

ГУАП
КАФЕДРА №51

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

THREAD

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ II

РАБОТУ ВЫПОЛНИЛ
СТУДЕНТ ГР. № 5511

подпись, дата

инициалы, фамилия

Санкт-Петербург, 2017

1 Задание

Реализовать класс `ThreadedMatrixProduct` 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

Реализовать многопоточное переборное решение задачи о рюкзаке.

3 Реализация

3.1 Основное задание

Для многопоточного перемножения матриц созданы отдельные классы `ThreadedMatrixProduct` и `MultiplicationThread`.

Класс `MultiplicationThread` наследован от класса `Thread`, хранит значения результирующей и перемножаемых матриц, номера строк в диапазоне которых будет происходить умножение. В переопределенном методе `run()` происходит перемножение выделенной части матрицы.

У класса `ThreadedMatrixProduct` есть метод `threadedProduct`, который получает в качестве параметров первую и вторую матрицы, а так же количество потоков, с помощью которых будет перемножаться матрица. Данный метод рассчитывает количество элементов матрицы, которые должен рассчитать каждый поток. Затем в цикле создаются `MultiplicationThread`, которым в конструкторе передаются необходимые параметры, а именно матрицы для перемножения, результирующая матрица, а так же начальный и конечные индексы в матрице для вычисления.

3.2 Дополнительное задание

Для многопоточного решения задачи о рюкзаке методом перебора были созданы классы `Knapsack` и `Item`. Класс `Item`, хранит информацию о предмете, а именно имя, стоимость и вес.

У класса `Knapsack` существует два главных публичных метода:

- `findBestCombination(ArrayList<Item> items)`, ищущий лучшую комбинацию из доступного списка предметов, рекурсивным методом.
- `threadedFindBestCombination(ArrayList<Item> items, int threadCount)`, который ищет лучшую комбинацию, разбивая доступный список по потокам и вызывая не многопоточный поиск лучшей комбинации для каждого потока.

4 Инструкция

При запуске программы основного задания она последовательно выводит на экран:

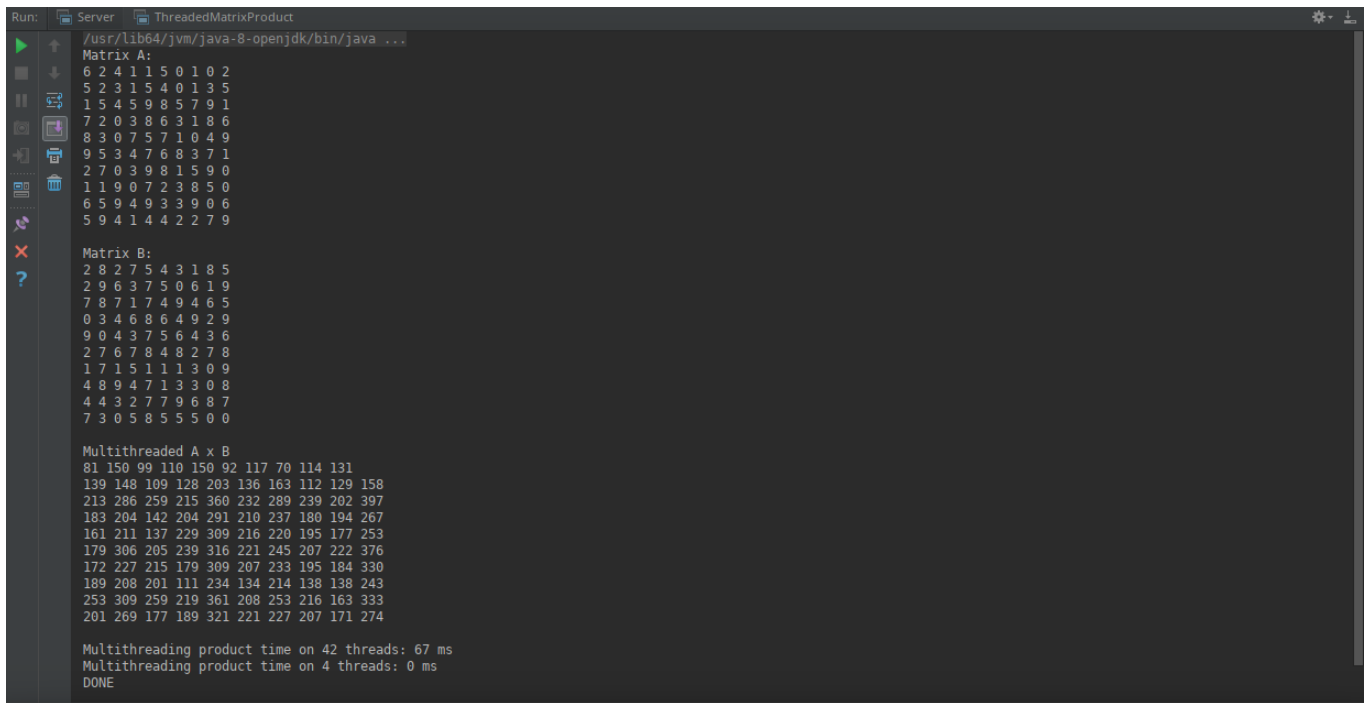
- Время и результат перемножения матриц на 42 потоках.
- Время и результат перемножения матриц на 4 потоках.

При запуске программы дополнительного задания она последовательно выводит на экран:

- Время и результат решения задачи о рюкзаке с заданными параметрами рекурсивным методом.
- Время и результат решения задачи о рюкзаке с заданными параметрами многопоточным методом.

5 Тестирование

5.1 Пример запуска программы



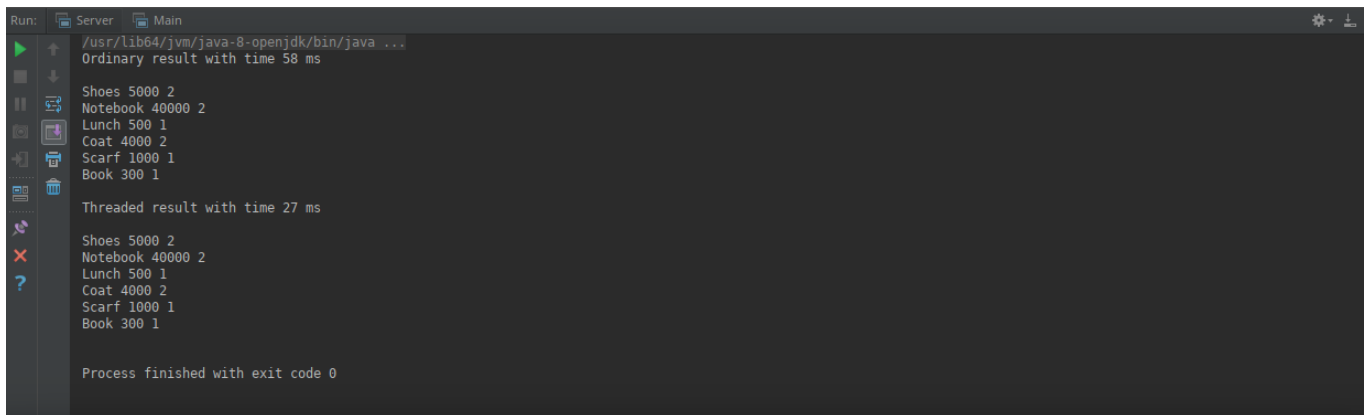
```
Run: Server ThreadedMatrixProduct
/usr/lib64/jvm/java-8-openjdk/bin/java ...
Matrix A:
6 2 4 1 1 5 0 1 0 2
5 2 3 1 5 4 0 1 3 5
1 5 4 5 9 8 5 7 9 1
7 2 0 3 8 6 3 1 8 6
8 3 0 7 5 7 1 0 4 9
9 5 3 4 7 6 8 3 7 1
2 7 0 3 9 8 1 5 9 0
1 1 9 0 7 2 3 8 5 0
6 5 9 4 9 3 3 9 0 6
5 9 4 1 4 4 2 2 7 9

Matrix B:
2 8 2 7 5 4 3 1 8 5
2 9 6 3 7 5 0 6 1 9
7 8 7 1 7 4 9 4 6 5
0 3 4 6 8 6 4 9 2 9
9 0 4 3 7 5 6 4 3 6
2 7 6 7 8 4 8 2 7 8
1 7 1 5 1 1 1 3 0 9
4 8 9 4 7 1 3 3 0 8
4 4 3 2 7 7 9 6 8 7
7 3 0 5 8 5 5 5 0 0

Multithreaded A x B
81 150 99 110 150 92 117 70 114 131
139 148 109 128 203 136 163 112 129 158
213 286 259 215 360 232 289 239 202 397
183 204 142 204 291 210 237 180 194 267
161 211 137 229 309 216 220 195 177 253
179 306 205 239 316 221 245 207 222 376
172 227 215 179 309 207 233 195 184 330
189 208 201 111 234 134 214 138 138 243
253 309 259 219 361 208 253 216 163 333
201 269 177 189 321 221 227 207 171 274

Multithreading product time on 42 threads: 67 ms
Multithreading product time on 4 threads: 0 ms
DONE
```

Рис. 1: Пример работы программы основного задания



```
Run: Server Main
/usr/lib64/jvm/java-8-openjdk/bin/java ...
Ordinary result with time 58 ms

Shoes 5000 2
Notebook 40000 2
Lunch 500 1
Coat 4000 2
Scarf 1000 1
Book 300 1

Threaded result with time 27 ms

Shoes 5000 2
Notebook 40000 2
Lunch 500 1
Coat 4000 2
Scarf 1000 1
Book 300 1

Process finished with exit code 0
```

Рис. 2: Пример работы программы дополнительного задания