

ГУАП
КАФЕДРА № 52

ПРЕПОДАВАТЕЛЬ

доцент, к.т.н.		Линский Е. М.
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. №	5022		К.А. Агаширинов
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2022

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`.

Дополнительное задание

Реализуйте многопоточную функцию для подсчета количества нечетных чисел в `ArrayList`.

```
int calcOddNum(ArrayList<int> array, int theardNum);
```

Функция в качестве параметра получает число потоков.

Инструкция

Класс `ParallelMatrixProduct` содержит в себе метод `public UsualMatrix multiThreading/Multiply(UsualMatrix firstMatrix, UsualMatrix secondMatrix)`, который возвращает произведение матриц `firstMatrix` на `secondMatrix`. Данный метод позволяет пользователю воспользоваться многопоточным умножением матриц.

Тестирование

1. Основное задание:

`firstMatrix` на `secondMatrix` заполняются в `main` случайным образом с помощью класса `Random()`. Помимо сравнения скорости работы многопоточного и обычного умножения, в `main` также с помощью метода `equals` проверяется, что результат многопоточного умножения равен результату обычного умножения. Время выполнения выводится на экран в миллисекундах.

Тест 1:

```
matrixSize = 1500  
amountOfThreads = 12  
Result:
```

```
Time for multi threading multiplication of matrix: 6211  
Time for default multiplication of matrix: 13934  
Result of multithreading and default multiplication is equals: true
```

Тест 2:

matrixSize = 1500
amountOfThreads = 4

Result:

```
Time for multi threading multiplication of matrix: 6796  
Time for default multiplication of matrix: 15472  
Result of multithreading and default multiplication is equals: true
```

Тест 3:

matrixSize = 1500
amountOfThreads = 1

Result:

```
Time for multi threading multiplication of matrix: 15199  
Time for default multiplication of matrix: 14434  
Result of multithreading and default multiplication is equals: true
```

2. Дополнительное задание:

Массив чисел заполняется случайным образом с помощью класса Random(). Помимо сравнения скорости работы многопоточного и обычного алгоритма нахождения количества нечетных чисел, в main также проверяются, что результат многопоточного алгоритма равен результату обычного алгоритма. Время выполнения выводится на экран в миллисекундах.

Тест 1:

matrixSize = 150000000
amountOfThreads = 12

Result:

```
Amount off odd numbers(default algorithm): 83323485  
time for default calculate amount off odd numbers: 3186  
Amount off odd numbers(multithreading algorithm): 83323485  
time for multithreading calculate amount off odd numbers: 1609
```

Тест 2:

matrixSize = 150000000
amountOfThreads = 4

Result:

```
Amount off odd numbers(default algorithm): 83335328  
time for default calculate amount off odd numbers: 3094  
Amount off odd numbers(multithreading algorithm): 83335328  
time for multithreading calculate amount off odd numbers: 2154
```

Тест 3:

matrixSize = 150000000

amountOfThreads = 1

Result:

```
Amount off odd numbers(default algorithm): 83317074
```

```
time for default calculate amount off odd numbers: 3233
```

```
Amount off odd numbers(multithreading algorithm): 83317074
```

```
time for multithreading calculate amount off odd numbers: 3176
```