Analiza danych epidemicznych w języku Java

Cel ćwiczenia

Celem ćwiczenia jest przygotowanie aplikacji prezentującej w postaci graficznej bieżące dane epidemiczne. W pierwszej kolejności zmiany ilości potwierdzonych przypadków osób chorujących na COVID-19 w funkcji czasu.

Instrukcja

1. Zlokalizuj źródło danych

   Bieżące dane epidemiczne prezentowane w tym ćwiczeniu należy pobrać z repozytorium JHU, pod adresem: https://github.com/CSSEGISandData/COVID-19 .

   Nie jest nam potrzebne całe repozytorium, lecz pojedynczy plik `time_series_covid19_confirmed_global.csv` zlokalizowany w katalogu csse_covid_19_data i podkatalogu `csse_covid_19_time_series` . Plik ten dostępny jest bezpośrednio z internetu pod adresem:

   https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv

   W pliku tym zawarte są ilości zakażonych w funkcji czasu z podziałem na kraje, a w niektórych przypadkach także na regiony czy prowincje.

2. Uzyskaj dostęp do bieżących danych

   Program ma pobierać z serwera najnowszy plik po każdorazowym uruchomieniu. W tym celu należy skorzystać standardowo z klasy Scanner i metody `openstream()` klasy URL .

   ```
   URL url = new URL("<ADRES_ZASOBU>");
   Scanner sc = new Scanner(url.openStream());
   ```

3. Badanie struktury danych

   W następnej kolejności należy w standardowy sposób wyświetlić zawartość pobranego pliku. Przy okazji proszę zwrócić uwagę na jego zawartość i strukturę. W pierwszej kolejności będziemy wyświetlali dane epidemiczne Polski, dlatego też przede wszystkim proszę przyjrzeć się danym zawartym w wierszu odpowiadającym Polsce.

4. Ekstrakcja wartości liczbowych

   Każda linia pobranego pliku zawiera wartości liczbowe, odpowiadające ilości zakażonych (z pozytywnym wynikiem testu) w dniach od 22-01-2020 do dnia wczorajszego oraz kilka dodatkowych danych, takich jak nazwa regionu, nazwa kraju i para współrzędnych geograficznych.

   Z uwagi na fakt, że dane te rozdzielone są przecinkami, można je łatwo wyekstrahować i zapisać w zmiennych. Tak więc po pobraniu linii tekstu za pomocą Scannera i zapisaniu jej do tymczasowej zmiennej

typu `String`, należy podzielić ją na indywidualne łańcuchy tekstowe zlokalizowane między przecinkami, a następnie zapisać do tymczasowej tablicy typu `String`.

```
String linia = sc.nextLine();

String[] token = linia.split().useDelimiter(",");
```

Tablica ta ma wielkość `token.length` i pierwsze cztery elementy tablicy zawierają elementy opisowe, a następne elementy zawierają wartości które chcemy pokazać na wykresie.

5. "Poland"

Zasadniczo nie planujemy analizować wszystkich danych zawartych w pliku, lecz jedynie wartości z wybranych miejsc, w tym przede wszystkim z Polski. W związku z tym, trzeba wyszukiwać potrzebny wiersz i tylko z niego odczytać potrzebne wartości.

Nazwy krajów zapisane są w drugiej kolumnie, co odpowiada drugiemu elementowi tablicy token, czyli `token[1]`. Należy przeszukiwać kolejne wiersze pliku tak długo, aż ten element będzie zawierał słowo "Poland".

```
if ("Poland".contains(token[1])){
 … zrób coś
}
```

Do porównania wartości elementu `token[1]` ze stringiem `"Poland"` nie wykorzystujemy znaku równości, ponieważ nie chcemy porównywać obiektów, tylko ich zawartości dlatego też używamy metody `contains()` klasy `String` . W porównaniu nieprzypadkowo najpierw napisałem ciąg `"Poland"` - wywołanie metody `contains()` na stringu, który nie istnieje doprowadziłoby do wystąpienia błędu i zatrzymania programu. Natomiast po zamianie w najgorszym przypadku warunek po prostu nie zostanie spełniony.

6. Konwersja tekstu do wartości liczbowych

Wartości zapisane w tablicy token są łańcuchami tekstowymi i przed wykorzystanie ich do wizualizacji należy zamienić je na wartości liczbowe. W tym celu należy skorzystać z metody parseDouble() klasy Double.

```
double[] tested = new double[token.length - 4];

for ( int i = 0; i < tested.length; i++ )

     tested[i]= Double.parseDouble(token[i + 4]);
```

O tej pory dysponujemy tablicą `tested` zawierającą ilości zakażonych w okresie czasu od 22 stycznia do dnia wczorajszego w postaci wartości liczbowych i możemy przystąpić do sporządzenia wykresu ilości zdiagnozowanych w funkcji czasu.

7. StdDraw.java

StdDraw jest biblioteką pozwalającą w łatwy sposób na rysowanie na ekranie komputera prostych figur

geometrycznych. Dzięki niej można w prosty sposób stworzyć różnorodne wykresy. Biblioteka dostępna jest pod adresem podanym poniżej, skąd należy ją pobrać i zapisać do pliku o nazwie (dokładnie) StdDraw.java .

https://introcs.cs.princeton.edu/java/stdlib/StdDraw.java

Następnie w Eksploratorze Pakietów programu Eclipse (okienko z lewej strony) należy odnaleźć swój projekt, rozwinąć go odszukać pakiet do którego zamierzamy zapisać importowaną klasę, a następnie kliknąć ikonę tego pakietu prawym przyciskiem myszy.

Następnie z pojawiającego się menu należy wybrać pozycję `Import⋯` , następnie `General > File System > Next` . W okienku Import w pierwszej kolejności należy wskazać katalog źródłowy z którego chcemy wczytać plik (`From directory`), a następnie z pojawiającej się listy w okienku z prawej strony plik StdDraw.java , który chcemy zaimportować, a następnie należy kliknąć przycisk `Finish.`

Po zaimportowaniu biblioteki do projektu zostanie wskazany błąd w pierwszej linii kodu źródłowego biblioteki. Błąd ten należy naprawić akceptując sugestię zmiany nazwy pakietu albo zmieniając nazwę na właściwą ręcznie.

8. Podstawy użytkowania StdDraw

Nie ma możliwości stworzenia instancji klasy StdDraw. Biblioteki tej używa się używa się wywołując jej metody statyczne. Dostępne właściwości, metody oraz kilka przykładów zawarte jest w kodzie źródłowym.

Okno pojawiające się po użyciu jakiejkolwiek metody klasy StdDraw ma domyślnie rozmiar 512 na 512 pikseli, można je jednak dowolnie modyfikować w obu kierunkach poprzez wywołanie metody setCavnasSize .

```
StdDraw.setCanvasSize(int canvasWidth, int canvasHeight);
```

Domyślnie obszarem rysowania jest kwadrat o współrzędnych od (0,0) do (1,1). Możliwe jednak jest dowolnie modyfikowanie zakresu współrzędnych X i Y tak, aby umieścić figury o dowolnej lokalizacji w układzie współrzędnych XY. Służą do tego dwie metody `setXScale oraz setYScale` .

```
StdDraw.setXscale(double min, double max);
StdDraw.setYscale(double min, double max);
```

Wygodną metodą do rysowania punktów na ekranie jest metoda point, pozwalająca na narysowanie punktu o współrzędnych x i y. Punkt ten jest kołem wypełnionym kolorem czarnym o domyślnym promieniu 0,002. Zarówno wielkość tego koła jak i jego kolor można zmieniać odpowiednio za pomocą metod `setPenRadius` oraz `setPenColor`. W przypadku zmiany domyślnego koloru na inny jako argument metody można skorzystać ze statycznych właściwości klasy `Std.NAZWA_KOLORU`.

```
StdDraw.setPenColor(Color color);
StdDraw.setPenRadius(double radius);
StdDraw.point(double x, double y);
```

9. Wizualizacja danych w StdDraw

Dysponując zbiorem współrzędnych punktów które chcemy narysować na wykresie w pierwszej kolejności należy ustalić poziomą i pionową skalę, czyli zakres zmienności wartości na osi X oraz na osi Y. W tym celu należy znaleźć minimalne i maksymalne wartości obu tych zakresów. W przypadku osi X nie jest to konieczne, ponieważ punktu będą rysowane po kolei, natomiast w przypadku osi Y praktycznie są nieznane, ponieważ za każdym razem pobieramy nowy plik.

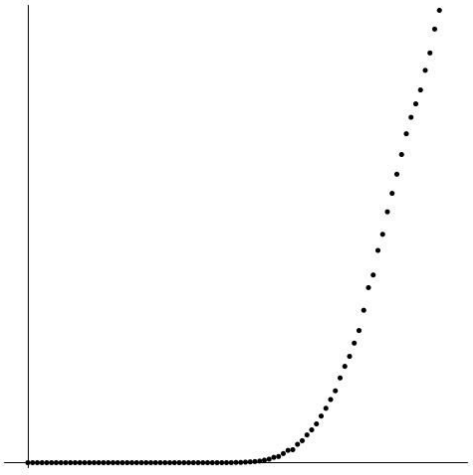
```
double maxY = Double.MIN_VALUE;

for( int i = 0; i < tested.length; i++ ){

     if ( tested[i] > maxY ) maxY = tested[i];
}

StdDraw.setXScale(-10, tested.length+10);
StdDraw.setYScale(-10, maxY+10);
```

W pierwszej kolejności warto narysować obie osie współrzędnych. W tym celu należy skorzystać z metody `line(double x1, double y1, double x2, double y2)`.

```
StdDraw.line(-10, 0, tested.length + 10, 0); // oś X
StdDraw.line(0, 10, 0, maxY + 10 );          // os Y
```

Następnie rysujemy wykres, umieszczając w okienku punkt po punkcie.

```
for ( int i = 0; i < tested.length; i++ ) StdDraw.point(i, tested[i]);
```



10. Modyfikacje wykresu

W następnej kolejności można wprowadzić wiele modyfikacji do wykresu, np. rysując tylko te punkty których wartość na osi Y jest znacząca i pomijając wartości bliskie zeru, dzięki czemu wykres będzie znacznie bardziej wyraźny. W tym celu ze zbioru danych należy usunąć niektóre punkty (na przykład

przepisując do mniejszej tablicy), a następnie zmodyfikować zakres skali na osi Y.

Do wykresu można dodać także tytuły, etykiety, groty strzałek na osiach (metoda `polygon`), czy kolejne zbiory punktów danych epidemicznych z innych krajów.