Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Информационные технологии и программирование

Лабораторная работа №8

Выполнила:

студентка группы БВТ2203

Лысикова К. А.

Москва, 2023 г.

**Цель работы:** изучение многопоточности в Java.

**Задание 1**: Реализация многопоточной программы для вычисления суммы элементов массива.

Вариант 2. Создать пул потоков с помощью класса ExecutorService и разделить массив на равные части, каждую из которых будет обрабатывать отдельный поток. После завершения работы всех потоков результаты будут складываться в главном потоке.

**Задание 2:** Реализация многопоточной программы для поиска наибольшего элемента в матрице.

Вариант 2. Создать пул потоков с помощью класса ExecutorService и разделить матрицу на равные части, каждую из которых будет обрабатывать отдельный поток. После завершения работы всех потоков результаты будут сравниваться в главном потоке для нахождения наибольшего элемента.

**Задание 3:**

У вас есть склад с товарами, которые нужно перенести на другой склад. У каждого товара есть свой вес. На складе работают 3 грузчика. Грузчики могут переносить товары одновременно, но суммарный вес товаров, которые они переносят, не может превышать 150 кг. Как только грузчики соберут 150 кг товаров, они отправятся на другой склад и начнут разгружать товары. Напишите программу на Java, используя многопоточность, которая реализует данную ситуацию.

Вариант 8. Использование CountDownLatch: Используйте CountDownLatch для синхронизации начала и завершения переноса товаров

# **Ход работы**

Пишу программу, которая вычисляет сумму элементов массива, с использованием класса ExecutorService. Создаю пул потоков из 4 потоков. В цикле for переопределяю метод run.



Рисунок 1 – Задание 1 (часть 1)

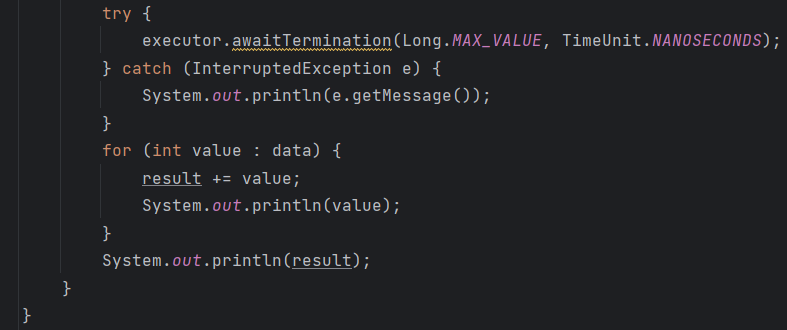


Рисунок 2 – Задание 1 (часть 2)

Пишу программу для поиска наибольшего элемента в матрице. Использую класс ExecutorService.



Рисунок 3 – Задание (часть 1)



Рисунок 4 – Задание 2 (часть 2)

Пишу программу для переноса товаров со склада на другой склад, использую CountDownLatch для синхронизации начала и завершения переноса товаров. Пишу класс Loader, у которого есть 5 полей. startSignal и doneSignal – сигналы начала и конца переноса товаров. Weights – список весов товаров, maxWeightPerTrip – максимальный вес, который может переносить грузчик, workerNum – номер рабочего, чтобы отслеживать перенос

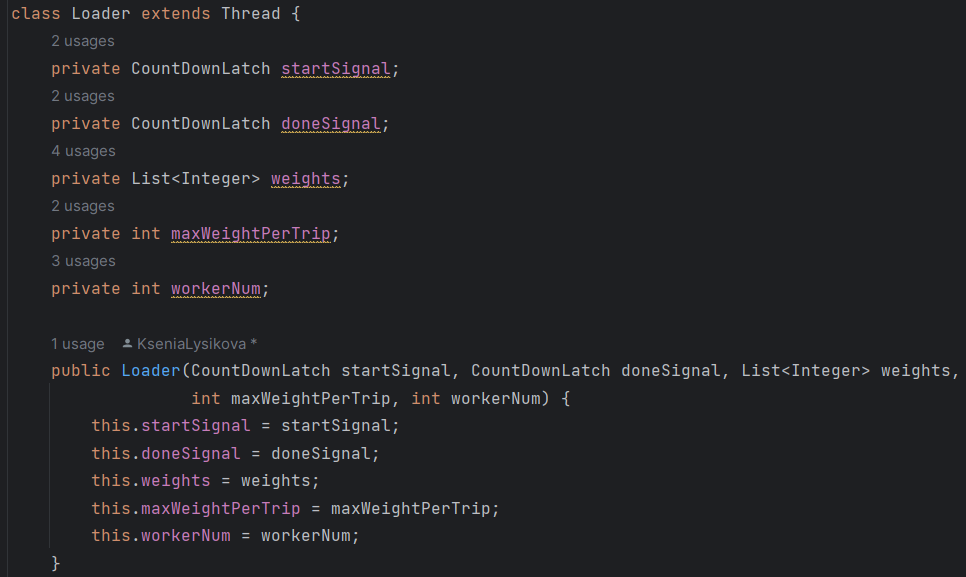


Рисунок 5 – Класс Loader (поля и конструктор)

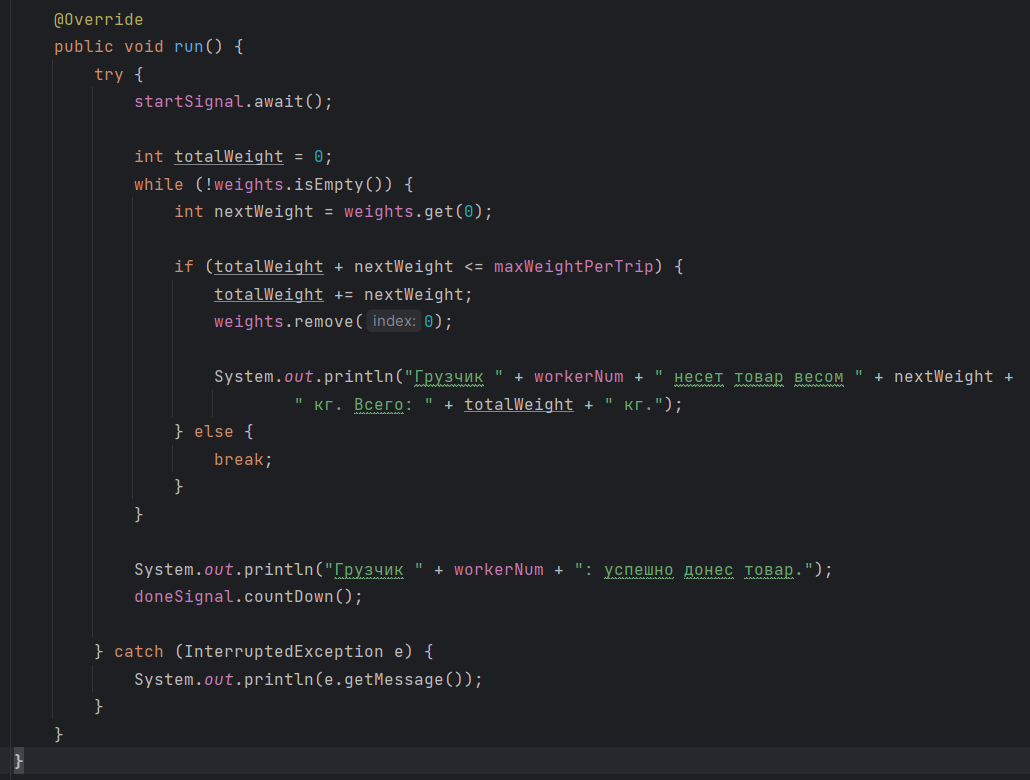


Рисунок 6 – метод run в классе Loader



Рисунок 7 - метод main в классе Loader

# **Вывод**

Изучила потоки в Java. Написала программу для поиска наибольшего элемента в матрице, с использованием потоков и класса ExecutorService. Написала реализацию многопоточной программы для поиска наибольшего элемента в матрице, с использованием ExecutorService. Написала программу для переноса товаров со склада на другой склад, используя CountDownLatch для синхронизации начала и завершения переноса товаров.