

Kolokwium 1, 2021

Przygotowane zostały dwa pliki CSV wygenerowane na podstawie danych o zakażeniach i zgonach z powodu COVID-19 zagregowanych przez system działający w Uniwersytecie Johna Hopkinsa w Baltimore. Plik `confirmed_cases.csv` zawiera informację o potwierdzonych zakażeniach, a plik `deaths.csv` o zgonach. Oba pliki mają taką samą strukturę: nagłówki, liczbę kolumn i wierszy, ale różnią się danymi liczbowymi.

Struktura plików: W pierwszym wierszu poczynając od drugiej kolumny znajdują się nazwy państw. W drugim wierszu znajdują się nazwy prowincji lub terytoriów zależnych tych państw. Jeżeli państwo nie posiada prowincji, w drugim wierszu pojawia się napis "nan" (patrz np. Afganistan). Jeżeli państwo ma prowincje, jego nazwa w pierwszym wierszu jest powtarzana przed każdą z prowincji (patrz np. Australia).

W pierwszej kolumnie, poczynając od trzeciego wiersza znajdują się daty zapisane w amerykańskim formacie M/d/yy. Na przecięciu daty i państwa/prowincji znajduje się, w zależności od pliku, liczba zakażeń lub zgonów. W szczególnym przypadku liczby te mogą być ujemne, należy je traktować tak, jak pozostałe.

W obu plikach kolumny rozdzielone są średnikami.

***Uwaga 1.** Przedstawione poniżej kroki stanowią propozycję kolejności rozwiązywania zadania. Po przeczytaniu całości można zdecydować o zmianie kolejności rozwiązywania.*

***Uwaga 2.** Jeżeli w treści zadania pojawia się sformułowanie o założeniu poprawności, nie ma konieczności sprawdzania tego warunku.*

***Uwaga 3.** Należy wysyłać wyłącznie pliki `.java` umieszczone w pojedynczym katalogu i spakowane do formatu `zip` lub `tar.gz`.*

Krok 1.

Napisz klasę abstrakcyjną `Country`, po której dziedziczyć będą klasy `CountryWithoutProvinces` oraz `CountryWithProvinces`.

Na tym etapie klasa `Country` powinna mieć prywatne, ostateczne pole `name` klasy `String` zawierające nazwę państwa oraz publiczne metody: akcesor do pola `name` oraz publiczny konstruktor przyjmujący jako argument nazwę państwa i ustawiający go w polu `name`.

Klasa `CountryWithoutProvinces` nie dodaje żadnych pól i w swoim konstruktorze wywołuje konstruktor klasy nadrzędnej.

Klasa `CountryWithProvinces` powinna posiadać prywatną tablicę obiektów `Country`, których nazwy powinny odpowiadać nazwom prowincji. Konstruktor klasy `CountryWithProvinces` powinien dodatkowo przyjmować jako parametr taką tablicę i ustawiać ją w polu klasy.

Krok 2.

W klasie `Country` zdefiniuj statyczne, prywatne pola zawierające ścieżkę do obu plików CSV. Napisz statyczną metodę klasy `Country` o nazwie `setFiles` ustawiającą te dwa pliki na wartości swoich argumentów. Metoda ta powinna zweryfikować, czy pliki istnieją i można je odczytać. Jeżeli nie będzie to

możliwe, należy rzucić wyjątek `FileNotFoundException` podając mu jako argument konstruktora ścieżkę do błędnego pliku.

W klasie `Country` napisz publiczną, statyczną metodę `fromCsv`, która przyjmie jako argument napis zawierający nazwę kraju, a zwróci polimorficzny obiekt typu `Country`. Metoda `fromCsv` powinna otwierać i zamykać pliki i może założyć, że ścieżki do nich są poprawne.

Krok 3.

Zdefiniuj klasę wyjątku `CountryNotFoundException` tak, aby niemożliwa była kompilacja bez przechwycenia go. Wyjątek powinien być napisany tak, aby wywołanie metody `getMessage()` zwróciło nazwę nieznanego państwa.

Wewnątrz klasy `Country` zdefiniuj prywatną, statyczną klasę `CountryColumns`. Klasa powinna posiadać publiczne, ostateczne, całkowite pola `firstColumnIndex`, `columnCount` ustawiane przy pomocy konstruktora.

W klasie `Country` zdefiniuj prywatną, statyczną metodę `getCountryColumns`, która otrzyma jako parametry: napis będący pierwszym wierszem pliku CSV oraz napis zawierający poszukiwane państwo. Metoda powinna zwrócić obiekt klasy `CountryColumns` zawierający informację o początkowej kolumnie oraz liczbie kolumn poświęconej państwu. Jest to jednocześnie informacja, czy państwo posiada prowincje.

Wywołaj metodę `getCountryColumns` wewnątrz metody `fromCsv` i przekaz dalej rzucany przez nią wyjątek `CountryNotFoundException`.

Krok 4.

Samodzielnie zaprojektuj w klasie `CountryWithoutProvinces` zawartość pozwalającą zapisać ile danego dnia w tym państwie było zakażeń i zgonów. Klasa powinna umożliwiać zapisanie wielu takich wpisów.

Zdefiniuj publiczną metodę `addDailyStatistic`, przyjmującą jako argumenty datę oraz dwie liczby całkowite - zachorowania i zgony, która dodaje je do zaproponowanej struktury. Daty należy zapisywać jako obiekty klasy `LocalDate`.

Krok 5.

W metodzie `fromCsv` w zależności od rodzaju państwa utwórz obiekt klasy `CountryWithoutProvinces` lub `CountryWithProvinces`. Następnie dla kolejnych linii z danymi liczbowymi wywołaj metodę `addDailyStatistic` na rzecz:

- obiektu `CountryWithoutProvinces`, lub
- kolejnych komórek tablicy prowincji obiektu `CountryWithProvinces`.

Należy zapisać statystyki dla wszystkich dat znajdujących się pliku.

Krok 6.

Napisz statyczną metodę przeciążającą `fromCsv`, która zamiast pojedynczej nazwy kraju przyjmuje tablicę takich nazw. Metoda powinna zwrócić tablicę obiektów `Country`. Jeżeli metoda `fromCsv(String)`

(poprzednia) rzuca wyjątek `CountryNotFoundException`, należy wyświetlić na standardowym wyjściu wartość zwracaną przez metodę `getMessage()` wyjątku i pominąć to państwo w wynikowej liście.

Krok 7.

W klasie `Country` napisz publiczne, czysto wirtualne metody `getConfirmedCases` oraz `getDeaths`, które przyjmują jako parametr datę, a zwracającą odpowiednio liczbę zdiagnozowanych przypadków i liczbę zgonów tego dnia. Zakładamy poprawność podanej daty.

Metody te powinny być zaimplementowane w klasach dziedziczących po `Country`:

- w `CountryWithoutProvinces` należy podać wartości zapisane w zdefiniowanej strukturze,
- w `CountryWithProvinces` należy wywołać tę metodę rekurencyjnie dla wszystkich prowincji i zsumować wynik.

Krok 8.

W klasie `Country` napisz publiczną, statyczną metodę `sortByDeaths`, która przyjmie listę obiektów `Country` oraz dwie daty: początkową i końcową. Metoda powinna posortować tablicę malejąco według liczby śmierci w okresie między datą początkową, a końcową włącznie z nimi. Zakładamy poprawność podanych dat oraz, że początkowa jest wcześniejsza niż końcowa.

Krok 9.

W klasie `Country` napisz publiczną metodę `saveToDataFile`, która przyjmie ścieżkę do pliku wynikowego. Zakładamy, że jest ona poprawna. Metoda powinna utworzyć plik składający się z trzech kolumn oddzielonych tabulatorami. W pierwszej kolumnie powinny znaleźć się daty w formacie `d.MM.yy` w drugiej liczba zdiagnozowanych przypadków w tym dniu, a w trzeciej liczba zgonów w tym dniu. W kolejnych wierszach pliku wynikowego należy zapisać wszystkie daty i odpowiadające im statystyki dostępne w plikach CSV.

(w wolnym czasie)

Zawartość tak utworzonego pliku można zwizualizować np. przy pomocy programu `Gnuplot`, którego wersja online jest dostępna na stronie <http://gnuplot.respawned.com/>

Zawartość pliku należy skopiować do pola "data", a w polu "plot script" należy wówczas wpisać:

```
set terminal svg size 800,300 enhanced fname 'arial' fsize 10 butt solid
set output 'out.svg'
set key left
set xdata time
set timefmt "%d.%m.%y"
set format x "%m.%y"
plot "data.txt" using 1:2 title 'Confirmed Cases' with lines,\
     "data.txt" using 1:3 title 'Deaths' with lines
```

Punktacja:

Klasa Country z pominięciem wyszczególnionych niżej metod - 5,5 p.

Metoda fromCsv z kroków 2-5 - 12,5 p.

Metoda fromCsv z kroku 6 - 2,5 p.

Metoda getCountryColumns - 6,5 p.

Metoda sortByDailyDeaths - 4 p.

Metoda saveToFile - 6 p.

Klasa CountryNotFoundException - 3 p.

Klasa CountryWithoutProvinces - 6 p.

Klasa CountryWithProvinces - 4 p.

W sumie 50 p.