

ГУАП
КАФЕДРА № 51

ПРЕПОДАВАТЕЛЬ

доцент, к. т. н.		Е. М. Линский
должность , уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8
СОЗДАНИЕ ПРОГРАММЫ НА ЯЗЫКЕ JAVA

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	5721		А.Е.Ковалева
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2019

Задание 8. Thread.

1. Основное задание

Реализовать класс `ParallelMatrixProduct` для многопоточного умножения матриц `UsualMatrix`. В конструкторе класс получает число потоков, которые будут использованы для перемножения (число потоков может быть меньше, чем число строк у первой матрицы).

В функции `main` сравнить время перемножения больших случайных матриц обычным и многопоточным способом. Получить текущее время можно с помощью методов класса `System`.

2. Дополнительное задание

Сделать многопоточную реализацию `mergesort`, для сортировки любых наследников `Object`.

`Void mergesort(Object array[], int threadNum)`

Подберите размер массива, на котором `mergesort(XXX, какое-то число>1)` выигрывает по времени работы у `mergesort(XXX, 1)`.

Инструкция

1. Основное задание.

После запуска программы на экран выводится время умножения матриц при многопоточном и однопоточном умножении.

Распределение матрицы на потоки происходит в 3 этапа:

1. Передача в метод количества потоков
2. Построчное разбиение матрицы на количество потоков
3. Запускается цикл на количество итераций, которые определяются числом потоков, переданных в функцию.

В тестах была рассмотрена матрица из 1000×1000 , число потоков 500.

После запуска программы на экран выводится время реализации многопоточного умножения матриц и время умножения при одном потоке.

2. Дополнительное задание.

Для сортировки наследников `Object` я в метод передаю массив наследников и количество потоков. Разбиваю массив на 2 части: от начала до середины и от следующего за серединой элементом до конца. Затем потоки сортируют

неотсортированные элементы по возрастанию. Разбиение происходит N раз, равное числу потоков.

В тестах был рассмотрен массив из 10000 элементов и число потоков, равное 20.

После запуска программы выводится время, за которое были отсортированы наследники Object при одном потоке.

Затем выводится время сортировки при нескольких потоках.

Тестирование

1. Выводятся на экран время многопоточного и однопоточного умножения матриц.

```
main x
"E:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\IntelliJ IDEA Commu
Скорость при многопоточном (500 потоков) = 5137
Скорость при одном потоке = 11611

Process finished with exit code 0
```

```
main x
"E:\Program Files\Java\jdk-11.0.2\bin\java.exe" "
Скорость при многопоточном (5 потоков) = 5058
Скорость при одном потоке = 5686

Process finished with exit code 0
```

```
main x
"E:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:
Скорость при многопоточном (1000 потоков) = 4726
Скорость при одном потоке = 8094

Process finished with exit code 0
```

2. Вывод времени реализации многопоточной сортировки и однопоточной.

```
Main x
"E:\Program Files\Java\jdk-11.0.2\bin\java.
Один поток сортирует за время: 180ms
5 потоков сортируют за время: 135ms

Process finished with exit code 0
```

```
M x
"E:\Program Files\Java\jdk-11.0.2\bin\java.exe
Один поток сортирует за время: 195ms
50 потоков сортируют за время: 165ms

Process finished with exit code 0
```

```
M x
"E:\Program Files\Java\jdk-11.0.2\bin\java.exe" "-javaagent:C:\I
Один поток сортирует за время: 309ms
20 потоков сортируют за время: 129ms

Process finished with exit code 0
```