КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

**по учебной практике УП 02 МДК 02.01 «Поддержка и тестирование программных модулей»**

**Специальность 09.02.07**

**«Информационные системы и программирование»**

**Специализация:**

**«Программист»**

Студент группы 493:

Лукьянов И. А.

Преподаватель: Полякова А.Н.

Санкт-Петербург 2023

**СОДЕРЖАНИЕ**

[1. Задание 3](#_Toc128649736)

[1.1. Разработки библиотеки классов 3](#_Toc128649737)

[1.2. Класс расчета материалов 3](#_Toc128649738)

[1.3. Разработка модульных тестов (Unit-tests) 4](#_Toc128649739)

[2. Ход работы 4](#_Toc128649740)

[2.1. Метод для расчета количества сырья 5](#_Toc128649741)

[2.2. Модульные тесты 5](#_Toc128649742)

# Задание

## Разработки библиотеки классов

Для того чтобы в производстве могли быстро и одинаково рассчитывать количество необходимого сырья для производства той или иной продукции, необходимо разработать библиотеку классов.

Данная библиотека будет подключаться к основному проекту и должна быть представлена в виде .dll/.jar файла или папки с файлом .py.

Чтобы система правильно интегрировалась вам необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

В качестве названия для библиотеки необходимо использовать: WSUniversalLib. Вам необходимо загрузить исходный код проекта с библиотекой в отдельный репозиторий с названием, совпадающим с названием проекта.

## Класс расчета материалов

Метод должен рассчитывать целое количество сырья, необходимого для производства определенного количества (count) продукции, учитывая возможный брак материалов. Для упрощения расчетов будем считать всю продукцию прямоугольного размера с известными значениями ширины (width) и длины (length).

Количество необходимого качественного сырья на одну единицу продукции рассчитывается как площадь продукции, умноженная на коэффициент типа продукции.

Коэффициенты типа продукции (product\_type):

Тип продукции 1 - 1.1,

Тип продукции 2 - 2.5,

Тип продукции 3 - 8.43.

При этом нужно учитывать процент брака материала в зависимости от его типа (material\_type):

Тип материала 1 - 0.3%,

Тип материала 2 - 0.12%.

При этом если в качестве параметров метода будут приходить несуществующие типы продукции/материалов или другие неподходящие данные, то метод должен вернуть -1.

Например, необходимо изготовить 15 единиц продукции 3 типа шириной 20 и длиной 45 из материала 1 типа. Количество качественного сырья (без учета брака) будет равно 113 805. Однако с учетом возможного брака материалов общее необходимое количество сырья должно быть увеличено до 114 147,442. Округлив полученное значение до ближайшего большего целого, получим 114 148 единиц необходимого сырья. Спецификация метода представлена в отдельном файле в ресурсах.

## Разработка модульных тестов (Unit-tests)

Для выполнения процедуры тестирования созданного вами метода библиотеки WSUniversalLib, возвращающего целое количество сырья для производства, вам необходимо создать отдельный проект модульных тестов.

В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Ничего страшного, если ваш метод работает не совсем идеально и тесты могут быть не пройдены в связи с этим - в данном модуле это не так важно.

Обратите внимание, что имена тестов должны отражать их суть, т.е. вместо TestMethod1() тест следует назвать, например, GetQuantityForProduct\_NonExistentProductType() для тестирования случая передачи несуществующего типа продукции.

Необходимо разработать модульные тесты, которые на основании исходных данных можно условно разделить на 2 группы следующим образом: 10 методов низкой сложности и 5 методов высокой сложности.

# Ход работы

## Метод для расчета количества сырья

Было разработано 10 модульных тестов для тестирования созданного метода библиотеки WSUniversalLib.

На рисунке 1 изображен метод для расчета количества сырья.

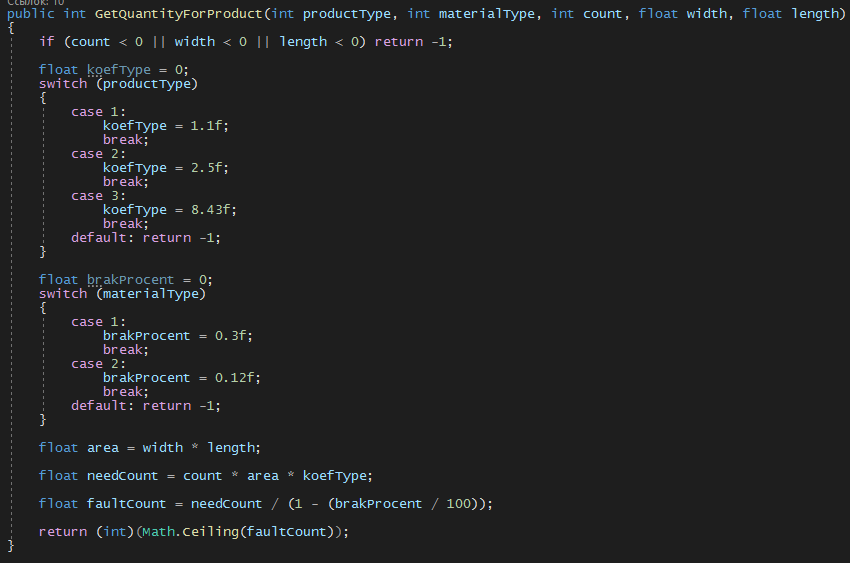


Рисунок 1 – Код метода для расчета количества сырья

## Модульные тесты

* + 1. Тестирование правильности расчетов

Данный тест проверят правильность расчетов в методе при корректных введённых данных. Код данного метода изображен на рисунке 2.

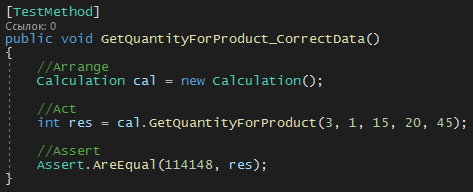


Рисунок 2 – Метод для тестирования правильности расчетов

* + 1. Тестирование несуществующего типа продукта

Данный тест проверят метод на обработку ввода несуществующего типа продукта. Код данного метода изображен на рисунке 3.

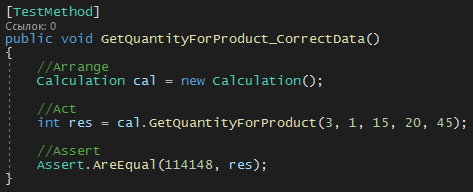


Рисунок 3 – Метод для тестирования несуществующего типа продукта

* + 1. Тестирование несуществующего типа материала

Данный тест проверят метод на обработку ввода несуществующего типа материала. Код данного метода изображен на рисунке 4.

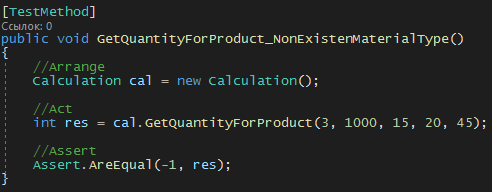


Рисунок 4 – Метод для тестирования несуществующего типа материала

* + 1. Тестирование отрицательного количества продукции

Данный тест проверяет метод на обработку ввода отрицательного количества продукции. Код данного метода изображен на рисунке 5.

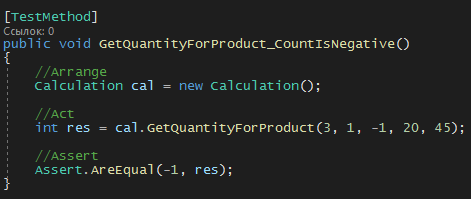


Рисунок 5 – Метод для тестирования отрицательного количества продукции

* + 1. Тестирование отрицательной ширины

Данный тест проверяет метод на обработку ввода отрицательной ширины. Код данного метода изображен на рисунке 6.

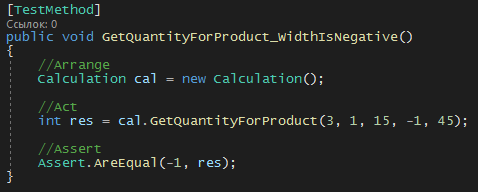


Рисунок 6 – Метод для тестирования отрицательной ширины

* + 1. Тестирование отрицательной длины

Данный тест проверяет метод на обработку ввода отрицательной длины. Код данного метода изображен на рисунке 7.

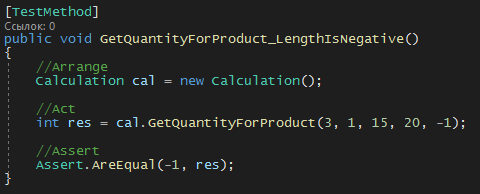


Рисунок 7 – Метод для тестирования отрицательной длины

* + 1. Тестирование округления к наименьшему целому

Данный тест проверяет метод на округление количества с учетом погрешности брака к меньшему целому. Код данного метода изображен на рисунке 8.

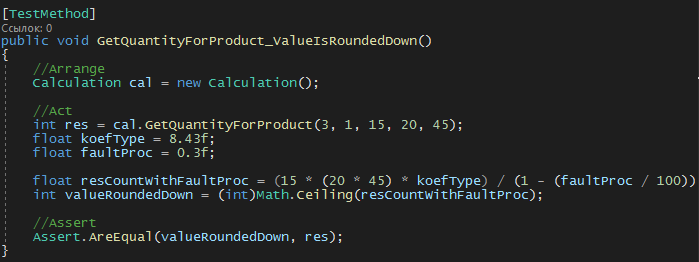


Рисунок 8 – Метод для тестирования округления к наименьшему целому

* + 1. Тестирование нулевого количества

Данный тест проверяет метод на обработку ввода нулевого количества. Код данного метода изображен на рисунке 9.

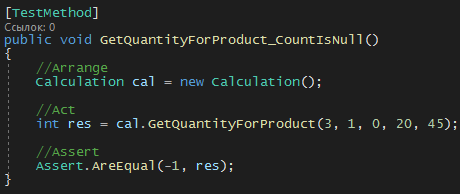


Рисунок 9 – Метод для тестирования нулевого количества

* + 1. Тестирование нулевой ширины

Данный тест проверяет метод на обработку ввода нулевой ширины. Код данного метода изображен на рисунке 10.

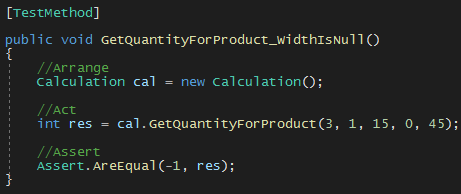


Рисунок 10 – Метод для тестирования нулевой ширины

* + 1. Тестирование нулевой длины

Данный тест проверяет метод на обработку ввода нулевой длины. Код данного метода изображен на рисунке 11.

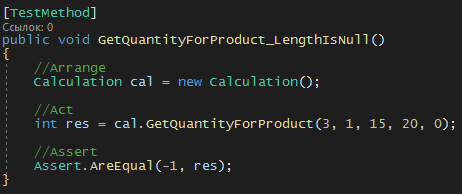


Рисунок 11 – Метод для тестирования нулевой количества

* + 1. Успешность методов тестирования

На рисунке 12 показан результат выполнения всех методов тестирования.

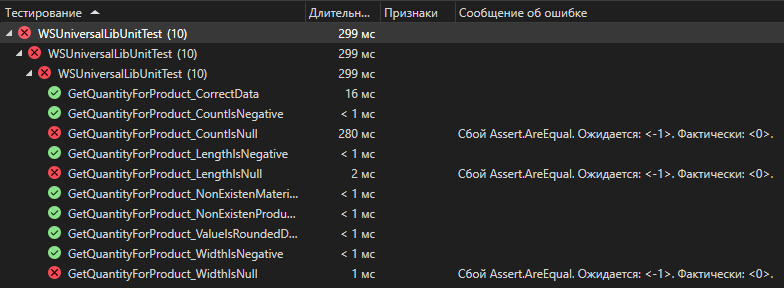


Рисунок 12 – Результат выполнения методов тестирования

## Тестирование «Test-Case»

* + 1. Тестовый пример 1

На рисунке 13 показан первый тестовый пример.

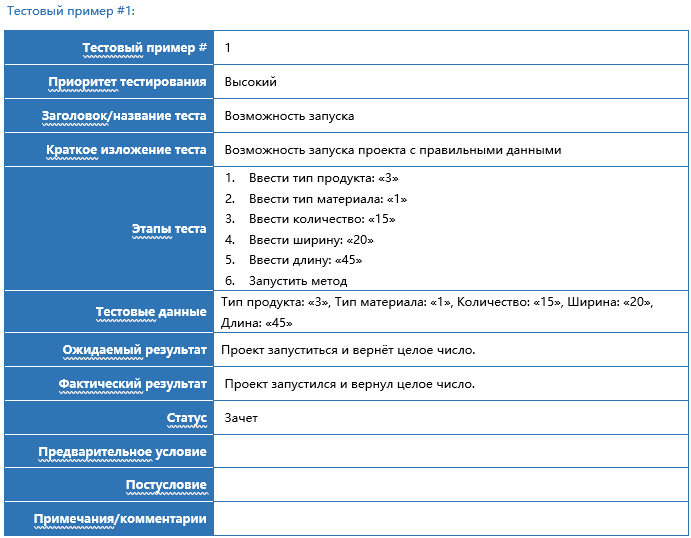


Рисунок 13 – Первый тестовый пример

* + 1. Тестовый пример 2

На рисунке 14 показан второй тестовый пример.

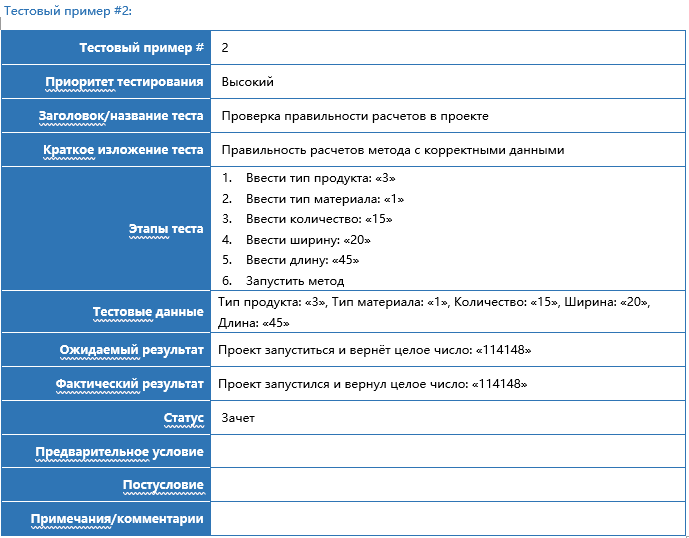


Рисунок 14 – Второй тестовый пример

* + 1. Тестовый пример 3

На рисунке 15 показан третий тестовый пример.

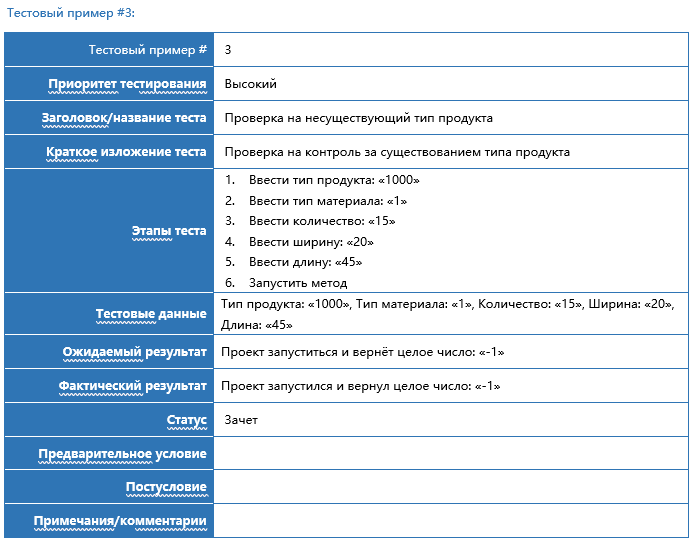


Рисунок 15 – Третий тестовый пример

* + 1. Тестовый пример 4

На рисунке 16 показан четвертый тестовый пример.

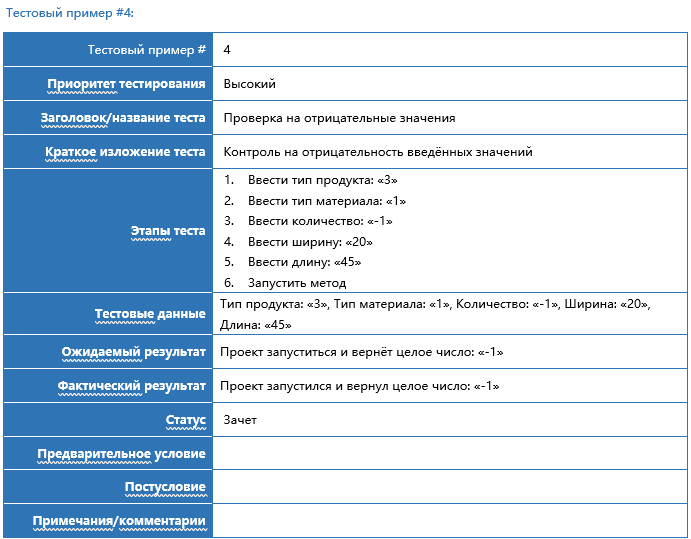


Рисунок 16 – Четвертый тестовый пример

* + 1. Тестовый пример 5

На рисунке 17 показан пятый тестовый пример.

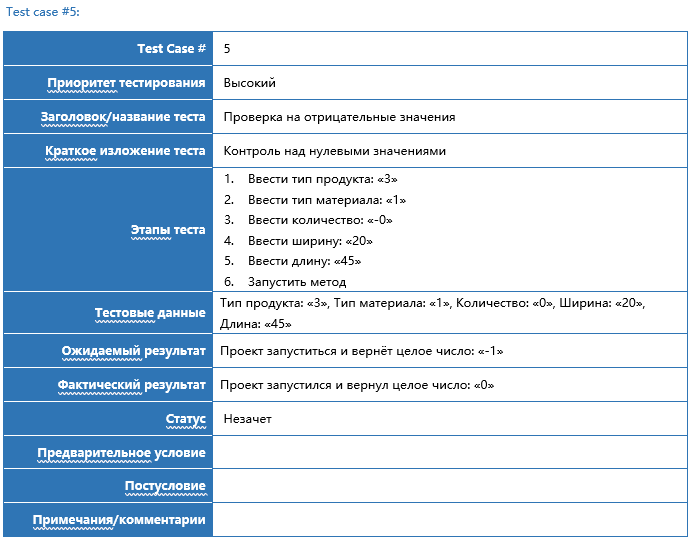


Рисунок 17 – Пятый тестовый пример