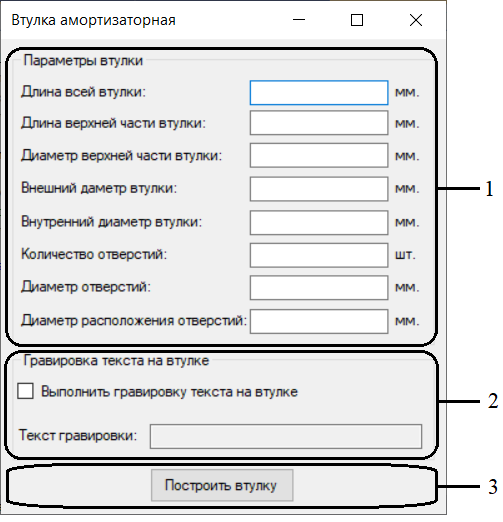


Рисунок 3.6 – Конечный макет пользовательского интерфейса (1 – область для ввода параметров втулки, 2 – область для добавления текстовой гравировки на втулке, 3 – область управления построением)

# **4 Описание программы для пользователя**

При инициализации библиотеки перед пользователем появляется окно, представленное на рисунке 4.1. Чтобы построить втулку амортизатора, необходимо заполнить все поля для ввода параметров втулки, в противном случае будет выведено сообщение об ошибке. При необходимости нанесения текстовой гравировки на втулке необходимо установить флажок «Выполнить гравировку текста на втулке», после чего поле для ввода текста гравировки станет доступным.

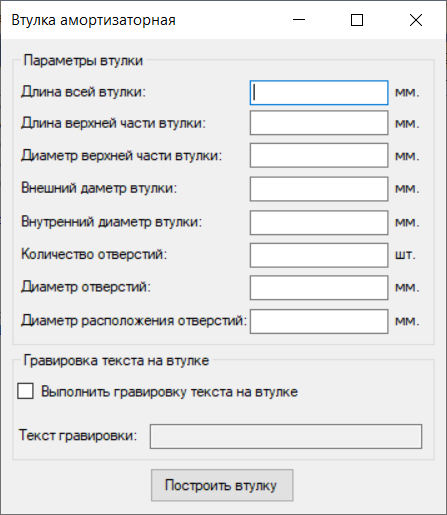


Рисунок 4.1 – Пользовательский интерфейс библиотеки «Втулка амортизатора»

Чтобы снизить количество возможных ошибок при заполнении, следует заполнять поля в установленном порядке. Если введенные данные удовлетворяют поставленным условиям, после нажатия на кнопку «Построить деталь» осуществится запуск САПР «Компас-3D», на рабочей области которой построится трехмерная модель детали по заданным параметрам (рисунок 4.2). Если построение детали выполняется несколько раз, то библиотека не запускает несколько копий программы «Компас-3D», а создает в ней новый документ.

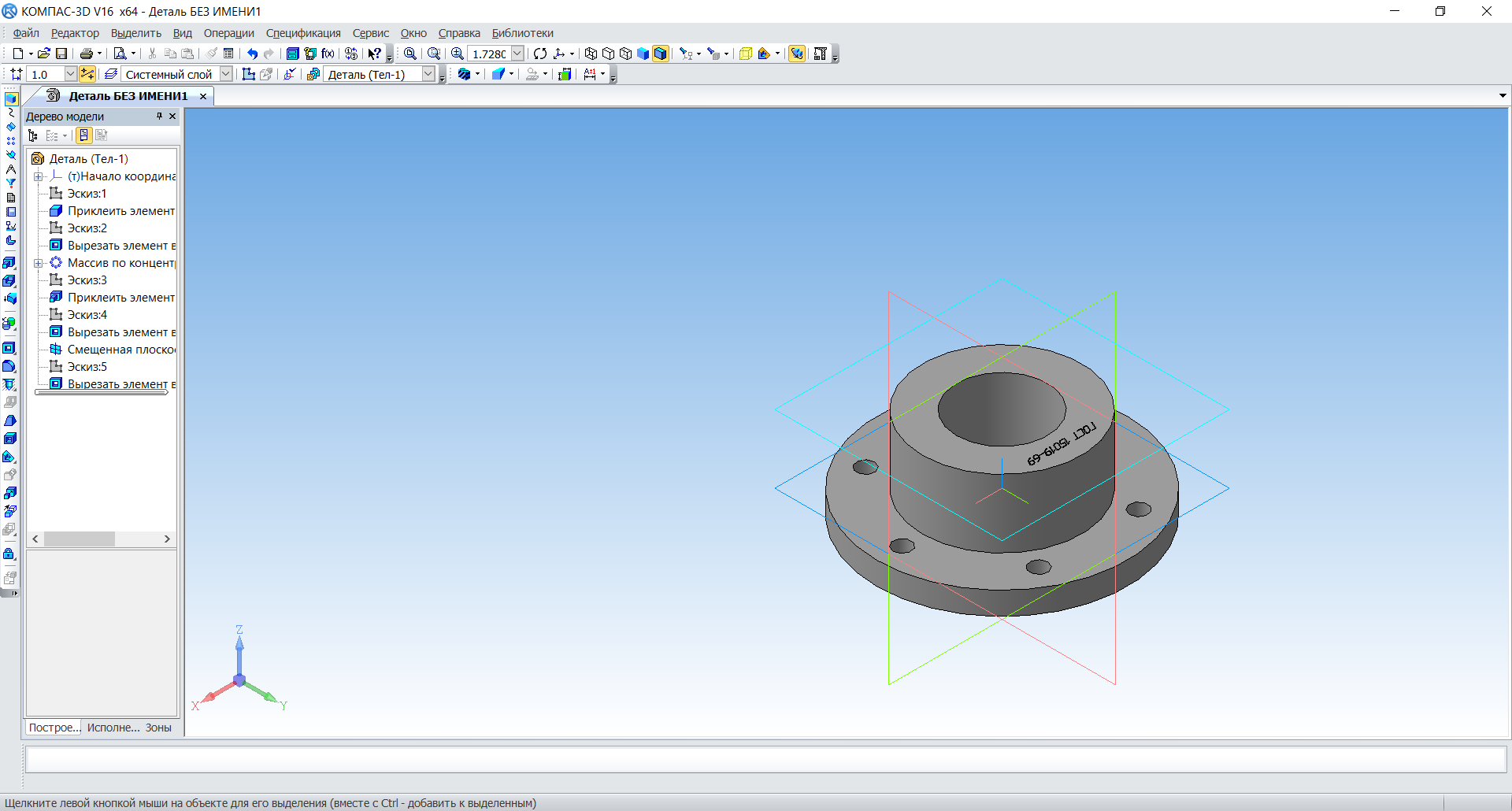


Рисунок 4.2 – Диалоговое окно документа САПР «Компас-3D»

Рассмотрим подробнее все поля для ввода параметров втулки:

* параметр «Длина всей втулки» принимает десятичные значения от 20 мм. до 100 мм;
* параметр «Длина верхней части втулки» принимает десятичные значения от 5 мм. до 1/2 длины всей втулки мм;
* параметр «Диаметр верхней части втулки» принимает десятичные значения от 55 мм. до 120 мм;
* параметр «Внешний диаметр втулки» принимает десятичные значения от 35 мм. до 2/3 диаметра верхней части втулки мм;
* параметр «Внутренний диаметр втулки» принимает десятичные значения от 20 мм. до 2/3 наружного диаметра втулки мм;
* параметр «Количество отверстий» принимает целые значения от 2 шт. до 6 шт;
* параметр «Диаметр отверстий» принимает десятичные значения от 4 мм. до 1/4 внутреннего диаметра втулки мм;
* параметр «Диаметр расположения отверстий» принимает десятичные значения от 3/4 до 4/5 диаметра верхней части втулки мм;
* параметр «Текст гравировки» принимает текст, длина которого не может быть более 15 символов.

В случае если пользователь ввел параметры, превышающие максимально и минимально установленные, будет выведено соответствующее сообщение об ошибке и подсвечено соответствующее поле.

# **5 Тестирование библиотеки**

Тестирование программного обеспечения – это проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом [8].

## **5.1 Функциональное тестирование**

Функциональное тестирование рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом [9].

Произведено функциональное тестирование построения втулки амортизатора в САПР «Комапс-3D» при минимально и максимально заданных параметрах, результаты построения приведены на рисунках 5.1 и 5.2 соответственно.

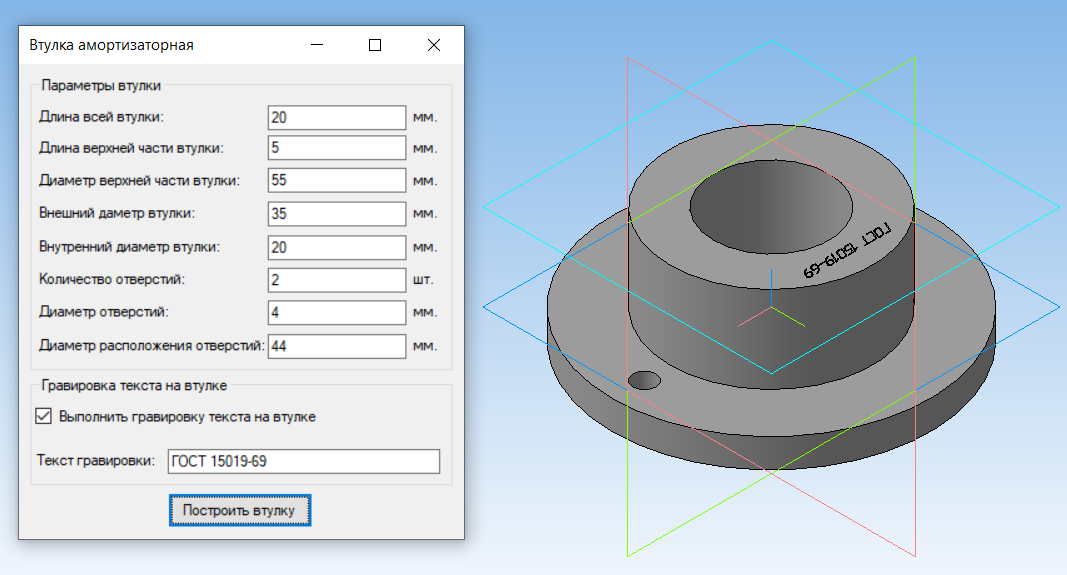


Рисунок 5.1 – Модель втулки амортизатора с минимально заданными параметрами

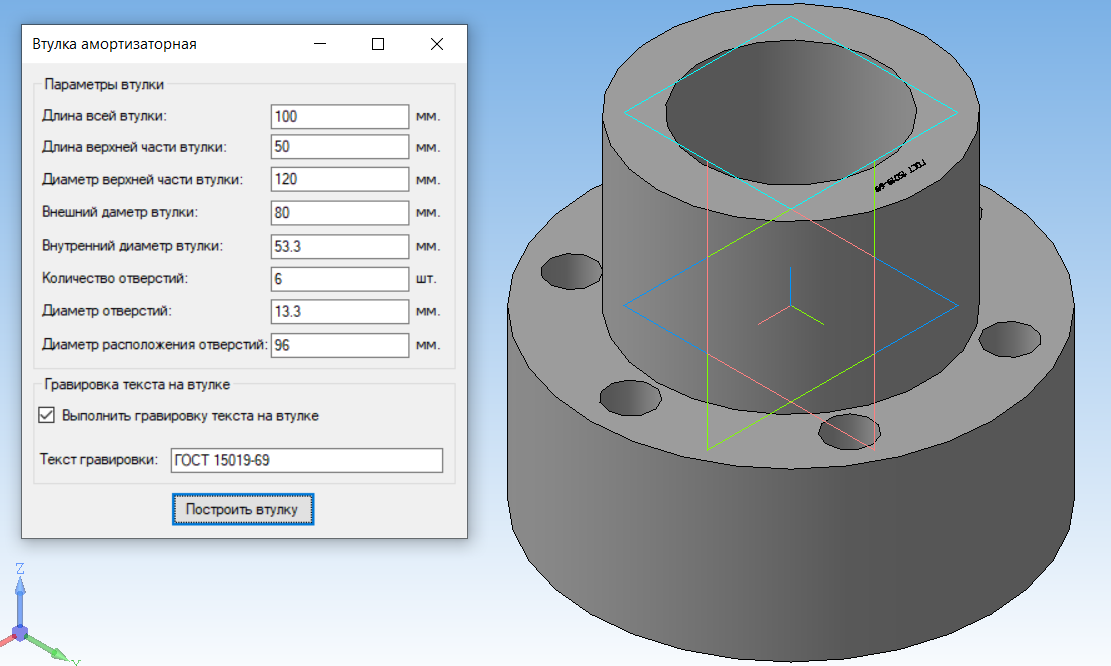


Рисунок 5.2 – Модель втулки амортизатора с максимально заданными параметрами

В случае если данные не заполнены полностью или неверно на момент нажатия кнопки «Построить», будут выводиться подробные сообщения об ошибке с подсветкой полей, которые заполнены неверно. Пример получения диалогового окна, сообщающего об ошибке, представлен на рисунке 5.3.

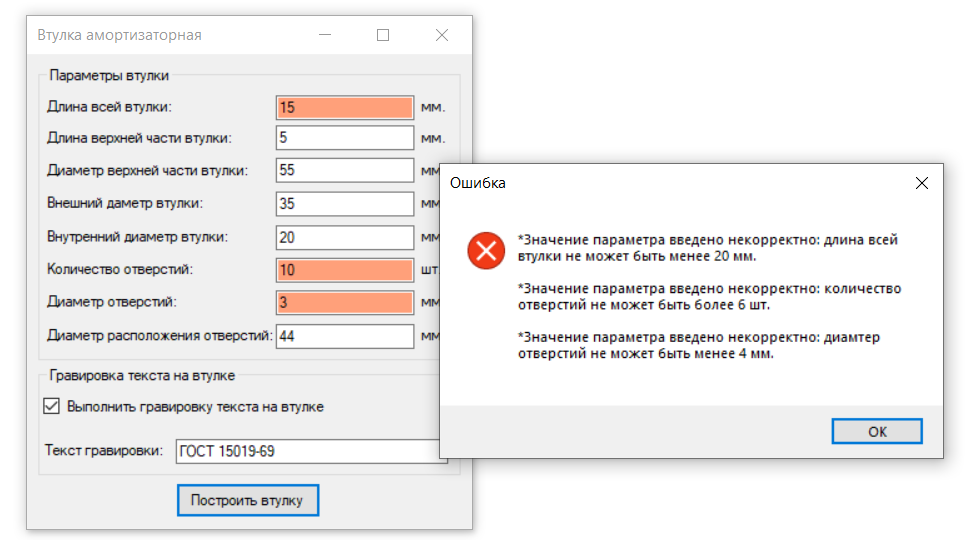


Рисунок 5.3 – Сообщение об ошибке

## **5.2 Модульное тестирование**

Модульное тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.) [10].

Для проверки открытых членов класса Bushing в модульном тестировании использовались тестовые методы, которые описаны в таблице 5.1.

Таблица 5.1 – Название и описание используемых в тестировании методов класса Bushing

|  |  |
| --- | --- |
| Название тестового метода | Описание |
| SetTotalLengthLess\_NegativeTest() | Тестирование при вводе длины всей втулки менее 20 мм. |
| SetTotalLengthMore\_NegativeTest() | Тестирование при вводе длины всей втулки более 100 мм. |
| SetTopLengthLess\_NegativeTest() | Тестирование при вводе длины верхней части втулки менее 5 мм. |
| SetTopLengthMore\_NegativeTest() | Тестирование при вводе длины верхней части втулки более 1/2 длины всей втулки мм. |
| SetTopDiametrLess\_NegativeTest() | Тестирование при вводе диаметра верхней части втулки менее 55 мм. |
| SetTopDiametrMore\_NegativeTest() | Тестирование при вводе диаметра верхней части втулки более 120 мм. |
| SetOuterDiametrLess\_NegativeTest() | Тестирование при вводе внешнего диаметра втулки менее 35 мм. |
| SetOuterDiametrMore\_NegativeTest() | Тестирование при вводе внешнего диаметра втулки более 2/3 диаметра верхней части втулки мм. |
| SetInnerDiametrLess\_NegativeTest() | Тестирование при вводе внутреннего диаметра втулки менее 20 мм. |

Продолжение таблицы 5.1

|  |  |
| --- | --- |
| SetInnerDiametrMore\_NegativeTest() | Тестирование при вводе внутреннего диаметра втулки более 2/3 внешнего диаметра втулки |
| SetNumberHolesLess\_NegativeTest() | Тестирование при вводе количества отверстий менее 2 шт. |
| SetNumberHolesMore\_NegativeTest() | Тестирование при вводе количества отверстий более 6 шт. |
| SetHolesDiametrLess\_NegativeTest() | Тестирование при вводе диаметра отверстий менее 4 мм. |
| SetHolesDiametrMore\_NegativeTest() | Тестирование при вводе диаметра отверстий более 1/4 внутреннего диаметра втулки мм. |
| SetLocationDiametrLess\_NegativeTest() | Тестирование при вводе диаметра расположения отверстий менее 3/4 диаметра верхней части втулки мм. |
| SetLocationDiametrMore\_NegativeTest() | Тестирование при вводе диаметра расположения отверстий более 4/5 диаметра верхней части втулки мм. |
| SetEngravingTextMore\_NegativeTest() | Тестирование при вводе текста гравировки более 15 символов |
| SetEngravingTextNull\_NegativeTest() | Тестирование при вводе пустого тексте гравировки |
| TestTotalLengthGet(double value) | Тестирование получения длины всей втулки |
| TestTopLengthGet(double value) | Тестирование поучения длины верхней части втулки |
| TestTopDiametrGet(double value) | Тестирование получения диаметра верхней части втулки |

Продолжение таблицы 5.1

|  |  |
| --- | --- |
| TestOuterDiametrGet(double value) | Тестирование получения внешнего диаметра втулки |
| TestInnerDiametrGet(double value) | Тестирование получения внутреннего диаметра втулки |
| TestNumberHolesGet(int value) | Тестирование получения количества отверстий |
| TestHolesDiametrGet(double value) | Тестирование получения диаметра отверстий |
| TestLocationDiametrGet(double value) | Тестирование получения диаметра расположения отверстий |
| TestEngravingTextGet (string value) | Тестирование получения текста гравировки |
| TestPresenceEngravingGet(bool value) | Тестирование получения наличия гравировки |

Результаты успешного прохождения всех модульных тестов приведены на рисунке 5.4.

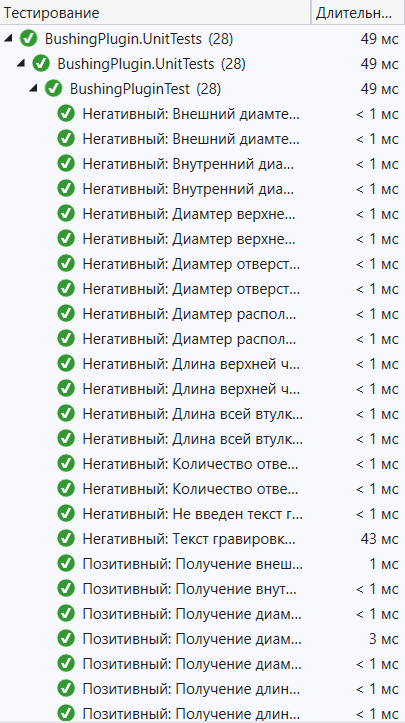


Рисунок 5.4 – Результат выполнения модульных тестов

## **5.3 Нагрузочное тестирование**

Нагрузочное тестирование – это тестирование 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) [11].

Нагрузочное тестирование проводилось со следующими параметрами втулки:

* длина всей втулки: 100 мм.;
* длина верхней части втулки: 50 мм.;
* диаметр верхней части втулки: 120 мм.;
* внешний диаметр втулки: 80 мм.;
* внутренний диаметр втулки: 53.3 мм.;
* количество отверстий: 6 шт.
* диаметр отверстий: 13.3 мм.;
* диаметр расположения отверстий: 96 мм.;
* текст гравировки: ТекстТекстТекст.

После построения 116 детали программа завершилась со сбоем из-за нехватки памяти.

На рисунке 5.5 представлен график потребляемой оперативной памяти относительно количества построенных деталей.

Рисунок 5.5 – График потребляемой оперативной памяти относительно числа созданных деталей

График зависимости загрузки программой центрального процессора от количества построенных деталей изображен на рисунке 5.6.

Рисунок 5.6 – График зависимости загрузки программой центрального процессора от количества построенных деталей

По полученным графикам можно сделать вывод, что зависимость прямая, т.е. используемая оперативная память и загрузка центрального процессора при увеличении количества деталей увеличивается линейно.

**Заключение**

В ходе выполнения данного проекта были изучены основные этапы разработки ПО на примере создания библиотеки для построения детали «Втулка амортизатора» для САПР «КОМПАС-3D V 16».

В разработанной библиотеке предусмотрено открытие и закрытие системы «КОМПАС-3D», проверка вводимых данных на корректность, построение детали по верно заданным параметрам, вывод сообщений об ошибке и подсветка неверно заполненных полей в ином случае.

Проведены модульные, функциональные и нагрузочные тесты разработанной библиотеки.

# **Список использованных источников**

1. КОМПАС-3D: О программе [Электронный ресурс]. – Режим доступа: http://kompas.ru/kompas-3d/about/ (дата обращения: 27.02.2020)

2. Программа автоматического построения 3D моделей и разверток по заданным значениям в AutoCAD «Лекало». Расчет и построение механических передач [Электронный ресурс] – Режим доступа: https://www.2d-3d.ru/3d-galereia/autocad/811-programma-dlya-autocad-lekalo.html (дата обращения: 27.02.2020)

3. Болты, гайки, шайбы... Быстрое моделирование [Электронный ресурс] – Режим доступа: https://3dtoday.ru/blogs/3dlab/bolts-nuts-washers-quick-modeling (дата обращения: 02.03.2020)

4. Втулки – Техноберинг [Электронный ресурс] – Режим доступа: https://technobearing.ru/eshop1/folder/vtulki (дата обращения: 02.03.2020)

5. ГОСТ 15019-69 – Втулки центрирующие и направляющие для литейных опок. Конструкция и размеры [Электронный ресурс] – Режим доступа: https://engenegr.ru/gost-15019-69 (дата обращения: 02.03.2020)

6. UML. Основы / Фаулер, М. – 3-е изд., пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.

7. Интерфейс пользователя – Национальная библиотека им. Н. Э. Баумана [Электронный ресурс] – Режим доступа: https://ru.bmstu.wiki/Интерфейс\_пользователя (дата обращения: 02.03.2020)

8. Про Тестинг – Тестирование программного обеспечения [Электронный ресурс] – Режим доступа: http://www.protesting.ru/testing/ (дата обращения: 03.05.2020)

9. Про Тестинг – Тестирование – Виды тестирования ПО – Функциональное тестирование [Электронный ресурс] – Режим доступа: http://www.protesting.ru/testing/types/functional.html (дата обращения: 03.05.2020)

10. Про Тестинг – Тестирование – Уровни Тестирования ПО – Компонентное или Модульное тестирование [Электронный ресурс] – Режим доступа: http://www.protesting.ru/testing/levels/component.html (дата обращения: 03.05.2020)

11. Про Тестинг – Тестирование – Виды тестирования ПО – Нагрузочное тестирование [Электронный ресурс] – Режим доступа: http://www.protesting.ru/testing/types/loadtesttypes.html (дата обращения: 03.05.2020)