Programowanie równoległe. Przetwarzanie równoległe i rozproszone.

Laboratorium 6

Cel:

- Opanowanie podstaw tworzenia wątków w Javie.
- Opanowanie podstawowych metod synchronizacji w Javie.

Zajęcia:

1. Pobranie ze strony przedmiotu plików z kodem dwóch klas Javy: *Obraz.java* oraz *Histogram_test.java*. Klasa *Obraz* posiada dwa podstawowe elementy składowe: dwuwymiarową tablicę znaków (o zadanym przez użytkownika rozmiarze) i histogram odpowiadający tablicy, przechowujący liczbę wystąpień każdego ze znaków w tablicy (**wszystkie atrybuty klasy mają być prywatne** – dostęp tylko poprzez odpowiednie metody, tak w wersji sekwencyjnej, jak i w równoległych). W konstruktorze obiektu klasy *Obraz* tablica znaków jest wypełniana za pomocą podwójnej pętli po wszystkich wyrazach i instrukcji:
   `tab[i][j] = (char)(random.nextInt(liczba_znaków)+33)`
   (można zmniejszyć liczbę znaków losowanych i umieszczanych w tablicy, dla ułatwienia sprawdzania poprawności działania kodu; nie można dowolnie zwiększać – nie można przekroczyć zakresu znaków w kodowaniu ASCII)
   Histogram przechowywany jest w tablicy *histogram*, a klasa *Obraz* posiada trzy funkcje wykonywane sekwencyjnie – czyszczenie histogramu, obliczanie histogramu i drukowanie histogramu, tych funkcji nie należy zmieniać – powinny pozostać do testowania poprawności działania funkcji napisanych dla wersji równoległej
2. Uruchomienie kodu, sprawdzenie poprawności działania (np. na małych tablicach umożliwiających naoczne sprawdzenie liczby znaków w tablicy)
3. Rozszerzenie i modyfikacja kodu, w taki sposób, żeby obliczanie histogramu odbywało się w sposób równoległy, przy użyciu wątków Javy. Wątki wykonujące kod (tworzące i drukujące histogram w obiekcie klasy *Obraz)*, pracują poprzez wywoływanie odpowiednich metod klasy *Obraz* – należy pisać nowe metody tej klasy dla wykonania wielowątkowego.
   a) histogram obliczony sekwencyjnie powinien pozostać w klasie *Obraz* jako tablica do porównania wyników uzyskanych w kodzie równoległym, wyniki uzyskane równolegle przez wątki należy zapisać w innej w tablicy, np. *hist_parallel*
   b) kod równoległy należy sprawdzić pod względem poprawności – na zakończenie programu zawsze powinny być porównywane tablice: *histogram* i *hist_parallel* (można w tym celu zaprojektować specjalną funkcję w klasie *Obraz* zwracającą odpowiednie komunikaty o poprawności wykonania)
4. Wariant 1 – każdy wątek jest związany z jednym znakiem (np. przesyłanym w konstruktorze):
   a) należy napisać kod dla obiektów klasy dziedziczącej po klasie Thread (obiekty te będą wątkami)
   b) w funkcji *main* tworzone jest tyle wątków, ile jest różnych znaków w tablicy, *main* zarządza uruchomieniem wątków i oczekuje na zakończenie ich pracy (szkielet takiego wzorca jest zapisany w dostarczonej funkcji *main*)
   c) każdy wątek pracuje wywołując odpowiednie funkcje z klasy *Obraz* do obliczania odpowiedniego fragmentu histogramu i odrębną funkcję do drukowania fragmentu histogramu (obiekt klasy *Obraz* jest przesyłany w konstruktorze wątków)
   d) wszystkie funkcje dokonujące operacji na obrazie i wypisujące jego właściwości mają

być zdefiniowane w jednej klasie *Obraz*, której obiekt ma być przesyłany do wątków (można wykorzystać szkielet klasy *Obraz* w plikach na stronie). Wątki wywołują odpowiednie metody klasy *Obraz* – funkcje te mają być nowymi metodami, powstałymi przez modyfikacje istniejących. Programista decyduje, które metody mają być synchronizowane.

e) funkcja w klasie *Obraz* wyświetlająca zawartość histogramu w wersji równoległej powinna umożliwiać każdemu wywołującemu wątkowi (bez przerywania ze strony innych wątków) stworzenie ilustracji graficznej, która będzie miała ostateczną postać:

```
Wątek 1:  &  ==============
Wątek 2:  %  ======
Wątek 3:  $  ==================
```

itd. **(ocena)**

5. Wariant 2: podział zadania na pod-zadania odbywa się na zasadzie dekompozycji w dziedzinie problemu (gdzie dziedziną jest zbiór znaków ASCII, a dekompozycja jest przeprowadzana w sposób blokowy)
   a) implementacja wątków ma w tej wersji wykorzystywać interfejs *Runnable*
   b) nie należy usuwać klas istniejących, lecz tworzyć nowe klasy i metody, można skopiować wszystkie pliki z wariantu 1 do odrębnego katalogu i odpowiednio modyfikować kod wszystkich klas
   c) każdy wątek ma tworzyć składowe tablicy histogram w obiekcie klasy *Obraz* (poprzez wywołanie odpowiednich metod tej klasy) dla sobie przydzielonych znaków
   d) każdy wątek powinien, poprzez wywołanie odpowiedniej metody w klasie *Obraz*, wyświetlać po kolei swoje znaki oraz graficznie (bez przerywania ze strony innych wątków) przedstawiać liczbę jego wystąpień, co ma prowadzić do wydruku w postaci:

```
Wątek 1:  &  ==============
Wątek 2:  %  ======
Wątek 3:  $  ==================
Wątek 2:  @  =========
```

   itd. **(ocena)**

---------- 3.0 ----------

6. Rozszerzenie programu o kolejne klasy implementujące interfejs Runnable oraz kolejne metody w klasie Obraz, tym razem dla alternatywnego sposobu obliczania histogramu, gdzie dokonywany jest podział w dziedzinie problemu, ale dotyczący nie znaków, lecz obrazu (będącego tablicą znaków).
   - Każdy wątek operuje na wszystkich znakach, ale tylko na fragmencie tablicy
   - Dla tej dekompozycji można zastosować optymalne obliczanie histogramu (histogram[(int)tab[i][j]-33]++;)
   - Każdy wątek w efekcie otrzymuje histogram tylko dla swojej części tablicy
     - jakie dodatkowe operacje trzeba wykonać, żeby otrzymać ostateczny histogram dla całego obrazu?
7. W pierwszym wariancie napisanie kodu dla dekompozycji 1D (np. podziału cyklicznego wierszowego)
   a) przetestowanie programu – tym razem drukowanie może być dokonywane przez wątek główny po zakończeniu pracy przez inne wątki **(ocena)**
8. W kolejnych krokach mozna rozważyć inne warianty dekompozycji (kolumnowe, blokowe) lub przejść do zadania 9

---------- 4.0 ----------

9. Rozważenie wariantów podziału 2D
   a) na początek np. blokowy - każdy wątek otrzymuje prostokątny blok tablicy
   b) należy zastosować odpowiednią numerację wątków (2D)**(ocena)**
10. Napisanie kodu z połączeniem obu wariantów dekompozycji – pojedynczy wątek dokonuje obliczenia histogramu dla fragmentu tablicy i zadanego zbioru znaków

---------- 5.0 ----------

Warunki zaliczenia:

1. Obecność na zajęciach i wykonanie co najmniej kroków 1-5
2. Oddanie sprawozdania o treści i formie zgodnej z regulaminem ćwiczeń laboratoryjnych - z krótkim opisem zadania (cel, zrealizowane kroki, wnioski), kodem źródłowym programów w Javie oraz wydrukami wyników (wydruki wklejone jako obrazy z identyfikacją osoby przeprowadzającej obliczenia – zgodnie z regulaminem laboratoriów)