Лабораторная работа 2

По дисциплине «Объектно Ориентированное Программирование»

**Настройка и использование библиотеки Junit для модульного тестирования программного обеспечения**

Выполнил:

студент группы ПИ101-БЗУ

ФИО Макуха Т.Б.

Проверил:

Казанцев А.В.

Дата сдачи

Отметка о сдаче

**Цель работы**

Целью работы является изучение библиотеки *JUnit* для тестирования программного кода и создания веток, тэгов и выполнения слияния в *Git*.

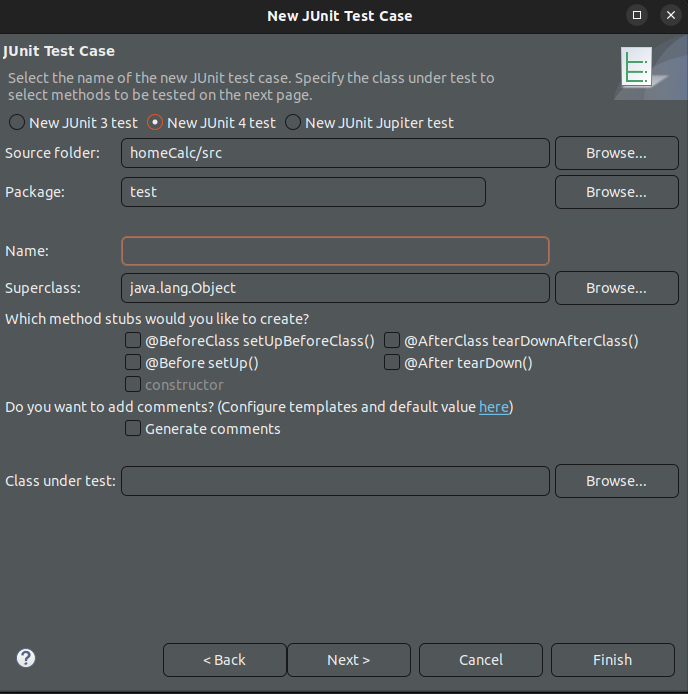
Вариант 12

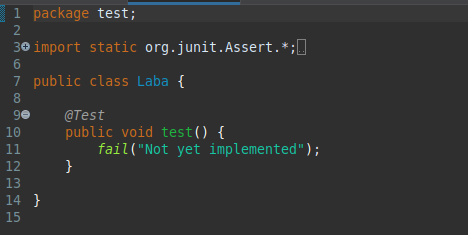
**Ход работы:**

Используется проект maven, для создания unit теста добавляю junit test case через контекстное меню.

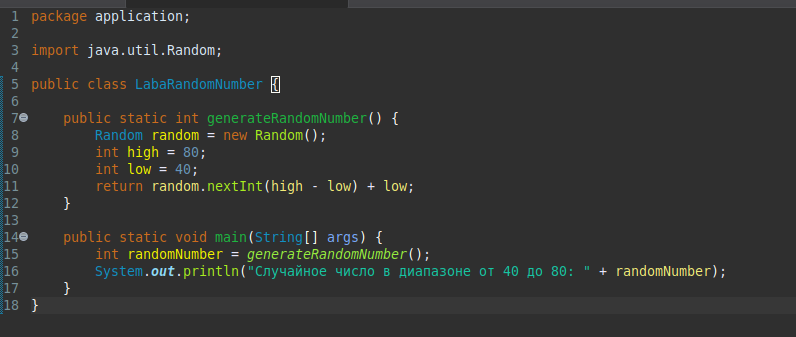
В открывшемся меню ввожу настройки тест кейса

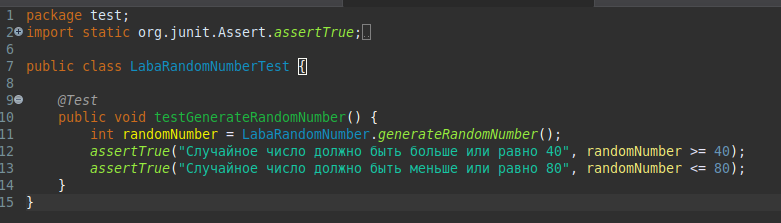
В итоге получаю созданный тест

Рисунок 1: Настройка тест кейса Junit



Задание: создать класс и соответствующий метод: генерирует случайные числа в диапазоне от 40 до 80

Задание 2: проверка, сгенерировалось ли целое число в нужном диапазоне:



**Заключение:**

В ходе выполнения данной лабораторной работы я научился создавать unit тесты с использованием Junit

**Контрольные вопросы:**

1. Что такое модульное тестирование?

Это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Его цель заключается в том, чтобы проверить, что каждая единица программного кода работает должным образом. Данный вид тестирование выполняется разработчиками на этапе кодирования приложения. Модульные тесты изолируют часть кода и проверяют его работоспособность. Единицей для измерения может служить отдельная функция, метод, процедура, модуль или объект.

1. Что такое *JUnit*?

Это фреймворк для проведения автоматического тестирования программ на Java

1. Как располагаются тесты в *JUnit*?

Тесты находятся внутри тестовых классов, которые в свою очередь находятся внутри тест кейса

1. Что такое *assert* и *assume*?

Assert — это конструкция, позволяющая проверять предположение о значениях произвольных данных в произвольном месте программы.

Assume — это конструкция, позволяющая проверять корректность теста перед его выполнением.

1. В чем отличие ветки от метки?

Тег похож на ветвь, которая не меняется. В отличие от веток, теги после создания не имеют дальнейшей истории коммитов.