КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное

бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

**ОТЧЕТ**

**по учебной практике МДК 01.02 «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. ПРАКТИЧЕСКАЯ РАБОТА №1 3](#_Toc129105352)

[1.1. ЗАДАЧА 3](#_Toc129105353)

[1.2. ХОД РАБОТЫ 5](#_Toc129105357)

[1.2.1 Метод для расчета количества сырья 5](#_Toc129105358)

[1.2.2. Модульные тесты 5](#_Toc129105359)

[1.2.3. Тестирование «Test-Case» 9](#_Toc129105371)

# ПРАКТИЧЕСКАЯ РАБОТА №1

ЗАДАЧА

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.

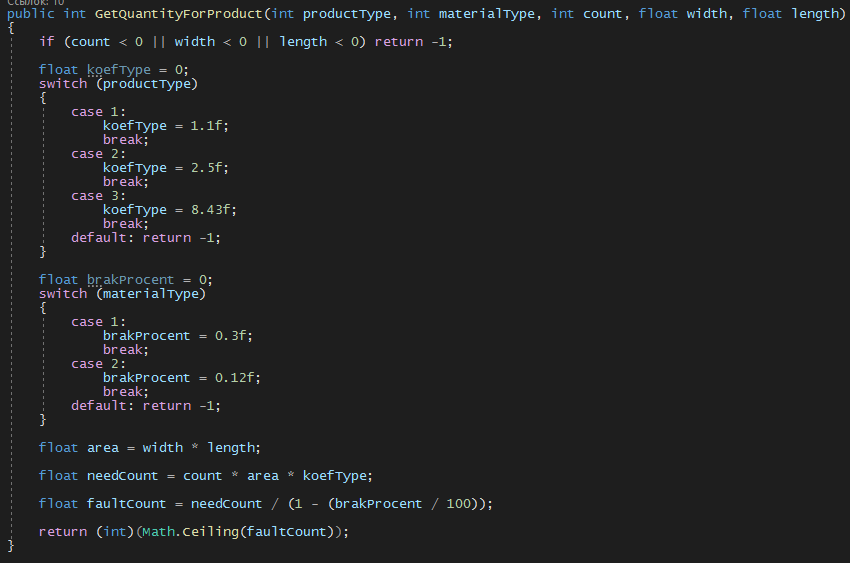


Рисунок 1 – Код метода для расчета количества сырья

* + 1. Модульные тесты

Тестирование правильности расчетов

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

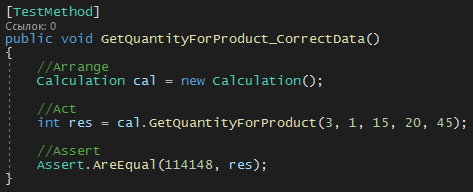


Рисунок 2 – Метод для тестирования правильности расчетов

Тестирование несуществующего типа продукта

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

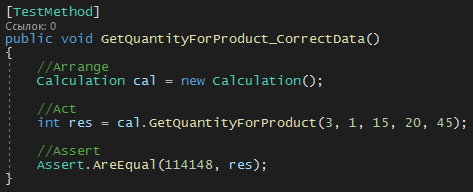


Рисунок 3 – Метод для тестирования несуществующего типа продукта

Тестирование несуществующего типа материала

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

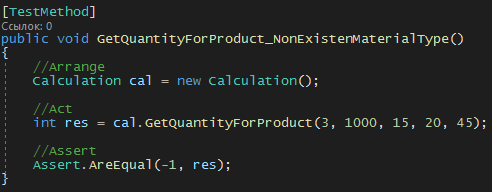


Рисунок 4 – Метод для тестирования несуществующего типа материала

Тестирование отрицательного количества продукции

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

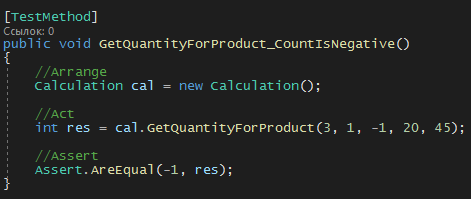


Рисунок 5 – Метод для тестирования отрицательного количества продукции

Тестирование отрицательной ширины

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

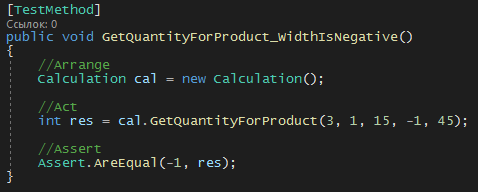


Рисунок 6 – Метод для тестирования отрицательной ширины

Тестирование отрицательной длины

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

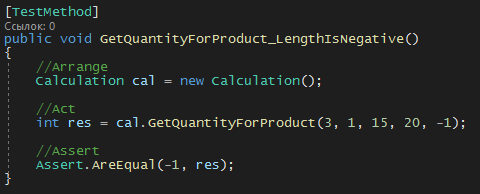


Рисунок 7 – Метод для тестирования отрицательной длины

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.

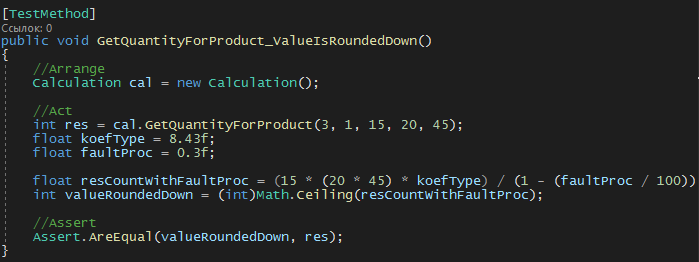


Рисунок 8 – Метод для тестирования округления к наименьшему целому

Тестирование нулевого количества

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

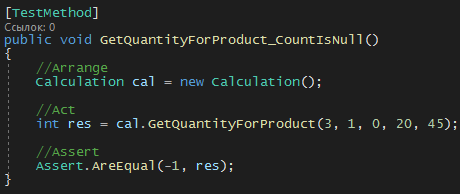


Рисунок 9 – Метод для тестирования нулевого количества

Тестирование нулевой ширины

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

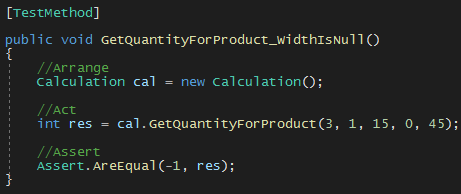


Рисунок 10 – Метод для тестирования нулевой ширины

Тестирование нулевой длины

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

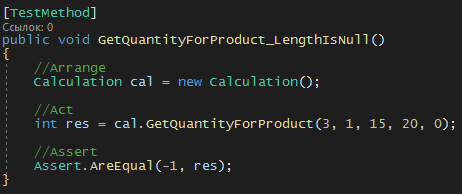


Рисунок 11 – Метод для тестирования нулевой количества

Успешность методов тестирования

На рисунке 12 показан результат выполнения всех методов тестирования.

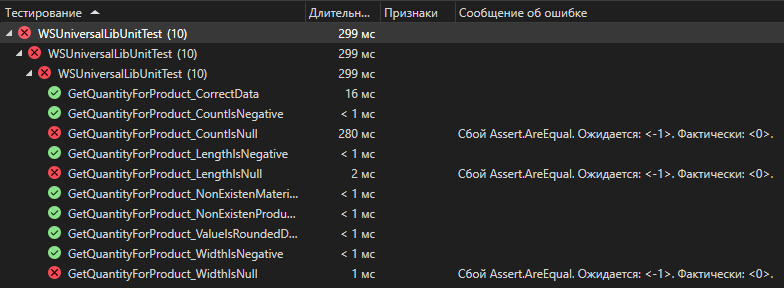


Рисунок 12 – Результат выполнения методов тестирования

## Тестирование «Test-Case»

Тестовый пример 1

На рисунке 13 показан первый тестовый пример.

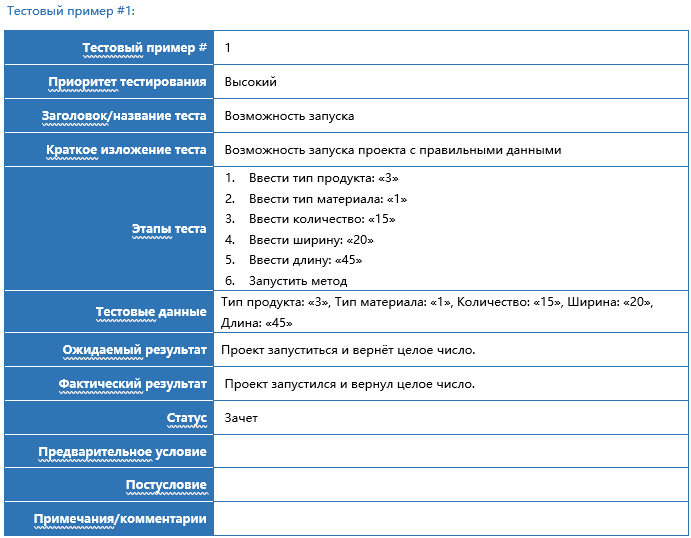


Рисунок 13 – Первый тестовый пример

Тестовый пример 2

На рисунке 14 показан второй тестовый пример.

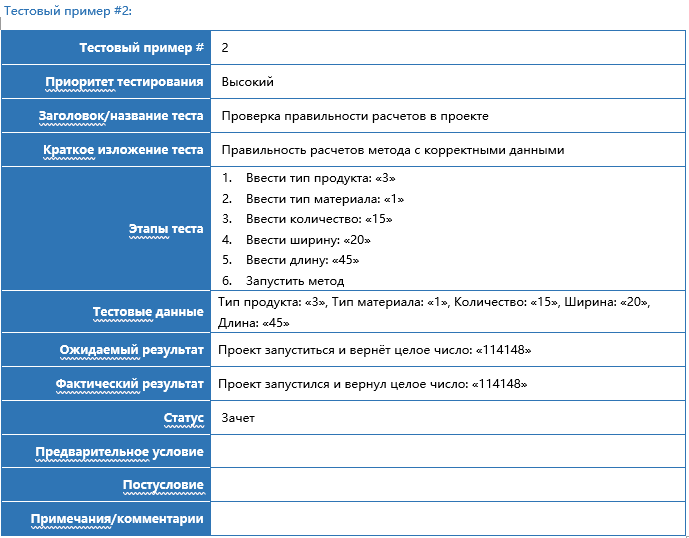


Рисунок 14 – Второй тестовый пример

Тестовый пример 3

На рисунке 15 показан третий тестовый пример.

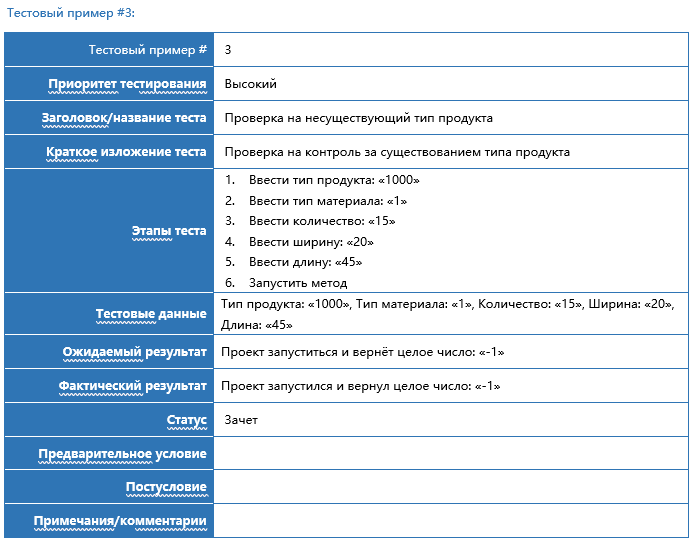


Рисунок 15 – Третий тестовый пример

Тестовый пример 4

На рисунке 16 показан четвертый тестовый пример.

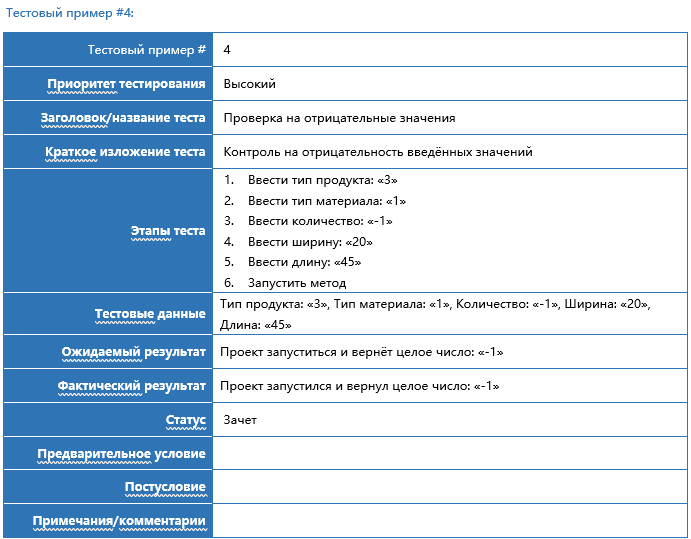


Рисунок 16 – Четвертый тестовый пример

Тестовый пример 5

На рисунке 17 показан пятый тестовый пример.

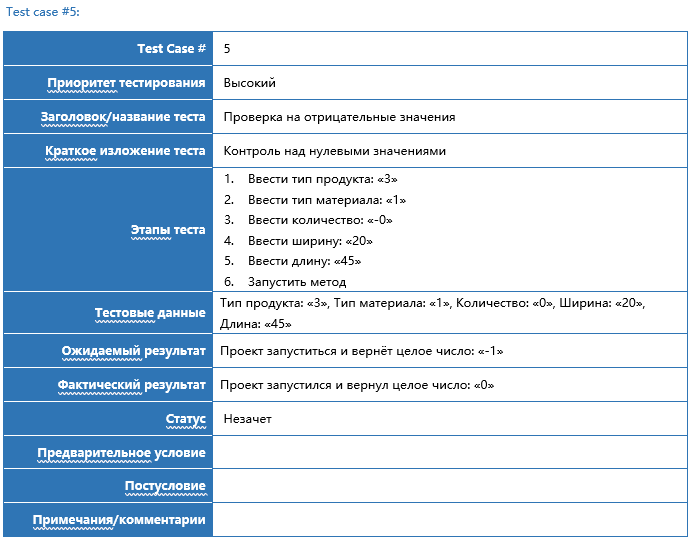


Рисунок 17 – Пятый тестовый пример

# ПРАКТИЧЕСКАЯ РАБОТА №2

## ЗАДАЧА

Согласно своему варианту по МДК 01.01 выполнить интеграционное тестирование:

* 5 unit-tests (связь между модулями)
* 10 test-case
* провести тестирование по test-case

Необходимо описать свой программный продукт.

## Ход работы

**Описание программного продукта:**

Информационная система ООО «Синема» – прикладное программное обеспечение, использующееся для выдачи билетов на киносеансы, просмотра информации по киносеансу, а также предназначена для решения следующих задач:

* контроль документооборота;
* формирование необходимой информации по итогам выдачи билета клиенту (показываемый фильм, дата сеанса, время сеанса, зал и место, дата продажи билета, цена за билет).

Основными целями внедрения системы являются:

* упрощение контроля выдачи билетов;
* полный контроль над статусом киносеанса;
* минимизация человеческого фактора при ведении документов;
* снижение рутиной работы при работе с документами.

Исходная информация в систему поступает из запросов клиента. Входными данными являются следующая информация:

* сведения о фильме;
* сведения о дате киносеанса;
* сведения о времени киносеанса;
* сведения о месте в зале (ряд и номер места).

Рассмотрим определение прецедентов (вариантов использования). Система требуется, прежде всего, следующим заинтересованным лицам:

* оператор кассового отделения;
* гость;
* менеджер;
* системный администратор.

Функционал каждой роли:

Системный администратор обладает возможностью просмотра, редактирования, добавления, удаления данных. Ко всему прочему системный администратор имеет возможность создавать пользователей для системы, удалять их или же изменять их учётные данные.

Гость может только просматривать данные.

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь кроме гостя должен иметь свой пароль доступа в систему.

Проект предназначен для кинотеатра, занимающегося выдачей билетов на киносеансы.

Внедрение данного проекта поможет автоматизировать процесс выдачи билетов, способствуя снижению рутиной работы с ведением документооборота и минимизации случаев человеческого фактора.

**Unit-тесты**

Все пройденные тесты представлены на рисунке 18

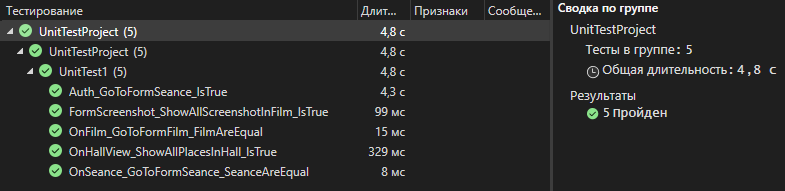


Рисунок 18 – Пройденные тесты

Первый тест ­­­­­­– связь между модулем авторизации и переход к форме списка сеансов. Данный тест представлен на рисунке 19.