

# Zaawansowane programowanie obiektowe

## Lab. 9 (Wątki)

### 1. (0.5 pkt, ROZGRZEWKA)

Napisz program tworzący 10 wątków, każdy tylko wypisujący: Hello from task x, gdzie x będzie jego numerem (0..9), ale jako pierwszy ma wypisać swój komunikat wątek ostatni (tj. nr 9), potem przedostatni, itd. do pierwszego. Pełne wyjście:

```
Hello from task 9
Hello from task 8
Hello from task 7
Hello from task 6
Hello from task 5
Hello from task 4
Hello from task 3
Hello from task 2
Hello from task 1
Hello from task 0
```

Pomyśl jak to zrobić najprościej.

### 2. (2 pkt, SITO ERATOSTENESA)

Sito Eratostenesa to klasyczny algorytm znajdowania liczb pierwszych, od początku do danego n. Zapoznaj się z tym prostym algorytmem, obejrzyj animację:

[https://pl.wikipedia.org/wiki/Sito\\_Eratostenesa](https://pl.wikipedia.org/wiki/Sito_Eratostenesa)

Napisz program wielowątkowy znajdujące wszystkie liczby pierwsze do 10 mln oraz zliczający ile jest liczb pierwszych od 2 do:

- 1 mln,
- 10 mln,
- 100 mln,
- 1 miliard,

bazujący na sicie E.

Prawidłowe odpowiedzi są w tab. 1 pod

<https://primes.utm.edu/howmany.html>

(funkcja  $\pi(x)$ ).

Rozwiąż ten problem na trzy sposoby. W każdym wariantcie początek i koniec jest szeregowy i realizujemy go w ten sam sposób:

- POCZĄTEK: znajdujemy (też sitem Eratostenesa) i zapamiętujemy na liście smallPrimes liczby pierwsze od 2 do  $\sqrt{n}$ , gdzie n to największy nasz zakres, tj. 1 mld,
- KONIEC: mając ukończoną procedurę sita do 1 miliarda, zliczamy liczby pierwsze do 1 mln, 10 mln, 100 mln, 1 mld oraz wrzucamy liczby pierwsze do 10 mln na listę.

Teraz trzy warianty równoległości, które należy zaimplementować i porównać ich czasy wykonania, osobno dla 2 i 4 wątków. Jeden z wariantów (np. c) puść również dla jednego wątku, żeby mieć czasy „bazowe”.

a)

Każdy wątek bierze pierwszą wolną liczbę pierwszą z listy smallPrimes i przesiewa nią w tablicy byte[] o rozmiarze 1 mld.

b)

Żaden wątek nie zacznie przesiewać z nowym krokiem (tj. z użyciem nowej liczby ze smallPrimes), zanim nie zakończy się krok poprzedni. Załóżmy że mamy  $t$  wątków. Dla NIEPARZYSTEGO kroku  $s$  pierwszy wątek „wykreśla” liczby:

$s^2$ ,  $s^2 + 2ts$ ,  $s^2 + 4ts$ ,  $s^2 + 6ts$ , ...

Drugi wątek wykreśla:

$s^2 + 2s$ ,  $s^2 + 2ts + 2s$ ,  $s^2 + 4ts + 2s$ ,  $s^2 + 6ts + 2s$ , ...

Trzeci wątek wykreśla:

$s^2 + 4s$ ,  $s^2 + 2ts + 4s$ ,  $s^2 + 4ts + 4s$ ,  $s^2 + 6ts + 4s$ , ...

Itd. następne. (Gdzie  $s^2$  to  $s$  kwadrat oczywiście.)

Wątki zatem sobie „nie wchodzą w drogę” – dla konkretnego  $s$  żadna para wątków nie wykreśli tej samej liczby.

Osobnym przypadkiem jest krok parzysty, czyli  $s = 2$  (innej liczby pierwszej parzystej oczywiście nie ma). Tym razem we wzorach nie będzie mnożnika 2, czyli:

$s^2$ ,  $s^2 + ts$ ,  $s^2 + 2ts$ ,  $s^2 + 3ts$ , ...

$s^2 + 2s$ ,  $s^2 + ts + s$ ,  $s^2 + 2ts + s$ ,  $s^2 + 3ts + s$ , ...

$s^2 + 4s$ ,  $s^2 + ts + 2s$ ,  $s^2 + 2ts + 4s$ ,  $s^2 + 3ts + 4s$ , ...

Itd.

WYJAŚNIENIE: jeśli wykreślamy np. wielokrotności liczby 5, czyli  $s = 5$ , to nie trzeba na początku wykreślić  $2 * 5$  i  $3 * 5$ , bo te liczby są już wykreślone przy, odpowiednio,  $s = 2$  i  $s = 3$ . Zatem zaczynamy od  $s^2$  (=25). Ale potem: nie trzeba wykreślać parzystych wielokrotności liczby 5, czyli np. 30 i 40. Wystarczy iść z krokiem co 10, czyli: 35, 45, 55, ... Taka realizacja odpowiada wersji szeregowej kodu. Dla wątków rozrzucamy te wartości, czyli przy 4 wątkach: pierwszy wykreśla: 25, 65, 105, ..., drugi: 35, 75, 115, ..., trzeci: 45, 85, 125, ..., czwarty: 55, 95, 135, ...

Trick z podwajaniem nie działa tylko dla  $s = 2$ . Dlatego ten przypadek (jedyna parzysta liczba pierwsza) musi być obsługiwany nieco inaczej, osobno.

c)

W tym wariantcie nasza duża tablica (o rozmiarze 1 mld bajtów) jest dzielona na  $t$  rozłącznych części dla  $t$  wątków i każdy z nich wykreśla wielokrotności liczb: 2, 3, 5, 7... (wziętych z listy smallPrimes) niezależnie, tylko w swojej części. Czyli przy 4 wątkach: pierwszy zajmuje się pierwszą ćwiartką miliarda, drugi – drugą ćwiartką miliarda itd. (w tej samej dużej tablicy).

3. (1 pkt)

$N = 100$  osób (osoba = wątek) rozważa pójście do kina na konkretny seans. Wątki te startują jednocześnie i każdy wątek po losowej liczbie sekund (od 1 do 4s, tylko całkowita liczba sekund) namyślnie podejmuje decyzję z prawdopodobieństwem „pozytywnym” (tj. wyjścia do kina)  $p1 = 0,05$ . Kino nie wyświetla filmu, jeśli nie zbierze się minimum 5 widzów. W takim

przypadku trzeba wyświetlić napis typu: „Przepraszamy, filmu nie będzie.” i program się kończy.

W przeciwnym wypadku wyświetla się film, którego symulowany czas trwa 4s. Ale film jest nudny i dokładnie w połowie seansu każdy z widzów podejmuje decyzję: wyjść (z prawdop.  $p_2 = 0,3$ ) lub zostać. Jeśli liczba osób, które została nadal wynosi  $\geq 5$ , to film jest kontynuowany. Jeśli nie, to film zostaje przerywany, a kierownictwo kina wypisuje obraźliwy komunikat (o frajerach i o tym, że pieniędzy za bilety nie oddają...).

**Wskazówki:** napisz klasy Osoba oraz Widz. Wąteków-osób będzie 100, wątków-widzów dużo mniej (zwykle około  $100 * p_1 = 5$ ). Skorzystaj z egzekutorów.