

## Praca domowa 05 – prod

Termin zwrotu : 27 listopada godz. 23.00

Zadanie uznaje się za zaliczone, gdy praca oceniona zostanie na co najmniej 6 pkt.

W pliku tekstowym określonym parametrem linii komend `<input_file>` zapisana jest w formacie swobodnym pewna ilość liczb całkowitych. Korzystając z mechanizmów współbieżności należy zdefiniować trzy obiekty : producenta i dwóch konsumentów.

Producent odczytuje z pliku wejściowego podany ciąg liczb, umieszczając kolejne wartości w losowo określanych odstępach czasu w kolejce o długości określonej parametrem programu `<size>`. W przypadku, gdy kolejka jest pełna (nie można umieścić w niej kolejnego elementu) producent oczekuje przez losowo określony odstęp czasu i ponawia próbę wstawienia elementu do kolejki. Postępuje tak aż do momentu wyczerpania ciągu wejściowego, po czym kończy działanie. Momenty czasu w których producent podejmuje kolejne działania określone są rozkładem normalnym o parametrach :  $m = 10^{-1}$  sek (wartość – średnia) oraz  $\sigma = 10^{-2}$  (odchylenie standardowe)

Każdy z konsumentów obserwuje kolejkę wejściową. Konsumenti pobierają z kolejki wg następujących zasad :

Konsument 1 pobiera wyłącznie elementy których wartość jest  $\leq 10$ , przy czym kolejne próby podejmuje zgodnie z rozkładem normalnym o parametrach  $m = 1.5 * 10^{-1}$  sek oraz  $\sigma = 2 * 10^{-1}$ . Jeżeli w kolejce brak oczekiwanego elementu, ponawia próbę pobrania po kolejnym losowo określonym odstępie czasu.

Konsument 2 pobiera wyłącznie elementy których wartość jest  $\geq 10$ , przy czym kolejne próby podejmuje zgodnie z rozkładem normalnym o parametrach  $m = 1.2 * 10^{-1}$  sek oraz  $\sigma = 3 * 10^{-1}$ . Jeżeli w kolejce brak oczekiwanego elementu, ponawia próbę pobrania po kolejnym losowo określonym odstępie czasu.

Każdy z konsumentów kończy pracę w sytuacji w której stwierdza, iż kolejka wejściowa jest pusta (tj. nie ma w niej jakiegokolwiek elementu).

Każdy z konsumentów niezależnie oblicza wartość średnią ‘obsłużonych’ przez niego liczb. Wynikiem działania programu jest różnica wartości średnich wyznaczonych odpowiednio przez konsumenta 2 oraz konsumenta 1.

Program ma być zapisany wyłącznie w dwóch plikach : `Prod.java` zawierającym implementację modelu producent-konsument, oraz `Main.java` – zawierającym programem główny. Program nie może korzystać z jakichkolwiek bibliotek zewnętrznych..

Proces kompilacji musi być możliwy z użyciem komendy

```
javac -Xlint Prod.java Main.java
```

Uruchomienie programu winno być możliwe z użyciem komendy

```
java Main <input_file> <size>
```

Wynik końcowy (w strumieniu wyjściowym nie powinny pojawiać się jakiegokolwiek inne elementy – np. wydruki kontrolne) działania programu musi zawierać pojedynczą liczbę stanowiącą wartość wyniku z dokładnością do trzech miejsc dziesiętnych, np.

Wynik : 31.649

#### Wymagania :

- Klasa implementująca problem winna zostać zdefiniowana w pliku `Prod.java`
- Klasa implementująca mechanizm program główny (metoda `main`) winny być zdefiniowane w pliku `Main.java`
- W pliku `README.pdf` winien być zawarty szczegółowy opis organizacji struktur danych oraz szczegółowy opis zastosowanego algorytmu.
- Proces obliczenia rozwiązania winien się kończyć w czasie nie przekraczającym 3 min (orientacyjnie dla typowego notebooka). Po przekroczeniu limitu czasu zadanie będzie przerywane, i traktowane podobnie jak w sytuacji błędów wykonania (czyli nie podlega dalszej ocenie).

#### Sposób oceny :

- 1 pkt – **Kompilacja** : każdy z plików winien być kompilowany bez jakichkolwiek błędów lub ostrzeżeń (w sposób omówiony wyżej)
- 1 pkt – **Wykonanie** : program powinien wykonywać się bez jakichkolwiek błędów i ostrzeżeń (dla pliku danych wejściowych zgodnych z wyżej zamieszczoną specyfikacją) z wykorzystaniem omówionych wyżej parametrów linii komend
- 2 pkt – **README** : plik `README.pdf` dokumentuje w sposób kompletny i właściwy struktury danych, oraz opis przyjętej koncepcji algorytmu mnożenia
- 1 pkt – **Komentarze wewnętrzne** : czy program jest skomentowany w sposób zapewniający zrozumienie jego działania, oraz wyjaśniający warunki, które muszą zachodzić przed i po wykonaniu każdej z funkcji.
- 1 pkt – **Styl kodowania** : czy funkcje i zmienne posiadają samo-wyjaśniające nazwy ? Czy podział na funkcje ułatwia czytelność i zrozumiałość kodu ? Czy funkcje eliminują (redukują) powtarzające się bloki kodu ? Czy wcięcia, odstępy, wykorzystanie nawiasów itp. (formatowanie kodu) są spójne i sensowne ?
- 4 pkt – **Poprawność algorytmu** : czy algorytm został zaimplementowany poprawnie a wynik odpowiada prawidłowej (określonej zbiorem danych testowej) wartości.