

Programowanie Obiektowe

Pliki

Zadanie oceniane nr 3b

20-05-2021

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie_oceniane_3b. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

"Zbiory przybliżone w praktyce"



Prof. dr hab. inż. Zdzisław Pawlak (PW) był twórcą interesującej Teorii Zbiorów Przybliżonych, pozwalającej min. na ekstrakcję zależności ukrytych w dużych ilościach danych. Dzięki niej można na przykład ustalić, który obserwowany parametr ma istotny wpływ na przynależność danego przypadku do jakiejś grupy, co stanowi ciekawe narzędzie wspierające tzw. systemy ekspertowe. Tematem dzisiejszego zadania będzie implementacja prostego mechanizmu parsującego z pliku wyniki obserwacji zachorowań na COVID-19, będącego dobrym punktem wyjścia do implementacji wspomnianej tutaj teorii (już poza zadaniem). W pliku zapisano objawy wykryte wśród kobiet i mężczyzn oraz informację czy zwiastowały one infekcję dobrze znanym nam wirusem czy też nie. Należy je odczytać, zaprojektować rodzinę klas do ich przechowywania, utworzyć na ich podstawie obiekty, wypełnić je wczytanymi danymi, dodać do kolekcji i coś jeszcze z nimi zrobić. Do dzieła.

Prace do wykonania:

1. Zastanowić się nad naturą danych, które przyjdzie Państwu parsować.

Plik pacjenci.txt zawiera pobieżne wyniki badań kobiet oraz mężczyzn pod kątem zakażenia Covid. Każda osoba ma swój wiersz w którym informacje są odseparowane znakiem #.

Linijka opisująca mężczyznę ma następujących schemat:

id#pleć#wiek#temperatura#kaszel?#czy był w wojsku?#Covid dodatni?

Przykład: 1#M#25#39.2#true#true#true

Wiersz opisujący kobietę jest zdefiniowany tak:

id#pleć#wiek#temperatura#kaszel?#nazwisko rodowe?#Covid dodatni?

Przykład: 2#K#37#39.6#false#Kowalska#true

2. Model danych

Stworzyć klasy których pola zawierają miejsca na informacje wczytane z pliku (pamiętając że dane o kobietach różnią się od danych o mężczyznach). Dla każdej wyizolowanej danej (temperatura, kaszel, covid dodatni, itp.) powinno w danej klasie być stworzone pole odpowiedniego typu (temperatura to liczba, a covid dodatni to wartość logiczna). Należy pamiętać, żeby obiekty klas opisujących dane kobiety oraz klas opisujących mężczyzn można było wrzucić do jednej kolekcji.

Należy pamiętać że:

- To czy dany obiekt jest równy drugiemu zależy od tego czy ich id (pobrane z pliku wraz z innymi danymi) są takie same
- Jedna linijka z pliku = jeden obiekt
- Każda klasa ma mieć zaimplementowaną swoją reprezentację tekstową
- poprawne zasięgi, metody dostępne (jeśli potrzebne), odpowiednia hierarchia to sprawy tutaj oczywiste.

3. Klasa MyParser

- Klasa zawiera metodę parsePatients() zwracającą kolekcję obiektów z których każdy zawiera wczytane dane dotyczące jednego przypadku (ileś tam wyników kobiet i ileś tam wyników mężczyzn)
- Plik pacjenci.txt znajduje się w projekcie (najlepiej w innym katalogu niż klasy). Dostęp do niego opiera się na ścieżce względem classpath (a nie miejsca uruchomienia aplikacji).
- Jest on zakodowany w UTF-8.

4. Klasa MyProcessor

- Zawiera metodę getWomanData, pobierającą kolekcję którą zwraca metoda parsePatients() i zwracającą kolekcję wyników samych kobiet
- Zawiera metodę trimData, pobierającą kolekcję zwróconą przez parsePatients, a zwracającą kolekcję bez wyników gdzie temperatura była mniejsza niż 36.4

5. Rozszerzenie

Zakładamy, że dane z pliku procesowane przez metodę parsePatients są poprawnie ustrukturyzowane, a co najważniejsze Id muszą być unikalne. Zatem w przypadku gdy dwa obiekty będą miały to samo id, należy potraktować taką okoliczność jak sytuację na obsłudze której się z nikim nie umwialiśmy i przerwać pracę w odpowiedni dobrze znany Państwu sposób. Przykładowy plik ze zdublowanym id to pacjenci_bad.txt. Demonstrator jak zwykle obecny.