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)

ПЛАГИН “БУТЫЛКА” ДЛЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ (САПР) AUTOCAD

Пояснительная записка по дисциплине “Основы разработки САПР”

Выполнил:

студент гр. 580-2

\_\_\_\_\_\_Сеченов В.В.

«18» декабря 2023 г.

Проверил:

К.т.н., Доцент кафедры КСУП

\_\_\_\_\_Калентьев А.А.

«\_\_\_» декабря 2023 г.

Томск 2023

# СОДЕРЖАНИЕ

[Введение 3](#_Toc153737311)

[1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ 4](#_Toc153737312)

[2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ 5](#_Toc153737313)

[3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ 7](#_Toc153737314)

[4 НАЗНАЧЕНИЕ ПЛАГИНА 8](#_Toc153737315)

[5 ОБЗОР АНАЛОГОВ 9](#_Toc153737316)

[6 ОПИСАНИЕ РЕАЛИЗАЦИИ 10](#_Toc153737317)

[7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ 19](#_Toc153737318)

[8 ТЕСТИРОВАНИЕ ПЛАГИНА 21](#_Toc153737319)

[8.1 Функциональное тестирование 21](#_Toc153737320)

[8.2 Модульное тестирование 29](#_Toc153737321)

[8.3 Нагрузочное тестирование 32](#_Toc153737322)

[Заключение 34](#_Toc153737323)

[Список использованных источников 35](#_Toc153737324)

# Введение

Любое производство существует до тех пор, пока оно является прибыльным. Одним из способов максимизации рентабельности компании является минимизация издержек.

Минимизация издержек в производствах имеет вид минимизации брака. Уменьшить количество брака до приемлемого минимума помогает моделирование перед непосредственным производством детали.

Бутылочные производства, как и любые другие, нуждаются в моделировании и оценке физических свойств модели детали перед ее производством. В современном мире стандартом является компьютерное моделирование при помощи систем автоматизированного проектирования (САПР). Одной из самых распространенных САПР в мире на данный момент является Autodesk AutoCAD, которая не имеет встроенного плагина для построения бутылок.

Плагин для автоматизации создания модели бутылки ускорит процесс моделирования, что позволит быстрее запустить производство и начать получать прибыль.

# 1 ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ

В рамках проекта была поставлены задачи:

1. Составить ТЗ (25.10.2023 – 09.10.2023);
2. Составить проект системы (09.10.2023 – 22.10.2023);
3. Реализовать систему (22.10.2023 – 05.11.2023);
4. Написать юнит-тесты (06.11.2023 – 20.11.2023);
5. Реализовать дополнительную функциональность, выбранную преподавателем (21.11.2023 – 05.12.2023);
6. Провести нагрузочное тестирование (06.12.2023 – 13.12.2023);
7. Написать пояснительную записку (14.12.2023 – 18.12.2023).

В процессе анализа задач были найдены следующие возможные проблемы, которые могут возникнуть при разработке плагина:

* Обработка различных типов исключений;

Несколько параметров модели, кроме валидации на вхождение в определенный диапазон, еще проходят валидацию на соответствие условиям, включающих другие параметры. Может возникнуть проблема со сбором всех исключений и их обработкой.

* Возникновение непредвиденных ошибок при использовании API, предоставляемого САПР.

При анализе использовалась официальная документация API ObjectARX и официальная документация по C#.

В виду простоты создаваемой модели и довольно исчерпывающей документации отрицательные результаты отсутствуют.

Положительными результатами анализа являются:

* Найдена возможность агрегации нескольких исключений их последовательной обработке;
* Найдены все классы и методы в API, необходимые для реализации модели.

# 2 ОПИСАНИЕ ПРЕДМЕТА ПРОЕКТИРОВАНИЯ

Предметом проектирования является модель бутылки, изначально обладающая следующими изменяемыми параметрами:

* длина основной части L (10 – 250 мм);
* ширина основной части W (10 – 250 мм);
* высота основной части H1 (10 – 250 мм);
* высота горлышка H2 (10 – 40 мм);
* радиус горлышка R (5 – 20 мм).

Некоторые из них обладают следующими зависимостями:

* высота горлышка H2 от высоты основной части H1:
* радиус горлышка R от ширины W:
* радиус горлышка R от длины L:

В процессе разработки преподаватель потребовал реализовать дополнительные функциональности, из-за чего появилось дополнительные параметры:

* длина основания горлышка L2 (5 – 20 мм);
* ширина основания горлышка W2 (5 – 20 мм);
* радиус основания основной части R1 (10 – 125 мм);

На рисунках 2.1 и 2.2 представлены изображения предмета проектирования.



Рисунок 2.1 – Модель бутылки с изначальными параметрами

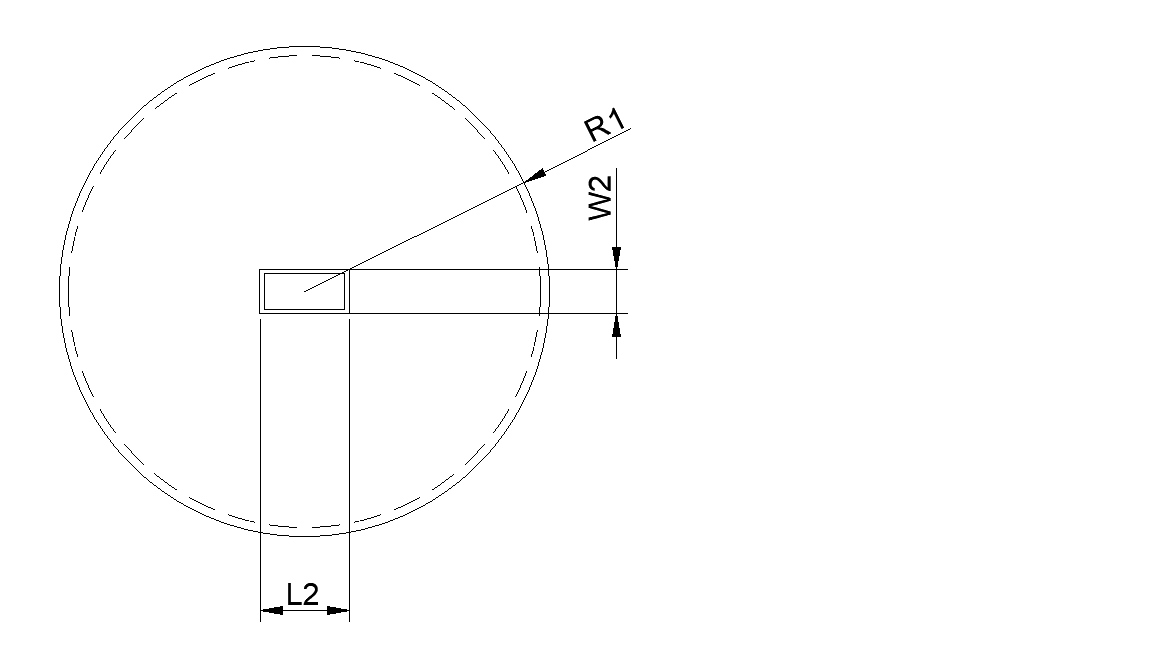


Рисунок 2.2 – Модель бутылки с дополнительными параметрами

# 3 ВЫБОР ИНСТРУМЕНТОВ И СРЕДСТВ РЕАЛИЗАЦИИ

В процессе разработки плагина использовались такие технологии как:

* .NET 6

.NET – это бесплатная кроссплатформенная платформа для разработчиков с открытым исходным кодом для создания множества различных типов приложений.

* WinForms

WinForms – это интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework.

* ObjectARX

Object ARX – это официальный API для системы автоматизированного проектирования AutoCAD. Написан на C++, но совместим с C#. Предоставляет классы и методы для построения моделей в приложении.

* ReSharper

Resharper – это дополнение, разработанное компанией JetBrains для повышения продуктивности работы с Microsoft Visual Studio. Не позволяет скомпилировать приложение до тех пор, пока не будут соблюдены правила оформления кода.

* StyleCop

StyleCop – это инструмент статического анализа кода с открытым исходным кодом от Microsoft, который проверяет код C# на соответствие рекомендуемым стилям кодирования StyleCop и подмножеству руководящих указаний Microsoft по проектированию .NET Framework.

# 4 НАЗНАЧЕНИЕ ПЛАГИНА

Назначение разрабатываемого плагина обусловлено быстрым моделированием бутылок разных типов. Благодаря данному расширению, производители бутылок могут рассмотреть спроектированную модель и при необходимости перестроить под необходимые им параметры.

# 5 ОБЗОР АНАЛОГОВ

Аналогом является плагин для создания бутылки Wine Bottle для программы для 3D дизайна и архитектурного проектирования SketchUp. На рисунке 5.1 представлен интерфейс плагина Wine Bottle.



Рисунок 5.1 – Интерфейс плагина Wine Bottle

Преимуществом данного плагина в сравнении с реализованным является возможность добавления скруглений бутылке.

Недостатком данного плагина в сравнении с реализованным является отсутствие выбора формы горлышка и основной части.

# 6 ОПИСАНИЕ РЕАЛИЗАЦИИ

На рисунке 6.1 представлена UML-диаграмма классов после проектирования.

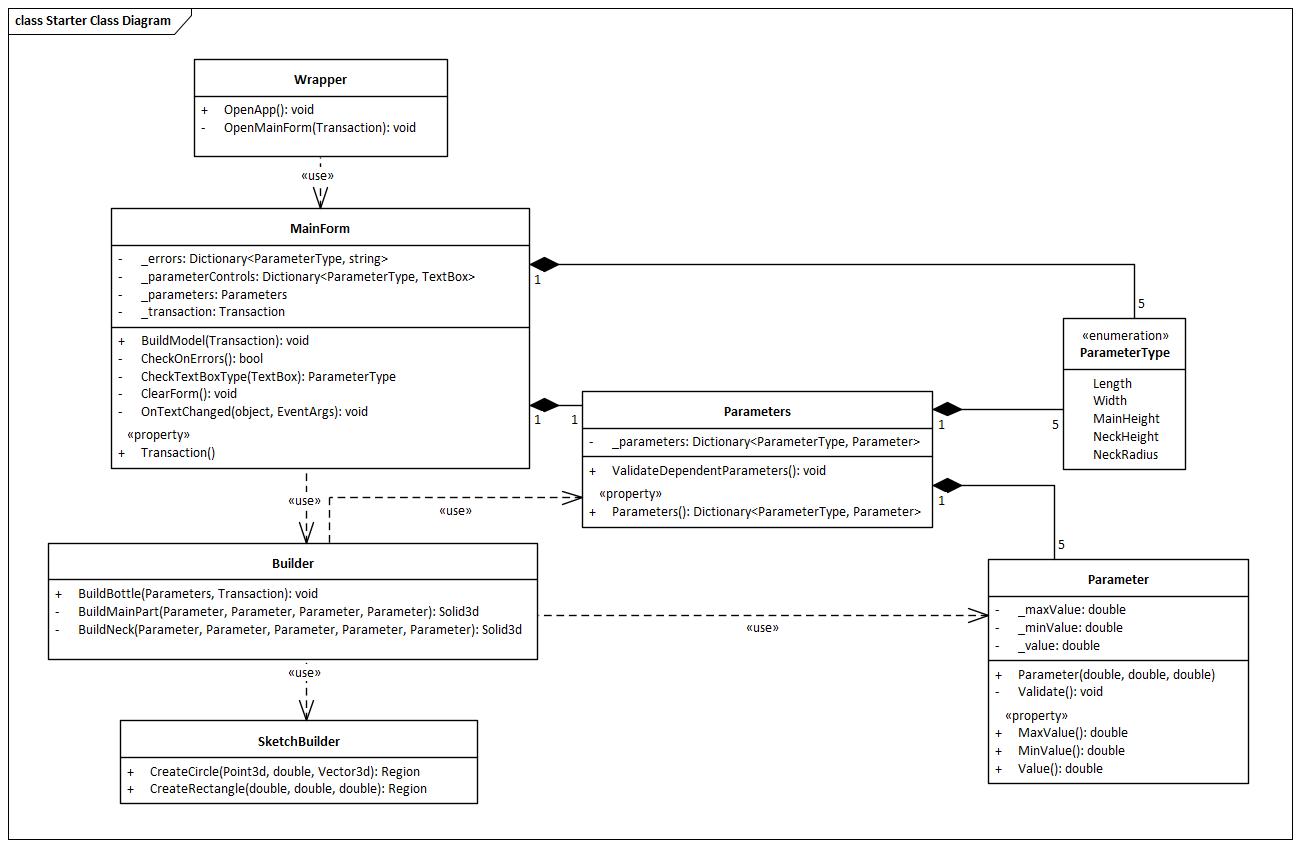


Рисунок 6.1 – Диаграмма классов после проектирования

На рисунке 6.2 представлена UML-диаграмма классов после реализации.

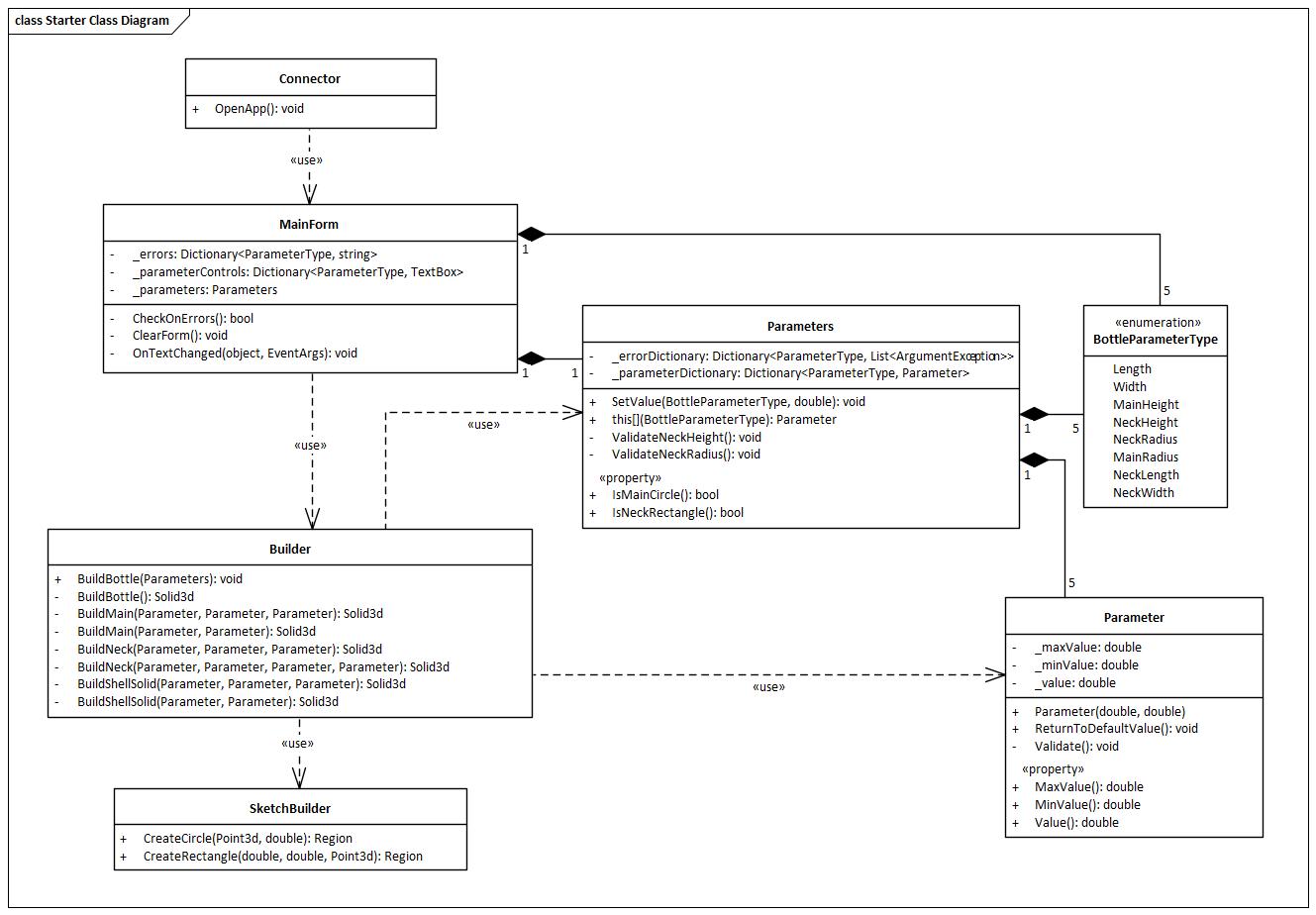


Рисунок 6.2 – Диаграмма классов после реализации

В таблицах 6.1 – 6.9 представлено описание программных сущностей и их членов.

Класс Parameter содержит в себе информацию об изменяемом параметре.

Таблица 6.1 – Свойства класса Parameter

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| MaxValue | double | Свойство, возвращающее максимальное значение, которое может принимать параметр. |
| MinValue | double | Свойство, возвращающее минимальное значение, которое может принимать параметр. |
| Value | double | Свойство, возвращающее текущее значение параметра. |

Таблица 6.2 – Методы класса Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| Parameter | double maxValue, double minValue | Parameter | Конструктор класса Parameter. |
| Validate | - | void | Валидирует текущее значение параметра. |
| ReturnToDefaultValue | - | void | Возвращает значение параметра к минимальному возможному для данного параметра. |

Класс ParameterType является перечислением, содержащим типы параметров, которые имеет модель. ParameterType может принимать значения Length, Width, MainHeight, NeckHeight, NeckRadius.

Класс Parameters содержит в себе все параметры модели.

Таблица 6.3 – Свойства класса Parameters

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | Описание |
| IsMainCircle | bool | Свойство, возвращающее значение, определяющее какая форма основной части выбрана. |
| IsNeckRectangle | bool | Свойство, возвращающее значение, определяющее какая форма горлышка выбрана. |

Таблица 6.4 – Методы класса Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| this[] | BottleParameterType | Parameter | Возвращает параметр определенного типа. |
| SetValue | BottleParameterType parameterType, double value | void | Присваивает параметру определенного типа новое значение. |
| ValidateNeckHeight | - | void | Валидирует высоту горлышка. |
| ValidateNeckRadius | - | void | Валидирует радиус горлышка. |

Класс SketchBuilder является статическим и содержит в себе методы для построения эскизов.

В классе SketchBuilder используются класс Region, представляющий собой замкнутый контур. Он является классом из API ObjectARX. Его описание содержится в описании API.

В классе SketchBuilder используются класс Vector3d, представляющий собой вектор. Он является классом из API ObjectARX. Его описание содержится в описании API.

В классе SketchBuilder используются класс Point3d, представляющий собой точку в трехмерном пространстве. Он является классом из API ObjectARX. Его описание содержится в описании API.

Таблица 6.5 – Методы класса SketchBuilder

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| CreateCircle | Point3d center, double radius | Region | Создает эскиз в виде окружности. |
| CreateRectangle | double width,  double length, Point3d center | Region | Создает эскиз в виде прямоугольника. |

Класс Builder является статическим и содержит в себе методы для построения частей модели.

В классе используются класс Solid3d, представляющий собой замкнутый контур. Он является классом из API ObjectARX. Его описание содержится в описании API.

Таблица 6.6 – Методы класса Builder

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| BuildMain | Parameter mainWidth, Parameter mainLength, Parameter mainHeight | Solid3d | Строит прямоугольную основную часть бутылки. |
| BuildMain | Parameter mainRadius, Parameter mainHeight | Solid3d | Строит круглую основную часть бутылки. |
| BuildNeck | Parameter neckHeight, Parameter neckRadius, Parameter mainHeight | Solid3d | Строит круглое горлышко бутылки. |

Продолжение таблицы 6.6

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| BuildNeck | Parameter neckHeight, Parameter neckLength, Parameter neckWidth, Parameter mainHeight | Solid3d | Строит прямоугольное горлышко бутылки. |
| BuildShellSolid | Parameter neckLength, Parameter neckWidth, Parameter mainHeight | Solid3d | Строит прямоугольный объект Solid3d для вырезания отверстия в основной части. |
| BuildShellSolid | Parameter neckRadius, Parameter mainHeight | Solid3d | Строит круглый объект Solid3d для вырезания отверстия в основной части. |

Класс MainFrom является формой, через которую пользователь взаимодействует с программой.

Таблица 6.7 – Методы класса MainForm

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| CheckOnErrors | - | bool | Проверяет форму на наличие ошибок. |
| ClearForm | - | void | Очищает форму. |
| OnTextChanged | object sender, EventArgs e | void | Обработчик события ввода текста в текстбокс. |

Класс Wrapper является статическим и содержит в себе методы для запуска формы и создания транзакции.

Таблицы 6.8 – Методы класса Connector

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Входные параметры | Тип возвращаемых данных | Описание |
| OpenApp | - | void | Запускает приложение. |

При реализации системы были сделаны следующие архитектурные изменения:

1. Wrapper/Connector

* Класс Wrapper переименован в Connector, так как Wrapper неправильно описывает функциональность данного класса. Класс просто открывает форму и не оборачивает никаких методов;
* Удален метод OpenMainForm, так как логика в данном методе не является переиспользуемой;
* В классе Connector больше не принимается объект Transaction, так как данный объект можно создать непосредственно при создании модели в классе Builder.

1. MainForm

* Удалено поле \_transaction и соответствующее ему свойство Transaction, т.к. форма больше не принимает объект транзакции извне;
* Удален метод BuildModel(Transaction), т.к. форма больше не принимает объект транзакции извне;
* Удален метод CheckTextBoxType(TextBox), т.к. был найден более эффективный способ определения типа текстбокса (элемента управления).

1. Parameters

* Был переименовано поле \_parameters в \_parameterDictionary, т.к. новое название точнее описывает объект;
* Был добавлен словарь списков исключений для каждого параметра. Это необходимо для совместной обработки исключений, выбрасываемых при валидации вхождения параметра в допустимый диапазон, и исключений, выбрасываемых при валидации зависимых параметров.
* Добавлены авто-свойства IsMainCircle и IsNeckRectangle, т.к. с реализацией дополнительной функциональности нужно следить за выбранной формой основной части и формой горлышка;
* Удалено свойство Parameters, так как нет необходимости напрямую присваивать параметрам значения;
* Добавлена перегрузка параметра [], т.к. появилась потребность в получении отдельных параметров
* Добавлен метод SetValue, присваивающий новое значение параметру и валидирующий это значение. Не реализован как свойство, т.к. в таком случае не была бы возможна валидация зависимых параметров;
* Общий метод валидации зависимых параметров был разделен на несколько методов ValidateNeckHeight и ValidateNeckRadius, отдельно валидирующих каждый зависимый параметр. Это необходимо для правильного сбора всех исключений, выбрасываемых при валидации.

1. ParameterType/BottleParameterType

* Изменено название перечисления, т.к. новое уточняет типы параметров какой модели содержатся в этом перечислении;
* Были добавлены значения для параметров дополнительной функциональности.

1. Builder

* Были добавлены перегрузки для методов построения основной части и горлышка, т.к. из-за дополнительной функциональности появилась необходимость построения основной части и горлышка различных форм;
* Были добавлены отдельные методы для создания объектов `Solid3d` для вырезания отверстий в основной части;
* Метод построения бутылки BuildBottle больше не принимает объект Transaction, т.к. объект транзакции может быть создан непосредственно внутри метода.

1. SketchBuilder

* Метод CreateCircle не принимает объект Vector3d, т.к. нет необходимости в построении окружностей в разных плоскостях;
* Метод CreateRectangle стал принимать третьим параметром центр, относительно которого будет строиться прямоугольник, т.к. возникла необходимость создания нескольких прямоугольников с различными координатами.

1. Parameter

* Конструктор теперь принимает только два значения: минимальное и максимальное значение, которое может принимать параметр;
* Добавлен метод ReturnToDefaultValue, который присваивает методу значение, равное минимальному возможному для данного параметра. Это нужно для правильной обработки исключений при отключении некоторых параметров при выборе формы горлышка и основной части;
* Свойства MinValue и MaxValue были сделаны авто-свойствами, т.к. методах get и set нет никакой дополнительной логики, кроме присваивания значения.

# 7 ОПИСАНИЕ ПРОГРАММЫ ДЛЯ ПОЛЬЗОВАТЕЛЯ

Для начала работы с плагином пользователь должен загрузить файл библиотеки коннектора. После загрузки файла пользователю станет доступна команда BOTTLEBUILDER, которая запускает плагин.

При открытии формы пользователю становится доступен ввод данных в элементы управления. Пользовательский интерфейс представлен на рисунке 7.1.

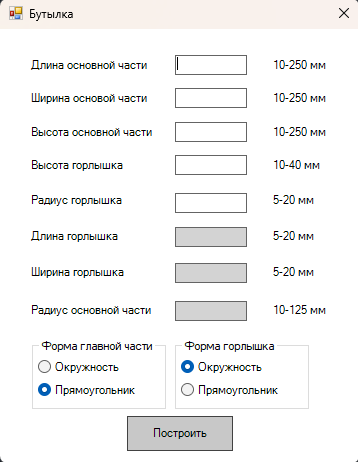


Рисунок 7.1 – Пользовательский интерфейс плагина

Сначала пользователь должен выбрать какой формы должны быть основная часть и горлышко. В зависимости от выбора будут доступны текстовые поля различных параметров.

Далее пользователь должен ввести параметры в доступные текстовые поля. Плагин не позволяет вводить в текстовые поля символы, которые не являются числами или управляющими символами. Кроме того, плагин проверяет введенные числа на вхождение в допустимый диапазон значений и соответствие дополнительным условиям (для зависимых параметров).

При вводе значений, не прошедших валидацию, соответствующее текстовое поле окрашивается красным и при попытке построить модель плагин не позволит этого сделать и покажет окно с текстом ошибок.

При вводе всех необходимых значений и нажатии кнопки “Построить” все поля плагина очистятся и модель построится в файле, в котором была загружена библиотека с плагином.

# 8 ТЕСТИРОВАНИЕ ПЛАГИНА

## 8.1 Функциональное тестирование

Вывод различных сообщений об ошибке показан на рисунках 8.1 – 8.3.

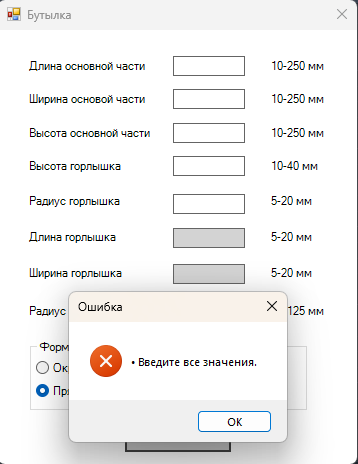


Рисунок 8.1 – Вывод сообщения при пустых текстовых полях

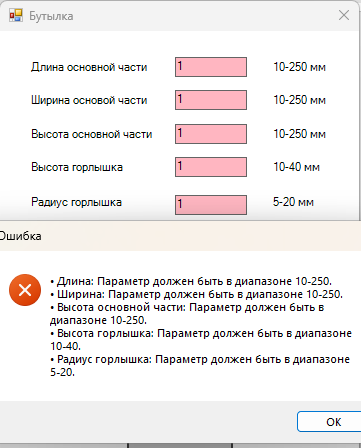


Рисунок 8.2 – Вывод сообщения при значениях, не входящих в допустимый диапазон

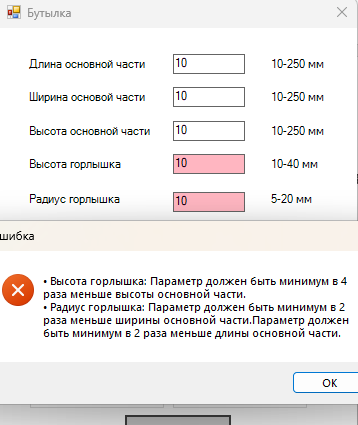


Рисунок 8.3 – Вывод сообщения при значениях, не подходящих дополнительным условиям

Успешный результат работы плагина со стандартными значениями параметров представлен на рисунках 8.4 – 8.7.

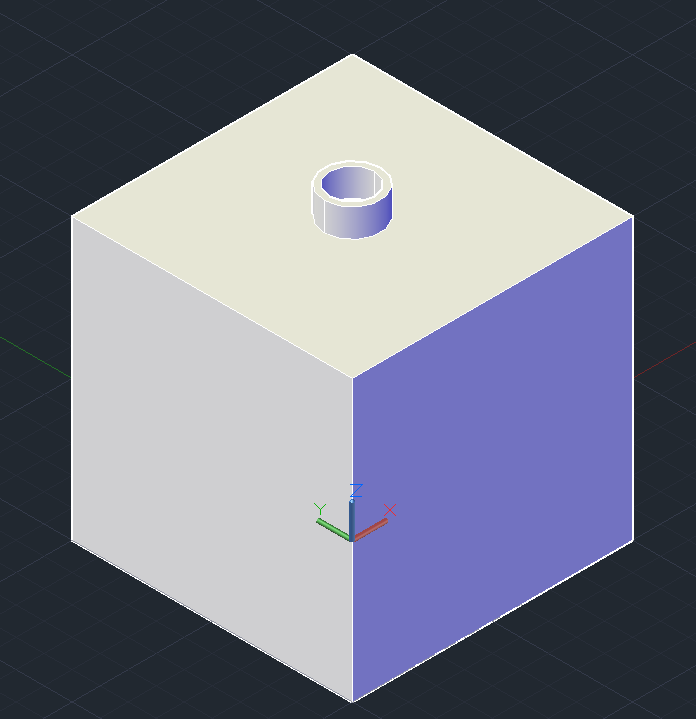


Рисунок 8.4 – Модель бутылки с круглым горлышком и прямоугольной основной частью

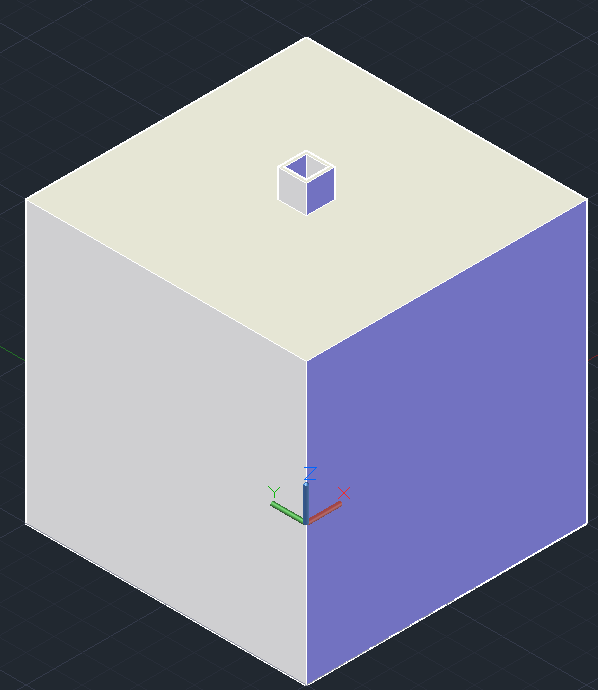


Рисунок 8.5 – Модель бутылки с прямоугольным горлышком и прямоугольной основной частью

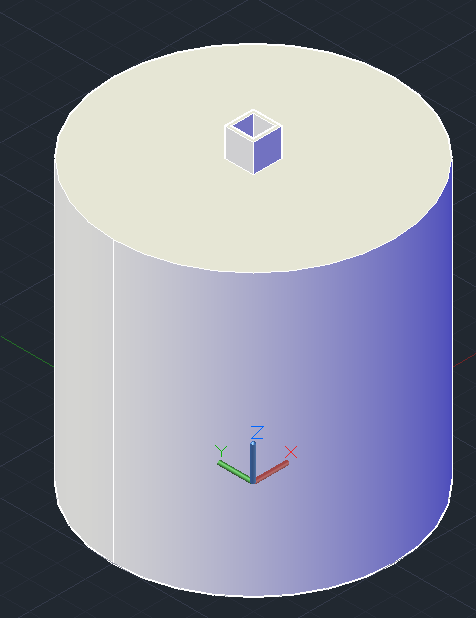


Рисунок 8.6 – Модель бутылки с прямоугольным горлышком и круглой основной частью

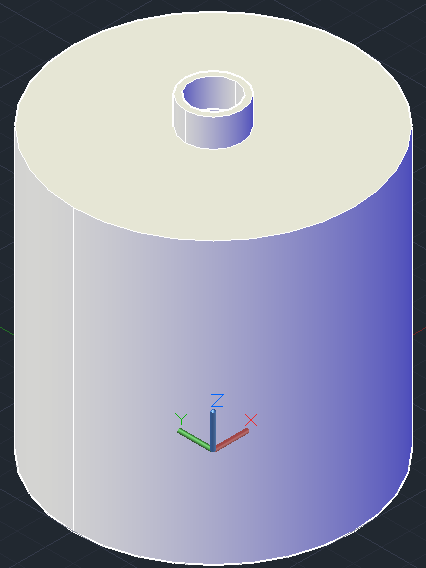


Рисунок 8.7 – Модель бутылки с круглым горлышком и круглой основной частью

Успешный результат работы плагина с минимальными значениями параметров представлен на рисунках 8.8 – 8.11.

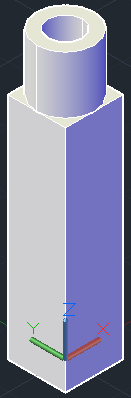


Рисунок 8.8 – Модель бутылки с круглым горлышком и прямоугольной основной частью

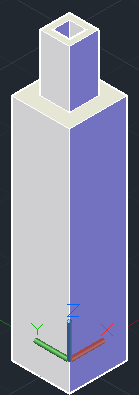


Рисунок 8.9 – Модель бутылки с прямоугольным горлышком и прямоугольной основной частью

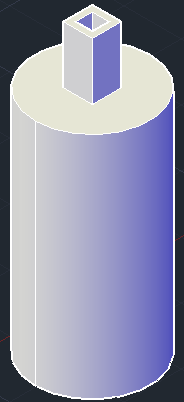


Рисунок 8.10 – Модель бутылки с прямоугольным горлышком и круглой основной частью

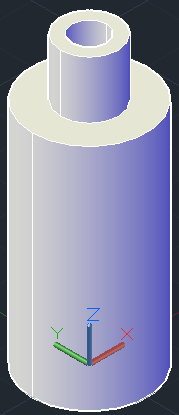


Рисунок 8.11 – Модель бутылки с круглым горлышком и круглой основной частью

Успешный результат работы плагина с максимальными значениями параметров представлен на рисунках 8.12 – 8..

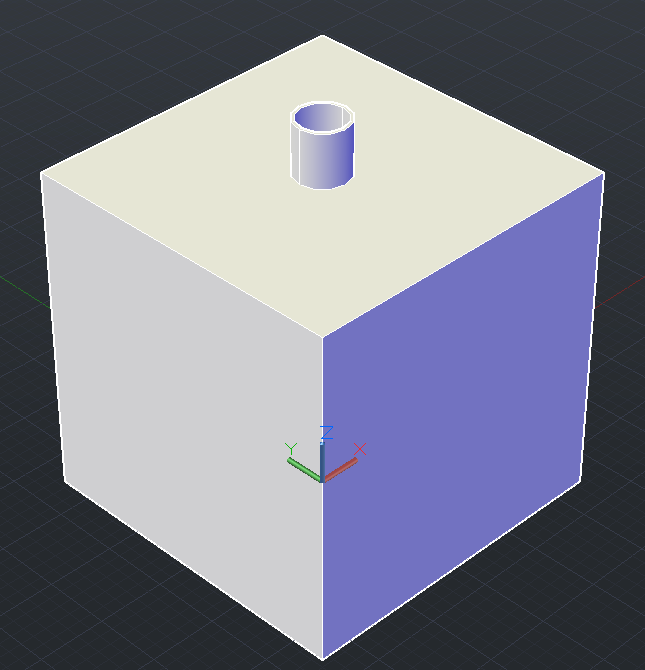


Рисунок 8.12 – Модель бутылки с круглым горлышком и прямоугольной основной частью

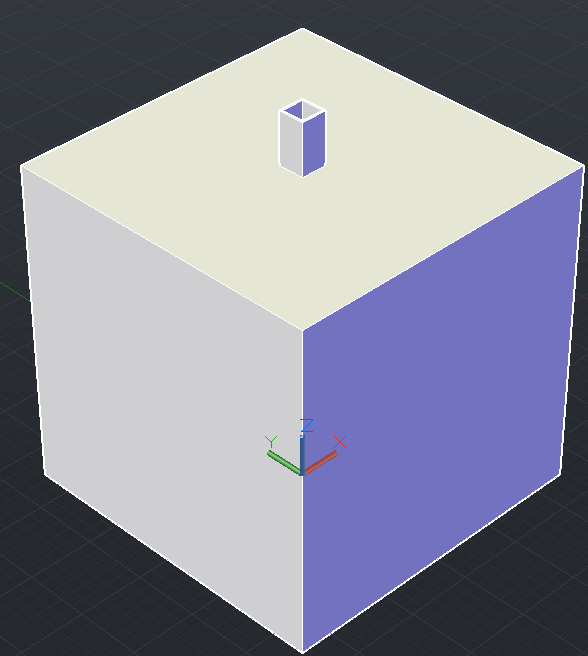


Рисунок 8.13 – Модель бутылки с прямоугольным горлышком и прямоугольной основной частью

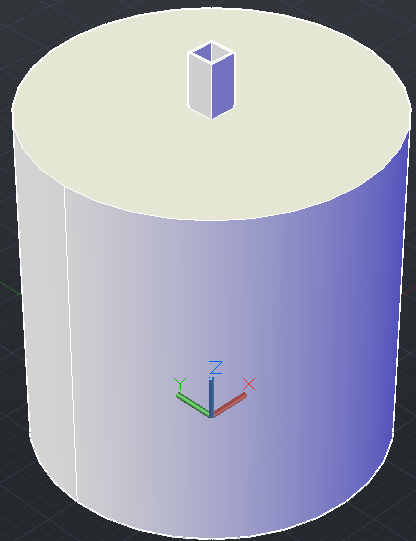


Рисунок 8.14 – Модель бутылки с прямоугольным горлышком и круглой основной частью

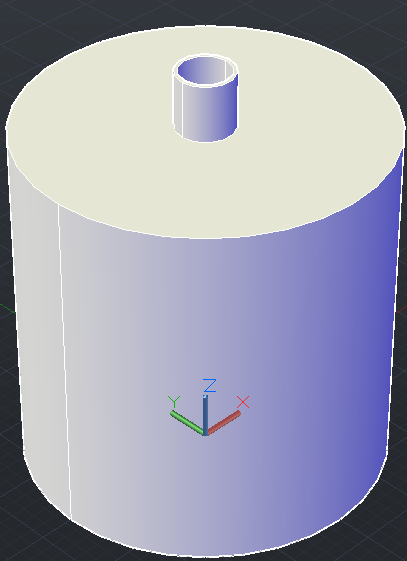


Рисунок 8.15 – Модель бутылки с круглым горлышком и круглой основной частью

## 8.2 Модульное тестирование

В рамках модульного тестирования было написано 35 тестов. Обозреватель юнит-тестов представлен на рисунке 8.16.

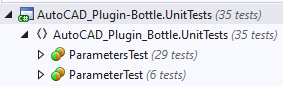


Рисунок 8.16 – Обозреватель юнит-тестов

В таблицах 8.1 и 8.2 представлена информация о написанных юнит-тестах.

Таблица 8.1 – Юнит-тесты класса Parameter

|  |  |
| --- | --- |
| Название теста | Описание |
| MaxValueProperty | Тест свойства MaxValue |
| MinValueProperty | Тест свойства MinValue |
| ReturnToDefaultValueTest | Тест метода ReturnToDefaultValue |
| ValuePropertyFailureLess | Тест присваивания значения, меньшего, чем допустимое минимальное значение |
| ValuePropertyFailureMore | Тест присваивания значения, большего, чем допустимое максимальное значение |
| ValuePropertySuccess | Успешный тест свойства Value |

Таблица 8.2 – Юнит-тесты класса Parameters

|  |  |
| --- | --- |
| Название теста | Описание |
| BracketsOperatorLength | Тест перегрузки оператора [] для параметра длины основной части. |
| BracketsOperatorMainHeight | Тест перегрузки оператора [] для параметра высоты основной части. |
| BracketsOperatorMainRadius | Тест перегрузки оператора [] для параметра радиуса основной части. |
| BracketsOperatorNeckHeight | Тест перегрузки оператора [] для параметра высоты горлышка. |

Продолжение таблицы 8.2

|  |  |
| --- | --- |
| Название теста | Описание |
| BracketsOperatorNeckLength | Тест перегрузки оператора [] для параметра длины горлышка. |
| BracketsOperatorNeckRadius | Тест перегрузки оператора [] для параметра радиуса горлышка. |
| BracketsOperatorNeckWidth | Тест перегрузки оператора [] для параметра ширины горлышка. |
| BracketsOperatorWidth | Тест перегрузки оператора [] для параметра ширины основной части. |
| IsMainCircleProperty | Тест свойства IsMainCircle. |
| IsNeckRectangleProperty | Тест свойства IsNeckRectangle. |
| SetValueFailureLength | Тест метода SetValue (провал, длина основной части). |
| SetValueFailureMainHeight | Тест метода SetValue (провал, высота основной части). |
| SetValueFailureMainRadius | Тест метода SetValue (провал, радиус основной части). |
| SetValueFailureNeckHeightOne | Тест метода SetValue (провал, высота горлышка, соответствует зависимым параметрам). |
| SetValueFailureNeckHeightTwo | Тест метода SetValue (успех, высота горлышка, не соответствует зависимым параметрам). |
| SetValueFailureNeckRadiusOne | Тест метода SetValue (провал, радиус горлышка, не соответствует зависимым параметрам, ширина – успех, длина – провал). |
| SetValueFailureNeckRadiusTwo | Тест метода SetValue (провал, радиус горлышка, не соответствует зависимым параметрам, ширина – провал, длина – успех). |
| SetValueFailureNeckRadiusThree | Тест метода SetValue (провал, радиус горлышка, не соответствует зависимым параметрам, ширина – провал, длина – провал). |
| SetValueFailureNeckWidth | Тест функции SetValue (провал, ширина горлышка). |
| SetValueFailureWidth | Тест функции SetValue (провал, ширина основной части). |

Продолжение таблицы 8.3

|  |  |
| --- | --- |
| Название теста | Описание |
| SetValueSuccessLength | Тест функции SetValue (успех, длина основной части). |
| SetValueSuccessMainHeight | Тест функции SetValue (успех, высота основной части). |
| SetValueSuccessMainRadius | Тест функции SetValue (успех, радиус основной части). |
| SetValueSuccessNeckHeight | Тест функции SetValue (успех, высота горлышка). |
| SetValueSuccessNeckLength | Тест функции SetValue (успех, длина горлышка). |
| SetValueSuccessNeckRadius | Тест функции SetValue (успех, радиус горлышка). |
| SetValueSuccessNeckWidth | Тест функции SetValue (успех, ширина горлышка). |
| SetValueSuccessWidth | Тест функции SetValue (успех, ширина основной части). |

На рисунке 8.16 представлен процент покрытия проекта юнит-тестами.

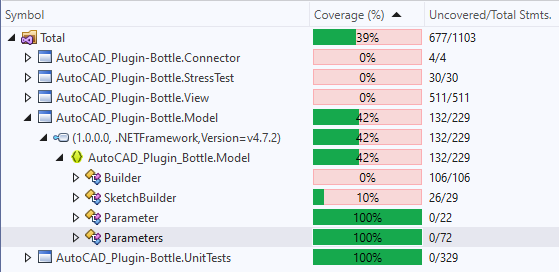


Рисунок 8.16 – Процент покрытия проекта юнит-тестами

## 8.3 Нагрузочное тестирование

Нагрузочное тестирование заключалось в построении трехмерной модели бутылки со средними параметрами:

* длина основной части L – 120 мм;
* ширина основной части W – 120 мм;
* высота основной части H1 – 120 мм;
* высота горлышка H2 – 15 мм;
* радиус горлышка R – 7 мм;

Данное тестирование заняло около 11 минут, по итогам которого было построено 40000 3D-моделей бутылки.

На рисунке 8.17 представлен график зависимости памяти ОЗУ от построения модели, а на рисунке 8.18 представлен график зависимости времени от построения модели.

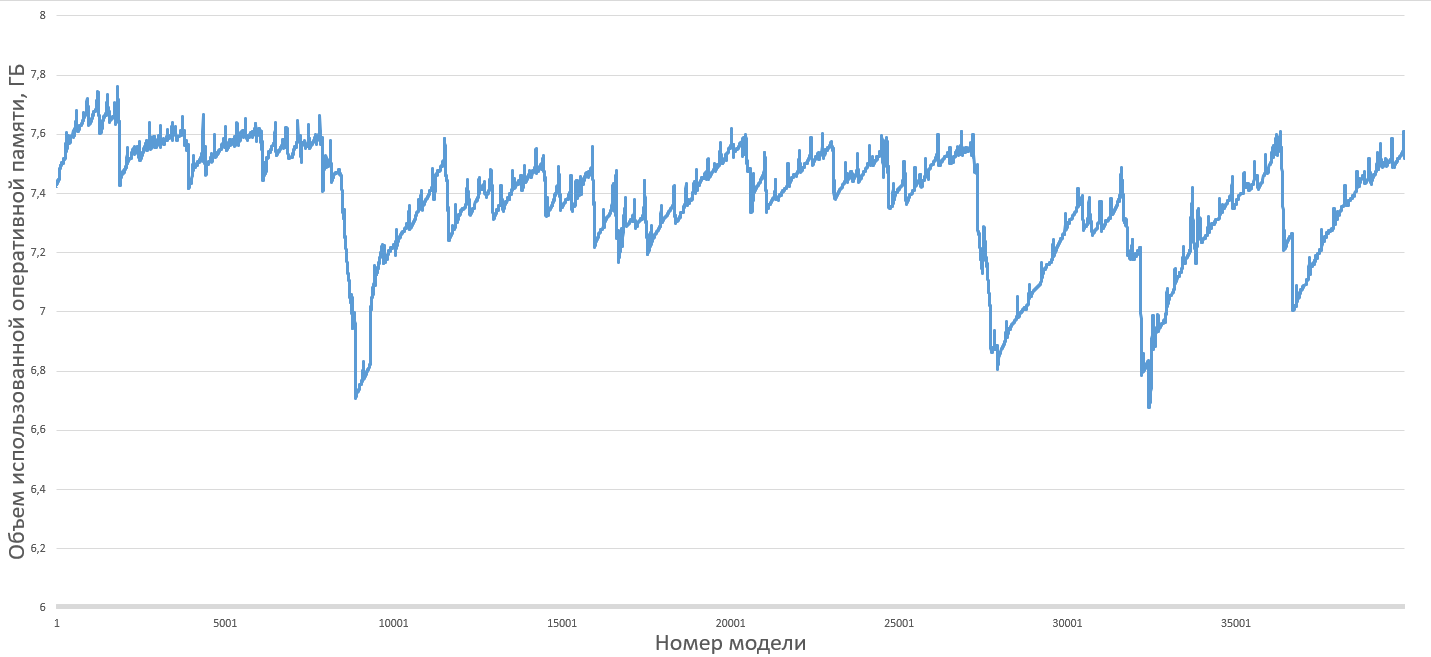


Рисунок 8.17 – График зависимости памяти ОЗУ от количества построенных моделей

Проанализировав график 8.17 можем сделать вывод, что объем оперативной памяти, затрачиваемый плагином на построение трехмерных моделей бутылки, линейно увеличивается до достижения предела объема оперативной памяти. По окончании свободного места оперативная память частично очищается, после чего данный процесс повторяется. Предположительно это объясняется использованием файла подкачки для компенсации недостатка оперативной памяти. После достижения первого максимума оперативная память уже больше не достигает таких значений и очищается на уровне 7.6 ГБ.

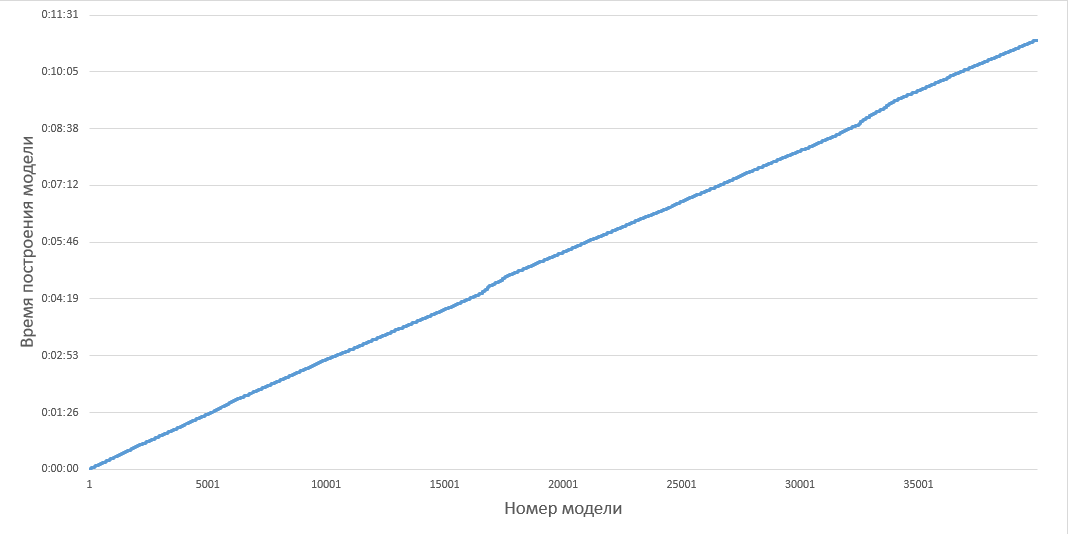


Рисунок 8.18 – График зависимости времени от построения модели

Проанализировав график 8.18 можем сделать вывод, что время построения моделей изменяется линейно. Это может быть объяснено тем, что модель бутылки достаточно проста и не содержит в себе сложных вычислений, из-за чего даже при значительных количествах построенных моделей их нагрузка на оперативную память и центральный процессор не достаточна для того, чтобы как-то замедлить построение.

# Заключение

В процессе разработки приложения был создан плагин, позволяющий создавать 3D-модели бутылок различной формы.

Для разработки были изучены новые библиотеки: Resharper, Stylecop.

При написании плагина мной был получен опыт использования новых библиотек, проведения нагрузочного тестирования и разработки дополнительной функциональности по требованию заказчика.

# Список использованных источников

1. Introduction to AutoCAD [Электронный ресурс]: сайт Iowa State University. URL: <https://iastate.pressbooks.pub/visualgraphiccomm/chapter/chapter-1/> (дата обращения: 21.10.2023).
2. What is API? [Электронный ресурс]: официальный сайт Postman. URL: [https://www.postman.com/what-is-an-api/](https://vk.com/away.php?to=https%3A%2F%2Fwww.postman.com%2Fwhat-is-an-api%2F&cc_key=) (дата обращения: 21.10.2023).
3. Wine Bottle [Электронный ресурс]: официальный сайт разработчика. URL: [http://www.nxtrender.com/wk/Wine\_Bottle.htm](https://vk.com/away.php?to=http%3A%2F%2Fwww.nxtrender.com%2Fwk%2FWine_Bottle.htm&cc_key=) (дата обращения: 21.10.2023).
4. Буч, Г. Язык UML. Руководство пользователя. 2-е изд. [Текст]/Г. Буч, Д. Рамбо, И. Якобсон. – М.: ДМК Пресс, 2006. – 496 с.