Programowanie Obiektowe

Pliki

Zadanie oceniane nr 3a

24-05-2021

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie_oceniane_3a. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

"Kebab"

W dniu dzisiejszym zajmiemy się zestawieniem dań, które pewne bistro wydało w ciągu jednego dnia swojej działalności. Dane dotyczące sprzedaży znajdują się w pliku, z którego trzeba je wczytać aby stworzyć na ich podstawie kolekcję obiektów. Te z kolei poddane zostaną dalszej obróbce. Zaczynamy!

Prace do wykonania:

1. **Zastanowić się nad naturą danych, które przyjdzie Państwu parsować.**
   Plik kebaby.txt zawiera listę wydanych dań. Każde z nich jest reprezentowane przez jeden wiersz w którym informacje są odseparowane ciągniem znaków: ---. Plik zawiera dwa rodzaje dań: w cieście i na talerzu, różniących się między sobą ilością danych je opisujących.

   Linijka opisująca kebab w cieście ma następujących schemat:
   rozmiar(S/M/L)---czy grube ciasto (yes/no)---mięso (kurczak/wołowina/mix)---waga---cena---
   Przykład: S---false---kurczak---250--12.5---

   Wiersz opisujący kebab na talerzu jest zdefiniowany tak:
   rozmiar(S/M/L)---dodatek (ryz/kasza/frytki)---mięso (kurczak/wołowina/mix)---waga---czy na wynos (true/false)---cena---
   Przykład: L---ryz---kurczak---371---false---24.5---

   Opisy nie zawierają znaków diakrytycznych.

2. **Model danych**
   Stworzyć klasy których pola zawierają miejsca na informacje wczytane z pliku (pamiętając że kebab na talerzu i kebab w cieście to dwa różne typy dań, które są opisywane przez różne zbiory danych). W celu rozróżnienia typów dań mogą przydać się metody Scannera z rodziny hasNextXXX. Dla każdej wyizolowanej danej (rozmiar, cena, rodzaj mięsa, itp.) powinno w danej klasie być stworzone pole odpowiedniego typu (należy pamiętać jak opisuje się typy mogące przyjmować tylko określony zbiór wartości). Ponadto trzeba mieć na uwadze, aby obiekty klas opisujących kebab na talerzu oraz klas opisujących kebab w cieście można było wrzucić do jednej kolekcji.

   Należy pamiętać że:
   - zakładamy że dwa obiekty są sobie równe jeśli wartości wszystkich ich pól są takie same (najlepiej wygenerować te reguły automatycznie)
   - jedna linijka z pliku = jeden obiekt
   - każda klasa ma mieć zaimplementowaną swoją reprezentację tekstową
   - paprawne zasięgi, metody dostępowe (jeśli potrzebne), odpowiednia hierarchia to sprawy tutaj oczywiste.
   - Tekst w pliku jest kodowany w UTF-8

3. **Klasa KebabParser**
   - Klasa zawiera metodę parseKebabs zwracająca kolekcję obiektów z których każdy zawiera wczytane dane dotyczące jednego dania. Należy ustalić typ parsowanego dania na podstawie informacji które mamy.
   - Plik kebaby.txt znajduje się w projekcie (najlepiej w innym katalogu niż klasy). Dostęp do niego opiera się na ścieżce względem classpath (a nie miejca uruchomienia aplikacji).

4. **Klasa KebabProcessor**
   - Zawiera metodę getSmallTellerKebabs, pobierającą kolekcję takiego samego typu jak kolekcja zwracana przez parseKebabs() i zwracająca kolekcję tylko dla obiektów konkretnego typu (kebab na talerzu) zawierającą wyizolowane dania: kebab na talerzu conajmniej rozmiaru M (średni) z kurczakiem.
   - Zawiera metodę getSummary, wypisująca wszystkie dania na konsolę. Na końcu

pojawia się średnia cena wszystkich dań oraz zsumowana waga wszystkich dań. Metoda ta przyjmuje kolekcję na której pracuje.

5. **Rozszerzenie**
   Zakładamy, że dane z pliku procesowane przez metodę parseKebabs są poprawnie ustrukturyzowane. Jedyny problem jaki może się ujawnić to nieznany rodzaj mięsa (inny niż kurczak, wołowina lub mix (taka sytuacja może mieć miejsce podczas parsowania pliku kebaby_bad.txt). Jest to okoliczność której nasza aplikacja nie musi obsługiwać, więc powinna przerwać swoje działanie w dobrze znany sposób. Demonstrator jak zwykle obecny.