

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А. Н. Тихонова

Департамент компьютерной инженерии

Отчет по практической работе №2

«Работа с приоритетами»

по курсу «Распределенные базы данных и сетевые вычисления»

Выполнили:

Студенты группы МКС212

Журсунова Найля

Шабалина Анастасия Владимировна

Приняла:

Байбикова Татьяна Николаевна

Москва, 2022 г.

Оглавление

1. Задание	2
2. Установка приоритетов	2
2.1. CarThreads	3
2.1.1. Реализация	3
2.1.2. Результат выполнения работы	4
2.2. CarRunnable	5
2.2.1. Реализация	5
2.2.2. Результат выполнения работы	6
3. Выводы	7

1. Задание

Изучить теоретический материал для практики №1. Разобрать прилагаемые примеры. Создать свои примеры, в которых потокам назначаются приоритеты.

Подготовить отчет.

В отчет включить:

- задание;
- разработанную программу или набор программ;
- результаты работы программы;
- краткие выводы.

Работа выполнялась на языке программирования Java.

2. Установка приоритетов

В Java для назначения потокам приоритетов существуют методы `getPriority()` и `setPriority()` в классе `Thread`, а также объявлены три константы:

- `MIN_PRIORITY`
- `MAX_PRIORITY`
- `NORM_PRIORITY`

2.1. CarThreads

Внесем изменения в функцию `main()` в классе `CarThreads` в файле `CarThread.java`. Назначаем потокам приоритеты при помощи метода `setPriority()` и констант (`MIN_PRIORITY`, `MAX_PRIORITY` и `NORM_PRIORITY`).

2.1.1. Реализация

```
public class CarThreads {

    public static void main(String[] args) {
        // Демонстрация создания потоков
        // На вход подается имя бензоколонки и кол-во автомобилей,
        //которые должны заправиться на этой бензоколонке
        PetrolStation t1 = new PetrolStation("Бензоколонка №1", 7);
        PetrolStation t2 = new PetrolStation("Бензоколонка №2",
11);
        PetrolStation t3 = new PetrolStation("Бензоколонка №3",
20);
        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.NORM_PRIORITY);
        t3.setPriority(Thread.MAX_PRIORITY);
        t1.start(); //создаем новый поток
        t2.start(); //создаем новый поток
        t3.start(); //создаем новый поток
    }
}
```

2.1.2. Результат выполнения работы

```
nasty@Zen:~/Загрузки$ java rbd/thread/CarThreads
Бензоколонка №3 открыта...
Бензоколонка №1 открыта...
Бензоколонка №2 открыта...
Заправлено автомобилей на Бензоколонка №3: 1
Заправлено автомобилей на Бензоколонка №1: 1
Заправлено автомобилей на Бензоколонка №2: 1
Заправлено автомобилей на Бензоколонка №1: 2
Заправлено автомобилей на Бензоколонка №3: 2
Заправлено автомобилей на Бензоколонка №2: 2
Заправлено автомобилей на Бензоколонка №1: 3
Заправлено автомобилей на Бензоколонка №3: 3
Заправлено автомобилей на Бензоколонка №2: 3
Заправлено автомобилей на Бензоколонка №1: 4
Заправлено автомобилей на Бензоколонка №3: 4
Заправлено автомобилей на Бензоколонка №2: 4
Заправлено автомобилей на Бензоколонка №1: 5
Заправлено автомобилей на Бензоколонка №3: 5
Заправлено автомобилей на Бензоколонка №2: 5
Заправлено автомобилей на Бензоколонка №1: 6
Заправлено автомобилей на Бензоколонка №3: 6
Заправлено автомобилей на Бензоколонка №2: 6
Заправлено автомобилей на Бензоколонка №1: 7
Заправлено автомобилей на Бензоколонка №3: 7
Заправлено автомобилей на Бензоколонка №2: 7
Заправлено автомобилей на Бензоколонка №3: 8
Заправлено автомобилей на Бензоколонка №2: 8
Поток на Бензоколонка №1 завершен.
Заправлено автомобилей на Бензоколонка №3: 9
Заправлено автомобилей на Бензоколонка №2: 9
Заправлено автомобилей на Бензоколонка №3: 10
Заправлено автомобилей на Бензоколонка №2: 10
Заправлено автомобилей на Бензоколонка №3: 11
Заправлено автомобилей на Бензоколонка №2: 11
Заправлено автомобилей на Бензоколонка №3: 12
Поток на Бензоколонка №2 завершен.
Заправлено автомобилей на Бензоколонка №3: 13
Заправлено автомобилей на Бензоколонка №3: 14
Заправлено автомобилей на Бензоколонка №3: 15
Заправлено автомобилей на Бензоколонка №3: 16
Заправлено автомобилей на Бензоколонка №3: 17
Заправлено автомобилей на Бензоколонка №3: 18
Заправлено автомобилей на Бензоколонка №3: 19
Заправлено автомобилей на Бензоколонка №3: 20
Поток на Бензоколонка №3 завершен.
```

2.2. CarRunnable

Внесем изменения в функцию `main()` в классе `CarRunnable` в файле `CarRunnable.java`. Назначаем потокам приоритеты при помощи метода `setPriority()` и констант (`MIN_PRIORITY`, `MAX_PRIORITY` и `NORM_PRIORITY`).

2.2.1. Реализация

```
public class CarRunnable {
    public static void main(String[] args) {
        System.out.println("Заправка открыта...");
        // Демонстрация работы дочернего потока
        Thread t1 = new Thread(new MyRunnable(12), "Бензоколонка1");
        Thread t2 = new Thread(new MyRunnable(5), "Бензоколонка2");
        Thread t3 = new Thread(new MyRunnable(18), "Бензоколонка3");
        t1.setPriority(Thread.NORM_PRIORITY);
        t2.setPriority(Thread.MAX_PRIORITY);
        t3.setPriority(Thread.MIN_PRIORITY);
        t1.start();
        t2.start();
        t3.start();
        try {
            t1.join();
        }
        catch (InterruptedException e) {
            System.out.println("Поток на бензоколонке1 прерван.");
        }
        try {
            t2.join();
        }
        catch (InterruptedException e) {
            System.out.println("Поток на бензоколонке2 прерван.");
        }
        try {
            t3.join();
        }
        catch (InterruptedException e) {
            System.out.println("Поток на бензоколонке3 прерван.");
        }
        System.out.println("Заправка закрыта.");
    }
}
```

2.2.2. Результат выполнения работы

```
nasty@Zen:~/Загрузки$ java rbd/thread/CarRunnable
Заправка открыта...
Бензоколонка открыта...
Бензоколонка открыта...
Бензоколонка открыта...
Заправлено автомобилей на Бензоколонка3: 1
Заправлено автомобилей на Бензоколонка2: 1
Заправлено автомобилей на Бензоколонка1: 1
Заправлено автомобилей на Бензоколонка3: 2
Заправлено автомобилей на Бензоколонка2: 2
Заправлено автомобилей на Бензоколонка1: 2
Заправлено автомобилей на Бензоколонка3: 3
Заправлено автомобилей на Бензоколонка1: 3
Заправлено автомобилей на Бензоколонка2: 3
Заправлено автомобилей на Бензоколонка3: 4
Заправлено автомобилей на Бензоколонка2: 4
Заправлено автомобилей на Бензоколонка1: 4
Заправлено автомобилей на Бензоколонка3: 5
Заправлено автомобилей на Бензоколонка2: 5
Заправлено автомобилей на Бензоколонка1: 5
Заправлено автомобилей на Бензоколонка3: 6
Заправлено автомобилей на Бензоколонка1: 6
Бензоколонка2 закрыта.
Заправлено автомобилей на Бензоколонка3: 7
Заправлено автомобилей на Бензоколонка1: 7
Заправлено автомобилей на Бензоколонка3: 8
Заправлено автомобилей на Бензоколонка1: 8
Заправлено автомобилей на Бензоколонка3: 9
Заправлено автомобилей на Бензоколонка1: 9
Заправлено автомобилей на Бензоколонка3: 10
Заправлено автомобилей на Бензоколонка1: 10
Заправлено автомобилей на Бензоколонка3: 11
Заправлено автомобилей на Бензоколонка1: 11
Заправлено автомобилей на Бензоколонка3: 12
Заправлено автомобилей на Бензоколонка1: 12
Заправлено автомобилей на Бензоколонка3: 13
Бензоколонка1 закрыта.
Заправлено автомобилей на Бензоколонка3: 14
Заправлено автомобилей на Бензоколонка3: 15
Заправлено автомобилей на Бензоколонка3: 16
Заправлено автомобилей на Бензоколонка3: 17
Заправлено автомобилей на Бензоколонка3: 18
Бензоколонка3 закрыта.
Заправка закрыта.
```

3. Выводы

В результате выполнения практической работы №2 был получен навык работы с языком программирования Java. Были изучены методы класса `Thread` (например, `setPriority()` и `getPriority()`) и применены на практике. С их помощью была добавлена приоритизация потоков.

Кроме этого, нами были закреплены знания, полученные в ходе выполнения предыдущей работы и получены новые знания относительно назначения приоритетов.