

# Zaawansowane programowanie obiektowe

## Lab. 8

### (Guava, programowanie sieciowe)

#### 1. (BISEKCJA, LISTA LENIWA) (1.5 pkt)

Algorytm bisekcji znajduje np. miejsce zerowe funkcji w danym przedziale. W takim przypadku wymagamy, aby funkcja była monotoniczna w tym przedziale i w jednym z krańców przedziału osiągała wartość ujemną, a w drugim wartość dodatnią.

Proszę napisać program realizujący algorytm bisekcji dla dowolnej prostej funkcji (np. kwadratowej), z wykorzystaniem `Collections.binarySearch(...)`. Uwaga: nie chcemy tablicować wszystkich wartości funkcji (z zadaniem krokiem, np. 0.001) w danym przedziale, gdyż mogłoby to być zbyt kosztowne. Niech funkcja będzie na bieżąco obliczana w danym punkcie.

Wskazówki:

- użyj listy leniwej (poszukaj „guava lazy list” etc, np. na stackoverflow) z biblioteki Guava,
- do doboru funkcji matematycznej i przedziału wejściowego pomocna może być strona: <https://www.desmos.com/calculator>

#### 2. (KLIENT-SERWER) (1 pkt)

Proszę napisać konsolową aplikację klient-serwer z połączeniem TCP. Klient przekazuje serwerowi parę liczb naturalnych  $s$  i  $e$  (jak start, end), które określają przedział  $[s; e]$ . Zakładamy poprawność danych. Serwer ma policzyć i zwrócić ilość liczb pierwszych w tym przedziale, a także czas obliczeń. Serwer ma wykorzystać do obliczeń 2 wątki (jeśli więc na serwerze są co najmniej 2 nieobciążone rdzenie, to możemy oczekiwać przyspieszenia względem wersji 1-wątkowej).

Algorytm sprawdzenia, czy liczba naturalna  $m$  jest pierwsza:

- jeśli  $m = 1$ , to NIE jest pierwsza (choć akurat ta liczba nie jest też złożona),
- jeśli  $m = 2$  lub  $m = 3$ , to jest pierwsza,
- jeśli  $m > 3$ , to sprawdź podzielność przez 2 oraz przez wszystkie liczby nieparzyste  $\geq 3$  do  $\sqrt{m}$  włącznie; jeśli  $m$  nie dzieli się przez żadną z nich, to jest pierwsza, w przeciwnym razie jest złożona.

Algorytm ten uruchamiamy osobno dla każdej liczby z przedziału  $[s; e]$ .

Uwaga: podział przedziału na równe części jest nieoptymalny; warto eksperymentalnie poszukać lepszego podziału. Przykładowo, dla przedziału  $[20000; 39999]$  być może warto wziąć podprzedziały:  $[20000; 30999]$  i  $[31000; 39999]$ .

Proszę zaobserwować w Menedżerze zadań Windows (albo np. z użyciem narzędzia *htop* pod linuxem) obciążenie procesora za sprawą obliczeń na serwerze.

Prosty przykład aplikacji TCP klient-serwer:

<http://systembash.com/content/a-simple-java-tcp-server-and-tcp-client/>