**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Информационные технологии»

**Отчет по лабораторной работе №5**

по дисциплине «Информационные технологии и программирование» на тему:

Многопоточность в Java

Выполнил:

студент группы БВТ2302

Боданюк Алексей Павлович

Проверил:

Харрасов К. Р.

Москва 2024

Цель работы: изучить многопоточность в Java и выполнить ряд заданий.

Задания:

Задание 1. Реализация многопоточной программы для вычисления суммы элементов массива. Вариант 1. Создать два потока, которые будут вычислять сумму элементов массива по половинкам, после чего результаты будут складываться в главном потоке.

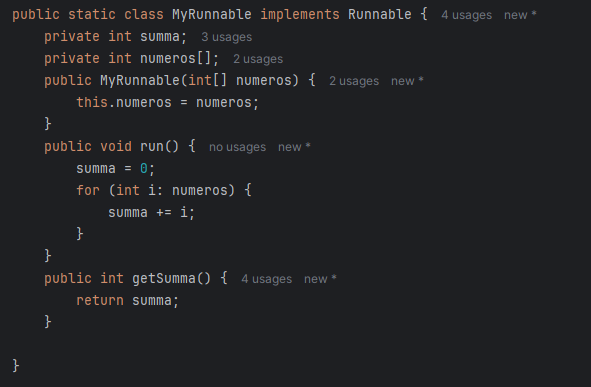
Задание 2. Реализация многопоточной программы для поиска наибольшего элемента в матрице. Вариант 1. Создать несколько потоков, каждый из которых будет обрабатывать свою строку матрицы. После завершения работы всех потоков результаты будут сравниваться в главном потоке для нахождения наибольшего элемента.

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. Использование Thread: Создайте классы Товар, Склад, и Грузчик. Каждый грузчик должен быть представлен в виде отдельного потока

Ход работы:

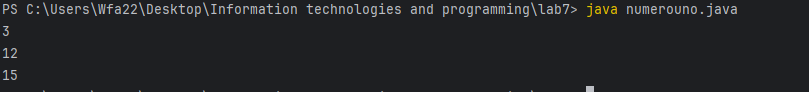
Приступим к выполнению задания 1.

Создадим класс руннер, в который будет передаваться массив чисел, и он будет подсчитывать его сумму:



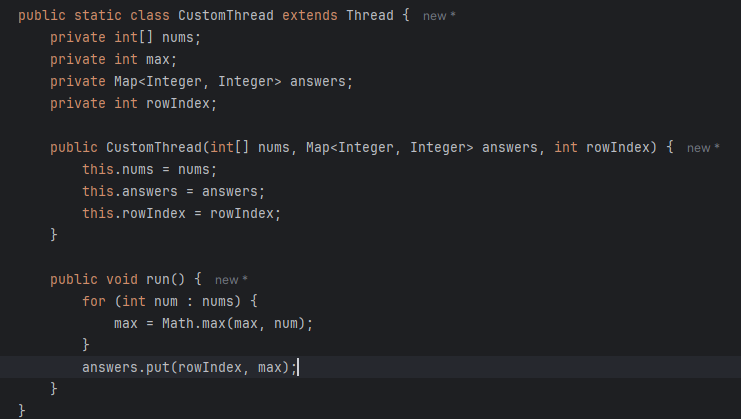
Далее в методе мейн возьмем наш массив, и передадим двум наших потокам его половинки:



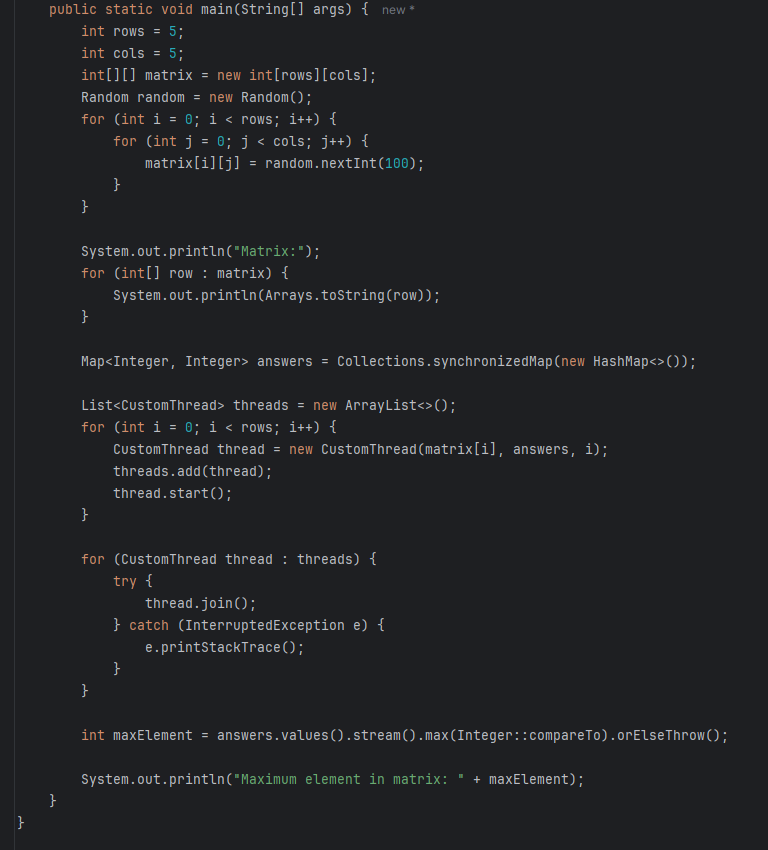


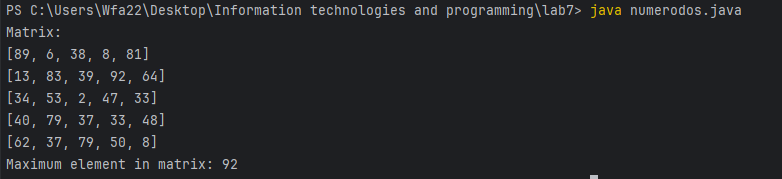
Задание номер 2:

Создадим кастомный класс потока, каждый из которых будет добавлять в общий объект Map добавлять пару значение-индекс строки максимального числа из строчки, которая была ему передана.



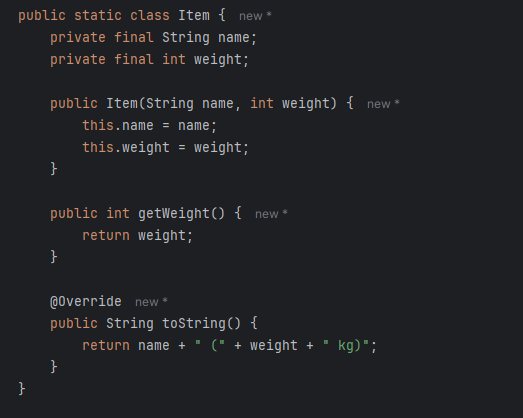
Напишем метод мейн, в котором мы создаем случайную матрицу, передаем её построчно в наши потоки



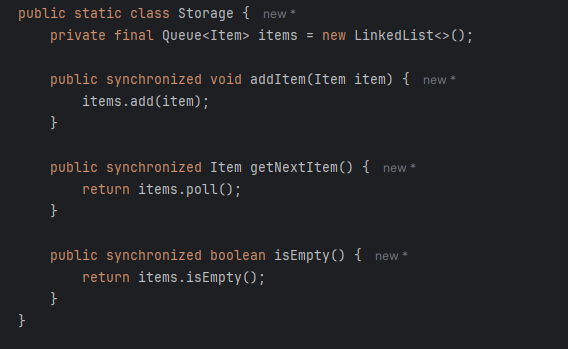


Приступим к заданию номер 3:

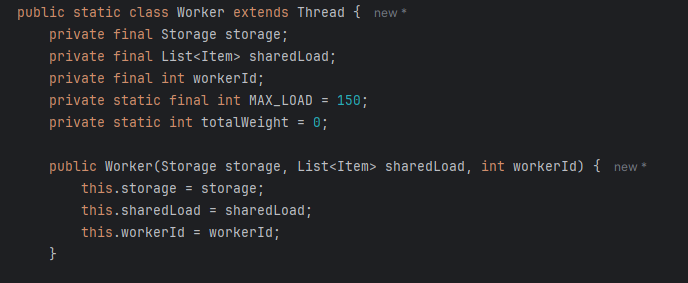
Создадим класс для хранения значений предметов:



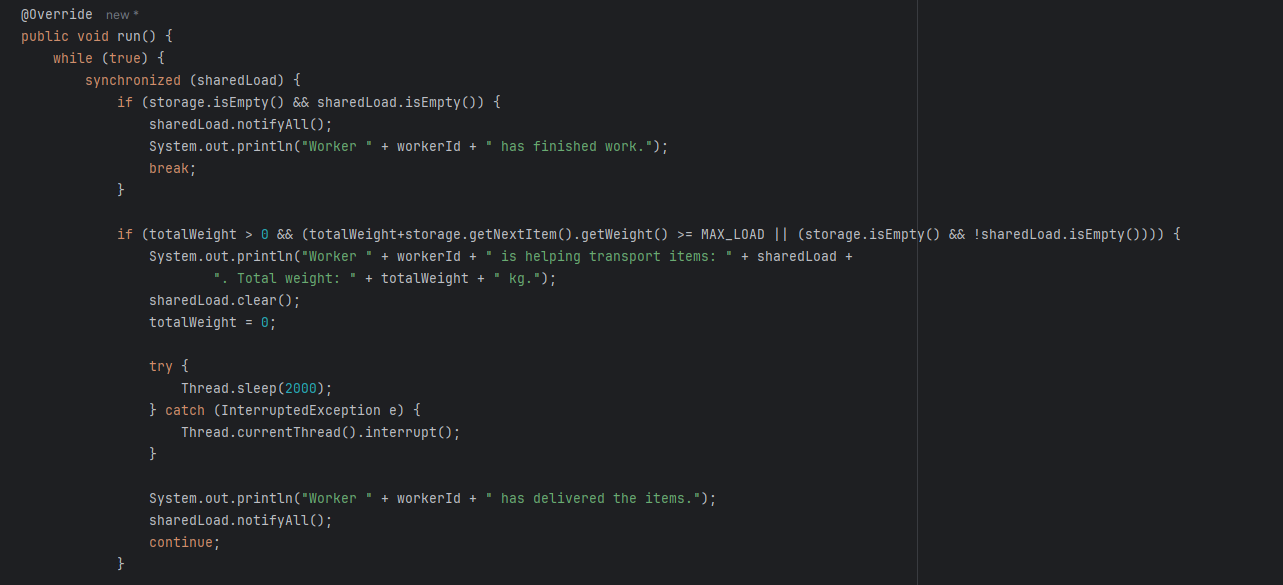
Создадим класс склада, который будет из себя представлять очередь айтемов



Создадим класс воркер и определим для него несколько переменных

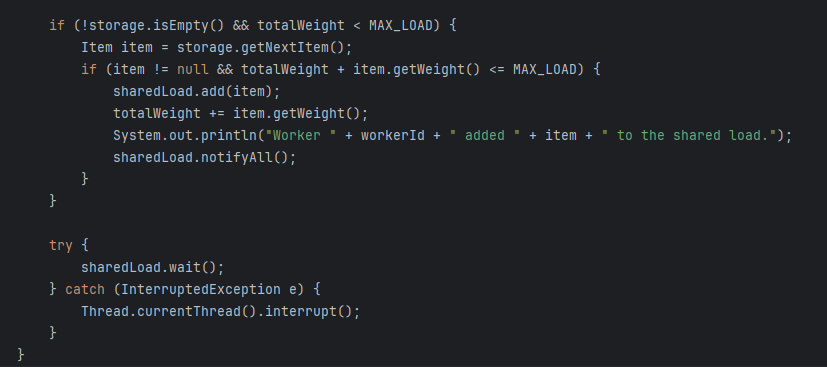


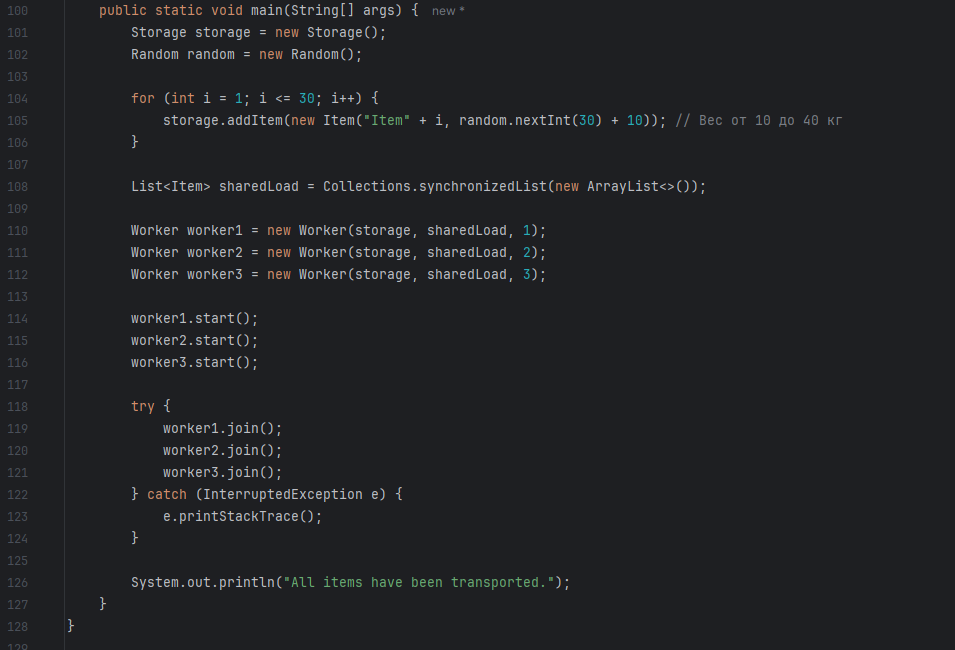
Склад – ссылается на наш склад. Sharedload- общее пространство, куда воркеры со склада сгружают товары. Workerid – уникальное для каждого рабочего значение, чтобы отличать их между собой. MAX\_LOAD – значение 150 кг, порог, который мы не должны превышать для перевоза. Totalweight – значение текущего веса товаров в sharedload.



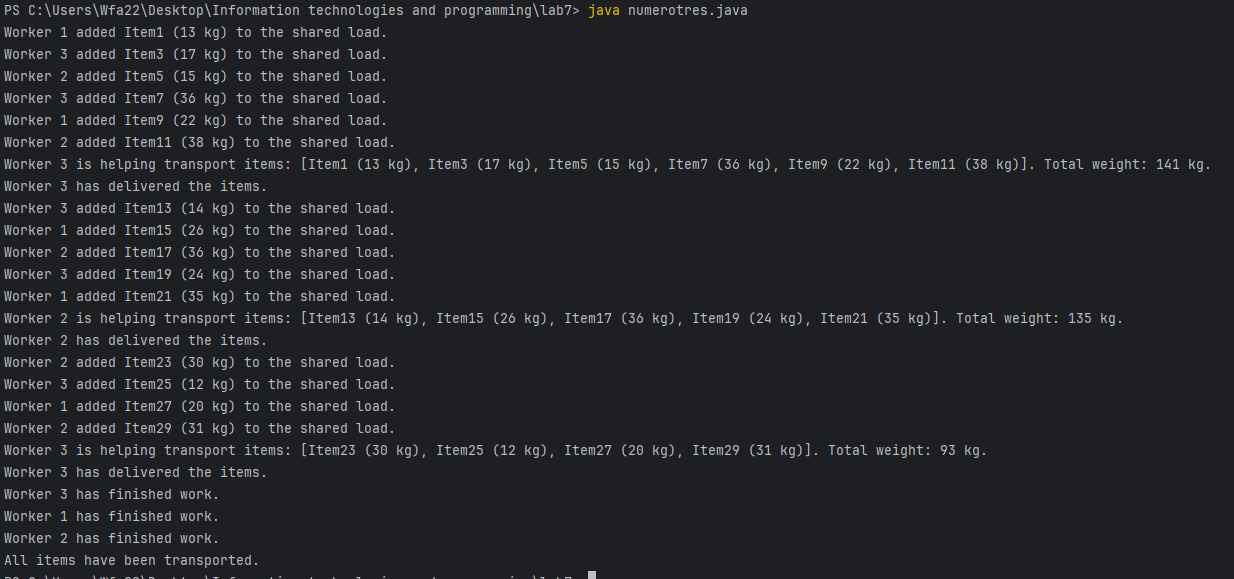
В коды выше мы выполнили проверку на окончание работы по окончанию доступных товаров. Затем есть проверка на то, что больше товаров мы загрузить не можем, и мы начинаем их доставку.

Затем добавили проверку на наличие места в погрузчике и добавление нового товара, а также на то, если никакое из условий выше не выполнилось, мы переводили бы грузчика в режим ожидания





Создали метод мейн и в нем мы добавляем 30 товаров на склад, создаем 3 рабочих и начинаем выгрузку.



Вывод: изучили многопоточность в Джаве, и выполнили задания.