

Разработка многопоточных приложений на языке Java

Источники

1. Хорстманн, Кей С. Java. Библиотека профессионала, том 1. Основы. 11-е изд.: Пер. с англ. СПб.: ООО "Диалектика", 2019.
2. Блинов, И. Н., Романчик, В. С. Java from EPAM : учеб.-метод. пособие / И. Н. Блинов, В. С. Романчик. —2-е издание. Минск: Четыре четверти, 2021. —560 с.
3. <https://math.hws.edu/javanotes/>

Правила оформления отчета

- Единый файл.
- Имя файла должно содержать фамилию автора отчета и номер лабораторной работы.

Задание 1.

Изучить (вспомнить) способы создания и запуска потока в Java.

[2, 369-370]

[1,12.1, 662-667]

Задание 2.

2.1.

Скомпилировать и запустить программу:

- class ThreadTest1

[3]

<https://math.hws.edu/javanotes/c12/s1.html>

Чтобы помочь вам понять, как несколько потоков выполняются параллельно, мы рассмотрим пример программы ThreadTest1.java. Эта программа создает несколько потоков. Каждый поток выполняет одну и ту же задачу. Задача — посчитать количество простых чисел меньше 10 000 000. (Конкретная выполняемая задача не важна для наших целей, поскольку она занимает нетривиальное количество времени. Это демонстрационная программа; в реальной программе было бы глупо иметь несколько потоков, которые сделайте то же самое, и метод, используемый для подсчета простых чисел, очень неэффективен.)

Главный поток не ожидает завершения дочерних потоков.

Знать:

- как замерять время выполнения вычислений

```
long startTime = System.currentTimeMillis();
int count = countPrimes(2, MAX);
long elapsedTime = System.currentTimeMillis() - startTime;
System.out.println("Thread " + id + " counted " +
    count + " primes in " + (elapsedTime/1000.0) + " seconds.");
```

2.2.

Жизненный цикл потока [2, 371-372].

2.3.

[3]

<https://math.hws.edu/javanotes/c12/s1.html>

Добавить, чтобы главный поток ожидал завершения дочерних потоков.

Знать:

- Определение состояния потока.
- Ожидание завершения потока.

2.4.

1. Последовательная программа.

Реализовать решение задачи «Посчитать количество простых чисел меньше 10 000 000» в одном потоке.

2. Многопоточная (параллельная) программа

class ThreadTest1 – разделить работу между потоками.

Проведение вычислительных экспериментов

Для оценки эффективности параллельного способа решения задачи проводят вычислительные эксперименты и измеряют время выполнения последовательной и параллельной версий программ для различных входных данных.

Одной из главных характеристик параллельной программы является *ускорение*:

$$S_p(n) = T_1(n) / T_p(n),$$

где:

$T_1(n)$ – время выполнения последовательной версии программы,

$T_p(n)$ – время выполнения параллельной программы на p -процессорной системе.

Еще одна важная характеристика параллельной программы – *эффективность*:

$$E_p(n) = T_1(n) / (p * T_p(n)) = S_p(n) / p.$$

При проведении вычислительных экспериментов измеряют время и заполняют таблицу.

Результаты вычислительных экспериментов

Размерность задачи	Время выполнения последовательной программы	Параллельная программа на 2 процессорах			... Параллельная программа на p процессорах		
		Время выполнения	Ускорение	Эффективность	Время выполнения	Ускорение	Эффективность

Задание 3.

Проведение вычислительных экспериментов:

- Измерить время работы Последовательной и Многопоточной программ.

- Провести эксперименты с многопоточной реализацией с разными размерностями задачи и количеством потоков.
- Заполнить таблицу.

Отчет:

- Текст многопоточной программы.
- Таблица с результатами экспериментов.