

## Ćwiczenie 2 (za 10 punktów, środowisko jednomaszynowe)

Dla grup: wt/TP07, wt/TP09

1. Przestudiuj opis i zrealizuj w domu (**przed zajęciami wirtualnymi**) aplikację opisaną w **Ćwiczenie 2, Java RMI, Zadanie -część I**, plik: **Cw2-JavaRMI-1a.pdf**
2. Podczas zajęć zdalnych (labo wirtualne) zbuduj aplikację Java RMI scharakteryzowaną w **Zadanie - część II** ww instrukcji, w której *klient* zleca obliczenia pewnej **STAŁEJ** liczbie *serwer-workerów*. W aplikacji utwórz **DWA** (=2) *serwer-workery*, które będą zdolne do realizacji jednakowych zadań – p. niżej.

*Worker* otrzymuje od *klienta* – jako parametr *we* – dwie liczby typu **long**, np. **x**, **y**.

Po ich otrzymaniu *worker* uruchamia dwa podzadania:

*pzad1*: - w podanym zakresie [**x..y**] zlicza liczby pierwsze, **count** oraz znajduje największą liczbę pierwszą **max**, taką że  $x \leq \text{max} \leq y$ .

*pzad2*: - generuje 10 000 001 liczb losowych o wartościach z [**x..y**], sortuje zbiór i wyznacza medianę **med**.

Po zakończeniu obu podzadań *worker* zwraca klientowi wartości **count**, **max** i **med** oraz *dokładny czas zakończenia zleconego zadania*, **time**.

*Klient* udostępnia użytkownikowi aplikacji tekstowy interfejs.

W pojedynczym cyklu obliczeniowym *klient* zaprasza *użytkownika* do wprowadzenia:

$n$  – ilość zestawów danych ( $n \geq 3$ )

$x_1, y_1$  – parametry zadania dla workera (z kontrolą formatu i zakresu)

...

$X_n, y_n$  – parametry zadania dla workera ( z kontrolą formatu i zakresu)

Po wprowadzeniu wszystkich danych *klient* lokalizuje *serwer-workery* i wywołuje ich metody przekazując parametry zadania do wykonania. Gdy odbierze wyniki od wszystkich workerów formatuje je i udostępnia użytkownikowi.

*Uwaga 1*: klient na hard-kodowany algorytm wyboru workera (=serwer-workera) do realizacji zadania obliczeniowego (np. zadanie [dekomponowane na podzadania] dla workera nr 1 jest budowane na podstawie zestawów danych o numerach nieparzystych, tj. 1, 3, .. a dla workera nr 2 o numerach parzystych)

*Uwaga 2*: dla uproszczenia workery można wywoływać sekwencyjnie.

*Uwaga 3*: komunikacja za pomocą serializowanych obiektów.

3. Zdokumentuj – w pliku *nazwisko.docx* i/lub *nazwisko.pdf* - kolejne kroki kompilacji i uruchamiania/testowania części serwerowej i klienckiej zbudowanej aplikacji w środowisku jednomaszynowym.

**Bardzo ważne:** dokumentacja - w postaci kroków - musi zawierać krótkie komentarze/polecenia i screeny wprowadzonych komend i uzyskiwanych wyników.

4. Na zakończenie prześlij=wgraj rozwiązanie i wraz dokumentacją (punkt 3) na **eportal**. Obowiązują następujące reguły:

- wszystkie pliki mają być skompresowane do jednego pliku
- nazwa pliku ma mieć format: **X-Nazwisko-Imie.zip**  
gdzie **X** - numer zadania (taki jak na witrynie),
  - bez użycia spacji - zamiast nich są myślniki (!),
  - bez polskich znaków diaktrycznych (!)np.: **3-Ziolkowski-Michal.zip**
- wgrywany **projekt powinien być wyczyszczony** ("odchudzony")!
- uwaga: Plik można przesłać (ewentualne kolejne wersje) tylko trzy razy.

**Uwaga, ważne:**

Pliki z kodem i załączona dokumentacja musi być tak napisana, że szybko i łatwo będę w stanie powtórzyć wszystkie kroki kompilacji, uruchamiania i testowania aplikacji.