

## Zaawansowane programowanie obiektowe

### Lab. 7

(Programowanie z ograniczeniami, strumienie IO/pliki, kolekcje)

#### 1. (JACOP) (1 pkt)

Narysuj (na kartce czy np. w Paintcie) graf nieskierowany reprezentujący fragment mapy politycznej Europy (<http://www.worldmap.pl/mapy/mapa-polityczna-europy.png>), a mianowicie następujące państwa:

Polska, Białoruś, Ukraina, Słowacja, Czechy, Austria, Węgry, Niemcy, Dania, w taki sposób, że państwa są wierzchołkami w grafie, a krawędź między dwoma wierzchołkami istnieje, jeśli te państwa mają wspólną granicę.



Następnie napisz program korzystający z biblioteki JaCoP, który znajdzie pokolorowanie tego grafu o minimalnej liczbie kolorów (w taki sposób, aby wierzchołki połączone wspólną krawędzią nie miały tego samego koloru).

Proszę reprezentować graf przy pomocy macierzy sąsiedztwa ([http://opendatastructures.org/ods-java/12\\_1\\_AdjacencyMatrix\\_Repres.html](http://opendatastructures.org/ods-java/12_1_AdjacencyMatrix_Repres.html)), ale dodatkowo zwizualizować na konsoli w postaci typu jak poniżej (x – istnieje krawędź, . – nie istnieje).

```
. . x .
. . xx
xx . x
. xx .
```

Założ, że nie potrzeba więcej niż 5 kolorów i przy rozwiązaniu wypisz kolory wg przyporządkowania: 1 --> czerwony, 2 --> czarny, 3 --> niebieski, 4 --> zielony, 5 --> żółty.

Wskazówka: skorzystaj z przykładu z <http://jacopguide.osolpro.com/guideJaCoP.html>, sekcja 1.1.

#### 2. (PLIKI) (0,5 pkt)

Napisz program generujący  $N$  liczb losowych o rozkładzie normalnym (gaussowskim) o zadanych parametrach  $m$  (średnia) i  $\sigma$  (odchylenie standardowe) i zapisujący je do pliku binarnego. Następnie plik ten jest odczytywany, liczba po liczbie, a dane zapisywane do

nowego pliku (tekstowego), każda liczba w osobnym wierszu. Dodatkowo należy zwizualizować (przy pomocy Excela lub w Pythonie/matplotlib) histogram tych liczb, aby sprawdzić, czy rzeczywiście rozkład przypomina krzywą dzwonową Gaussa. Zapis do pliku tekstowego ma wykorzystywać konwencję formatu przyjętą w Polsce (tj. przecinki zamiast kropek dziesiętnych).

Parametry  $N$  (liczba całkowita),  $m$  i  $\sigma$  (liczby zmiennoprzecinkowe) mają być argumentami wiersza poleceń. Przy pomocy mechanizmu asercji dopilnuj, aby odchylenie standardowe nie było liczbą ujemną.

Wskazówki.

- a) Generacja liczb o rozkładzie normalnym – poszukaj w klasie Random.
- b) Zapis/odczyt liczb w postaci binarnej: klasy DataOutputStream / DataInputStream.
- c) Utwórz odpowiedni obiekt klasy NumberFormat.

### 3. (KOLEKCJE, PLIKI) (1 pkt)

Stefan testuje pewną klasę generatorów liczb pseudolosowych.

Wszystkie generatory z tej klasy mają taką właściwość, że jeśli jakaś liczba pojawi się w wygenerowanym ciągu po raz drugi, to wszystkie następne również są zgodne z następnikami tamtego wcześniejszego wystąpienia.

Przykład: jeśli mamy wygenerowany ciąg (liczb z  $[0...9999]$ ):

1057, 86, 31, 6232, 9401, 7070, 31,

to ciąg dalszy musi już wyglądać tak:

6232, 9401, 7070, 31, 6232, 9401, 7070, 31, 6232, 9401, 7070, 31, ...

(czyli pojawił się okres o długości 4 elementów)

Generator uznajemy za kiepski, jeśli pojawi w nim okres o długości  $\leq 1000$  elementów.

Niestety, Stefan nie potrafi testować generatorów pod kątem ich jakości inaczej niż generując pseudolosowe ciągi i szukając w nich okresów. Co gorsza, ograniczenie pamięciowe nie pozwala mu na przechowywanie więcej niż 1001 elementów w pojedynczej kolekcji (typu dowolnego, czyli jakiejś listowej, zbiorowej lub słownikowej).

W plikach gen1.dat, gen2.dat, gen3.dat są wygenerowane ciągi liczb typu int (nieujemnych, być może dużych). Dla każdego z tych plików należy odpowiedzieć czy zawiera on okres (o długości  $\leq 1000$ ), a jeśli tak, to na których dwóch pozycjach (numerując od zera) znajduje się pierwsza para powtarzających się elementów, i wypisać również długość okresu.

Przykładowo, dla ciągu:

57, 86, 31, 2, 70, 94, 31, 2, 70, 94, 31, 2, 70, 94, 31, 2, 70, 94 (założmy, że tu koniec)

wyjście ma wyglądać tak:

BAD SEQ, 2, 6, period length = 4

Ale gdybyśmy mieli:

93, 14, 6, 5, 82, 50, 38, 43, 19, 22, 33, 46

to na wyjściu byłoby:

GOOD SEQ

Wskazówka: pomyśl o odpowiedniej kolekcji.