

Zaawansowane programowanie obiektowe

Lab. 9

(wątki)

1. (1.5 pkt)

Napisz aplikację wielowątkową wypisującą na konsoli napisy z tablicy w następujący sposób:

- a) po znaku, z zachowaniem kolejności wątków (każdy wątek wypisuje po 1 znaku, po czym do działania przechodzi następny wątek i ta kolejność wątków jest zachowywana; chyba że dany wątek zakończy swoją pracę, wtedy oczywiście jest pomijany),
- b) po znaku, ale w dowolnej kolejności (brak synchronizacji wątków). Liczba wątków wypisujących ma być równa rozmiarowi tablicy napisów (proszę nie zakładać jej z góry).

Przykład:

```
private String[] strings = {"aaaa", "bb", "cccccccccccc", "dddddd"};
```

W wersji a) zadania wynikiem ma być:

```
abcd abcd acd acd cd cd c c c c c c
```

(spacje dodane tylko dla przejrzystości).

2. (1 pkt)

Napisz program wielowątkowy znajdujące wszystkie liczby pierwsze z przedziału [s; e] tym razem w aplikacji lokalnej (nie: klient-serwer), nie rozdzielając pracy między wątki „ręcznie”, ale wykorzystując strategię Fork-Join. Uwaga: tym razem nie wystarcza nam ilość liczb pierwszych w zadanym przedziale, ale chcemy otrzymać kolekcję zawierającą wszystkie te liczby. (Pomyśl o odpowiedniej kolekcji listowej, która jest bezpieczna wątkowo (thread-safe).)

Dodatkowo pomierz czas obliczeń i wypisz na ekranie 5 największych par liczb pierwszych bliźniaczych wśród liczb znalezionych (np. 5 i 7 są liczbami pierwszymi bliźniaczymi, bo różnią się o 2).

Materiały:

http://szgrabowski.kis.p.lodz.pl/zpo17/Java_2017_05.ppt, S90–95

http://homes.cs.washington.edu/~djg/teachingMaterials/grossmanSPAC_forkJoinFramework.html

1

<http://docs.oracle.com/javase/tutorial/essential/concurrency/forkjoin.html>