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 ПЛАГИНА «Бутылка» ДЛЯ «КОМПАС-3D v18»**

Пояснительная записка к лабораторному проекту

по дисциплине «Основы разработки САПР»

«Построение бутылки в системе КОМПАС 3D v18»

Выполнил:

студент гр.587-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Степанюга А.И.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент каф. КСУП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Калентьев А. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Оглавление

[**Введение** 3](#_Toc70589916)

[**1** **Описание САПР** 4](#_Toc70589917)

[**1.1 Описание программы** 4](#_Toc70589918)

[**1.2 Описание API** 4](#_Toc70589919)

[**1.3** **Обзор аналогов** 5](#_Toc70589920)

[**2** **Описание предмета проектирования** 9](#_Toc70589921)

[**3 Описание технических и функциональных аспектов проекта** 10](#_Toc70589922)

[**3.1 Описание полей, свойств и методов, используемых в проекте** 10](#_Toc70589923)

[**3.2 Макет пользовательского интерфейса** 11](#_Toc70589924)

[**4 Тестирование** 13](#_Toc70589925)

[**4.1 Функциональное тестирование** 13](#_Toc70589926)

[**4.2 Модульное тестирование** 13](#_Toc70589927)

[**4.2** **Нагрузочное тестирование** 14](#_Toc70589928)

[**Заключение** 16](#_Toc70589929)

[**Список литературы** 17](#_Toc70589930)

**Введение**

Современные пакеты программного обеспечения для автоматизированного проектирования варьируются от 2D-векторных систем черчения до 3D-моделей твердого тела и поверхности. CAD пакеты также часто допускают вращение в трех измерениях, позволяя просматривать проектируемый объект под любым желаемым углом, даже изнутри наружу. Некоторые программы CAD способны к динамическому математическому моделированию. Технология CAD используется при проектировании инструментов и механизмов, а также при проектировании всех типов зданий, от небольших жилых домов до крупнейших коммерческих и промышленных сооружений (больниц и заводов). [1]

САПР является важным звеном в промышленном конструировании, широко используемым во многих отраслях, в том числе в автомобильной, судостроительной и аэрокосмической промышленности, промышленном и архитектурном проектировании, протезировании и многих других. САПР также широко используется в создании компьютерной анимации для спецэффектов в фильмах, рекламных и технических материалах, часто называемых цифровым контентом. Современное повсеместное распространение компьютеров означает, что даже флаконы для духов и диспенсеры для шампуней сегодня разрабатываются с использованием информационных технологий, невиданных инженерами 1960-х годов. Из-за своей огромной экономической важности, САПР стал основной движущей силой исследований в области вычислительной геометрии, компьютерной графики (как аппаратной, так и программной) и дискретной дифференциальной геометрии. [2]

1. **Описание САПР**

**1.1 Описание программы**

КОМПАС-3D — система трехмерного проектирования, ставшая стандартом для тысяч предприятий, благодаря сочетанию простоты освоения и легкости работы с мощными функциональными возможностями твердотельного и поверхностного моделирования. Ключевой особенностью продукта является использование собственного математического ядра C3D 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.[3]

**1.2 Описание API**

В КОМПАС на данный момент существуют API двух версий: API 5 и API 7. Обе версии реализуют различные функции системы и взаимно дополняют друг друга. Отсюда очевидно, что обе версии программных интерфейсов в равной мере поддерживаются и развиваются с учетом самих изменений в системе. В основном, для создания полноценных подключаемых модулей достаточно методов и свойств интерфейсов API 5.

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно при работе под управлением внешнего приложения (контроллера) - после вызова стандартной системной функции. Методы этого интерфейса, реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа).

* 1. **Обзор аналогов**

**Плагин PmDF**

Данный плагин позволяет производить экспорт моделей и сборок из КОМПАС-3D в формат PDF формат. [3] Основной особенностью является возможность интерактивного взаимодействия пользователя с сохраненной 3D сценой внутри PDF файла. Например, пользователь может вращать, масштабировать, передвигать детали и сборки внутри 3D PDF файла. Также доступно создание анимации сборки и разборки изделий. Это полезно для подготовки интерактивных сборочных инструкций, создания маркетинговых материалов, презентаций, а также для налаживания взаимодействия между проектировщиками и заказчиками. В подобных ситуациях традиционным подходом являлся экспорт сборки или детали КОМПАС-3D в промежуточный формат и дальнейшее сохранение в формат 3D PDF. Используемый подход в плагине исключает использование промежуточных файлов для осуществления 3D преобразования, что существенно повышает качество выходной 3D модели в формате PDF.

Ключевые возможности плагина:

* сохранение деталей и сборок в формате 3D PDF для интерактивного просмотра при помощи бесплатной программы Adobe Reader;
* создание анимаций, имитирующих естественный порядок сборки и разборки создание имитации анимации гибки листовых тел;
* вставка в существующие PDF документы, содержащие основной текст, фоновые картинки, таблицы спецификаций, эмблемы, логотипы;
* пакетный режим для поочередной конвертации всех файлов.

Ниже в таблице 1.1 представлены свойства и методы интерфейса KompasObject, которые будут использованы при разработке плагина. [4]

Таблица 1.1 – Некоторые методы и свойства интерфейса KompasObject

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Document3D() | ksDocument | Метод для получения указателя на интерфейс трехмерного графического документа (детали или сборки) |
| GetMathematic2D() | ksMathematic2D | Метод для получения указателя на интерфейс для работы с математическими функциями |
| GetParamStruct(short structType) | StructType2D | Метод для получения указателя на интерфейс графического объекта |
| GetDynamicArray(long type) | ksDynamicArray | Метод для получения указателя на интерфейс динамического массива ksDynamicArray |
| Visible | bool | Свойство видимости приложения |

В таблице 1.2 представлены методы интерфейса ksEntity, которые были использованы при разработке плагина.

Таблица 1.2 – Некоторые методы интерфейса ksEntity

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

В таблице 1.3 представлены свойства и методы интерфейса ksDocument3D, которые были использованы при разработке плагина.

Таблица 1.3 – Некоторые свойства и методы интерфейса ksDocument3D

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Create (bool invisible, bool \_typeDoc) | bool | Создать документ-модель (деталь или сборку) |
| UpdateDocumentParam() | bool | Активизировать измененные параметры документа |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |

В таблице 1.4 представлены методы интерфейса ksPart, которые были использованы при разработке плагина.

Таблица 1.4 – Некоторые свойства и методы интерфейса ksPart

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| EntityCollection(short objType) | ksEnintyCollection | Формирует массив объектов и возвращает указатель на его интерфейс |
| GetDefaultEntity(short objType) | ksEntity | Получить указатель на интерфейс объекта, создаваемого системой по умолчанию |
| GetPart(int type) | ksPart | Получить указатель на интерфейс компонента в соответствии с заданным типом |
| NewEntity(short objType) | ksEntity | Создать новый интерфейс объекта и получить указатель на него |

Таблица 1.5 – Свойства и методы интерфейса ksPlaneOffsetDefinition.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип входных параметров | Тип возвращаемых данных | Описание |
| SetPlane | plane - указатель на интерфейс плоскости ksEntity или IEntity. | TRUE - в случае успешного завершения. | Изменить указатель на интерфейс базовой плоскости |

В таблице 1.6 представлены типы объектов документа-модели, которые были использованы при разработке плагина.

Таблица 1.6 – Некоторые типы объектов документа-модели

|  |  |  |
| --- | --- | --- |
| Идентификатор объекта | Название объекта | Интерфейс параметров |
| o3d\_planeXOZ | Плоскость XOZ | ksPlaneParam |
| o3d\_sketch | эскиз | [ksSketchDefinition](about:blank) |
| o3d\_planeOffset | смещённая плоскость | [ksPlaneOffsetDefinition](about:blank) |
| o3d\_edge | ребро | [ksEdgeDefinition](about:blank) |
| o3d\_chamfer | операция "фаска" | [ksChamferDefinition](about:blank) |
| o3d\_cutExtrusion | вырезать выдавливанием | [ksCutExtrusionDefinition](about:blank) |
| o3d\_axisOY | Ось OY | ksPlaneParam |
| o3d\_circularCopy | операция копирования по концентрической сетке | [ksCircularCopyDefinition](about:blank) |
| o3d\_baseExtrusion | базовая операция выдавливания | [ksBaseExtrusionDefinition](about:blank) |

1. **Описание предмета проектирования**

Предметом проектирования является модель бутылки. Данная модель имеет 5 основных параметров:

1. Высота бутылки H (100≤Н≤250) мм;
2. Высота основания O (68≤О≤Н\*2/3) мм;
3. Высота горлышка G (20≤G≤Н/5) мм;
4. Диаметр основания D (25≤D≤65) мм;
5. Диаметр горлышка d (13≤d≤D\2) мм.

На рисунке 2.1 представлен чертеж бутылки.

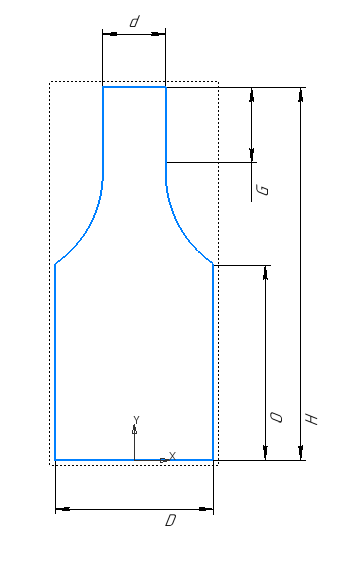


Рисунок 2.1 – Чертеж бутылки.

**3** **Описание технических и функциональных аспектов проекта**

**3.1 Описание полей, свойств и методов, используемых в проекте**

При использовании UML была построена диаграмма классов. Данная диаграмма представлена на рисунке 3.3 [5].

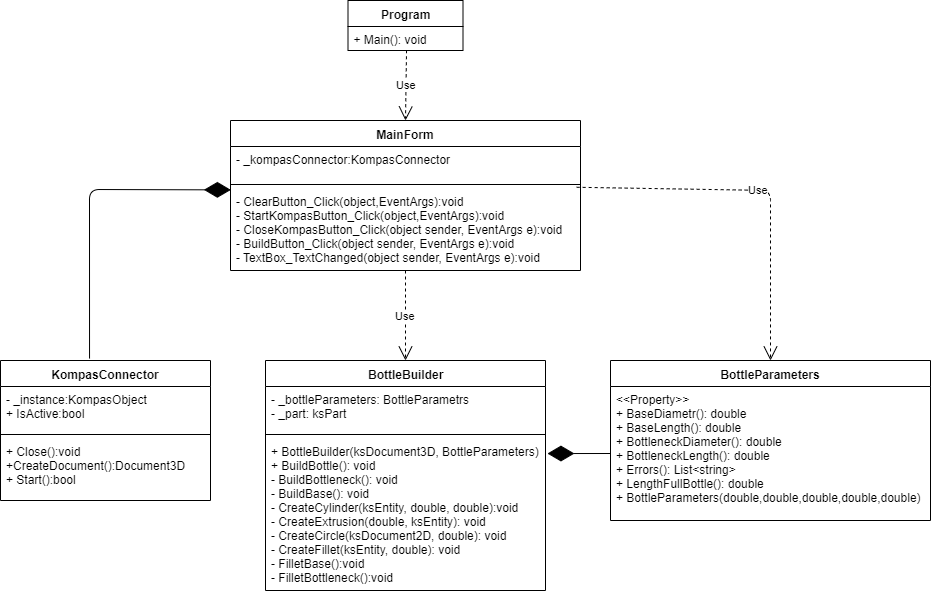


Рисунок 3.1 – UML-диаграмма классов.

В ходе реализации дополнительной функциональности, UML диаграмма претерпела изменения и добавление новых методов.

Для реализации дополнительной функциональности были добавлены методы в классе BottleBuilder:

CreateOpener – метод, реализующий построение открывашки на дне бутылки;

CreateRectangle – метод, для создания экскиза четырехугольника;

MakeCutExtrude – метод, реализующий вырез.

Был изменен тип доступа поля \_instance в классе KompasConnector на public

Новая диаграмма показана на рисунке 3.2

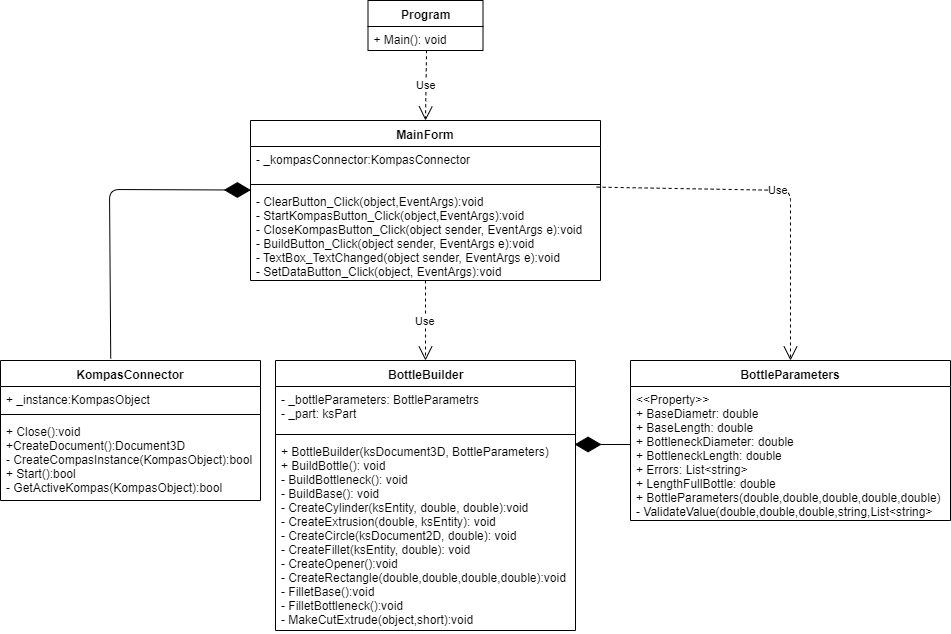


Рисунок 3.2 – Обновленная UML –диаграмма классов

KompasConnector – класс, реализующий работу с API Компас.

BottleParametrs – класс, описывающий параметры бутылки.

BottleBuilder – класс, реализующий методы построения бутылки.

**3.2 Макет пользовательского интерфейса**

Плагин представляет собой меню и пользовательскую форму с ячейками для ввода параметров. Ниже находятся формы для заполнения: «Высота бутылки», «Высота основания», «Высота горлышка», «Диаметр основания» и «Диаметр горлышка».

Под этими формами располагаются кнопки «Очистить», «Построить» и «Заполнить поля».

Макет пользовательского интерфейса представлен на рисунке 3.3.

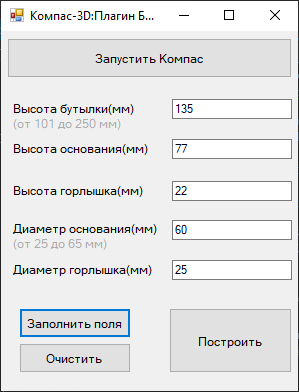


Рисунок 3.3 – Макет пользовательского интерфейса программы.

При вводе некорректных значений, программа выдаст окно с параметрами, которые не подходят для построения. Сообщение показано на рисунке 3.4.

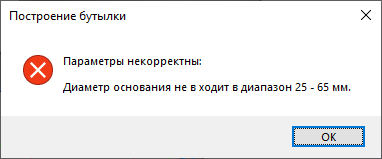


Рисунок 3.4 – Вывод сообщения о некорректных данных

**4 Тестирование**

**4.1 Функциональное тестирование**

При функциональном тестировании проверяется работа программы на соответствие требований, которые были заложены. Проверка будет осуществляться путем ввода минимальных и максимальных параметров параметров.

Минимальные параметры:

* Длина бутылки (H) 101;
* Высота основания (O) 66,7;
* Высота горлышка (G) 20;
* Диаметр основания (D) 25;
* Диаметр горлышка (d) 12,5.

Построенная модель показана на рисунке 4.1

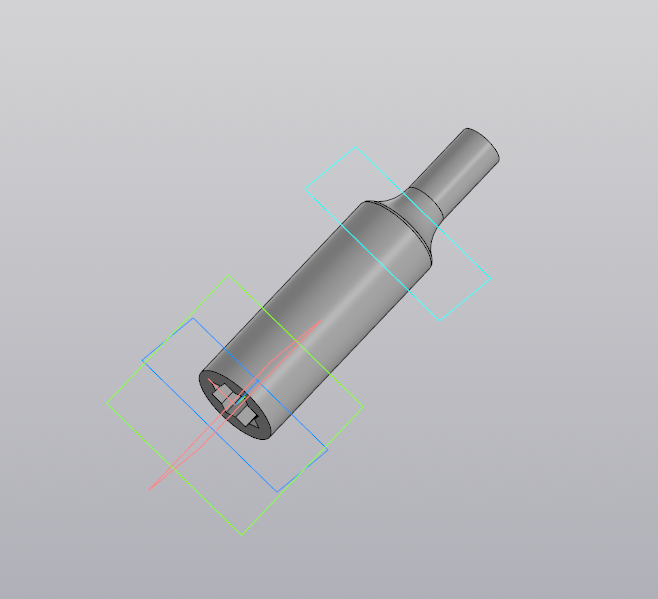


Рисунок 4.1 – Модель с минимальными параметрами

Максимальные параметры:

* Длина бутылки (H) 250;
* Высота основания (O) 166,6;
* Высота горлышка (G) 50;
* Диаметр основания (D) 65;
* Диаметр горлышка (d) 32,5.

Построенная модель показана на рисунке 4.2

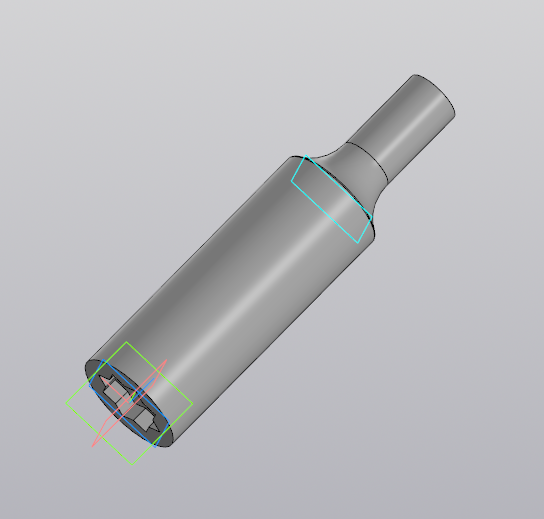


Рисунок 4.2 – Модель с максимальными параметрами

**4.2 Модульное тестирование**

Юнит-тестирование— тестирование отдельного элемента изолированно от остальной системы. Относительно парадигмы объектно-ориентированного программирования, системой является вся программа, а отдельным элементом — класс или его метод. Юнит-тестирование предназначено для проверки правильности работы отдельно взятого элемента. Чтобы исключить из результатов тестирования влияние потенциальных ошибок других элементов, тестируемый элемент должен быть максимально изолирован, то есть не использовать объекты и методы других классов [6].

Тестирование проводилось с помощью фреймворка модульного тестирования NUnit 3.13.1 для языков платформы .NET.

В данном плагине проводилось тестирование класса BottleParametrs. На рисунке 4.3 показаны проводимые тесты.

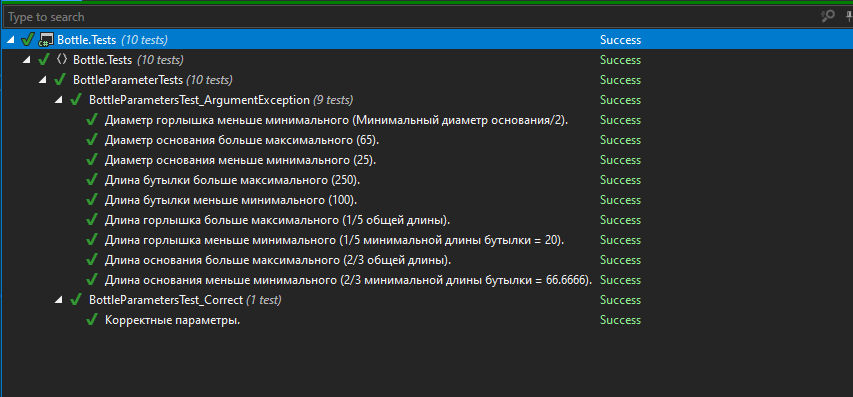
****

Рисунок 4.3 – Список юнит - тестов

Описание тестовых случаев представлены в приложении А в таблице 1.

На рисунке 4.4 показана площадь покрытия тестами

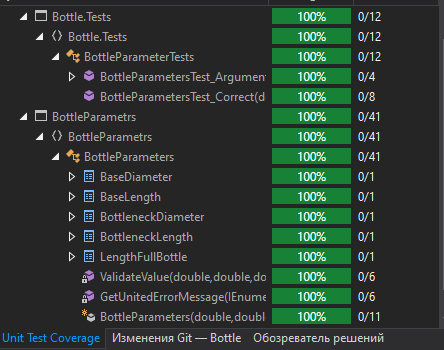


Рисунок 4.4 – Площадь покрытия тестов

* 1. **Нагрузочное тестирование**

**Заключение**

В ходе прохождения курса «Основы разработки САПР» были:

* приобретены навыки проектирования плагина для САПР КОМПАС 3D
* Получены умения использования API КОМПАС 3D
* Закреплены знания об этапах проектирования программного продукта, составлении технического задания и UML диаграмм классов;
* Закреплены знания в написании юнит-тестов.

Результатом является спроектированный плагин для САПР «КОМПАС 3D для построения модели бутылки с открывашкой на дне.

**Список литературы**

1. Норенков И. П. Автоматизированное проектирование. Учебник. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2000. — 188 с.
2. Морозов К. К., Одиноков В. Г., Курейчик В. М. Автоматизированное проектирование конструкций радиоэлектронной аппаратуры. — М.: Радио и связь, 1983. — 280 с. — (Учебное пособие для вузов).
3. Плагин PDF3D для САПР КОМПАС-3D [Электронный ресурс]. – URL: http://gkmsoft.ru/pdf3d\_kompas.html (дата обращения: 16.11.18)
4. Экспорт из Компас-3D в формате 3D PDF [Электронный ресурс]. – URL: <http://sapr-journal.ru/novosti/eksport-iz-kompas-3d-v-formate-3d-pdf/> (дата обращения 15.03.2020)
5. М. Фаулер. «UML. Основы». Издательство: Символ-Плюс; Booch Jacobson Rumbaugh: 2018 – 192 с.
6. Новые технологии в программировании: учебное пособие / А.А.Калентьев, Д.В.Гарайс, А.Е.Горяинов— Томск : Эль Контент, 2014.—176 с.

**Приложение А**

**Таблица 1 – Тестовые сценарии**

|  |  |  |
| --- | --- | --- |
| Метод | Входные параметры | Описание теста |
| BottleParametrsTest\_ArgumentException(Double, Double, Double, Double, Double) | BaseDiameter 24,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 130 | Диаметр основания меньше минимального(25) |
| BaseDiameter 30,  BaseLength 65,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 130 | Длина основания меньше минимальной(66,7) |
| BaseDiameter 30,  BaseLength 80,  BottleneckDiameter 9,  BottleneckLenght 22,  LengthFullBottle 130 | Диаметр горлышка меньше минимального(15) |
| BaseDiameter 30,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 19,  LengthFullBottle 130 | Длина горлышка меньше минимальной(20) |
| BaseDiameter 30,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 99 | Длина бутылки меньше минимальной(101) |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| BottleParametrsTest\_ArgumentException(Double, Double, Double, Double, Double) | BaseDiameter 66,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 130 | Диаметр основания больше максимального(65) |
| BaseDiameter 30,  BaseLength 88,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 130 | Длина основания больше максимальной(86,6) |
| BaseDiameter 30,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 28,  LengthFullBottle 130 | Длина горлышка больше максимального(26) |
| BaseDiameter 30,  BaseLength 80,  BottleneckDiameter 20,  BottleneckLenght 22,  LengthFullBottle 255 | Длина бутылки больше максимальной(250) |
| BottleParametersTest\_Correct(Double, Double, Double, Double, Double) | BaseDiameter 60,  BaseLength 77,  BottleneckDiameter 18,  BottleneckLenght 22,  LengthFullBottle 135 | Ввод корректных значений |