

Algorithms and Data Structures / List No. 1

Wroclaw University of Science and Technology - Faculty of Computer Science and Management
Marcin Stachowiak (marcin.stachowiak@pwr.edu.pl) - kurs poprawkowy

Keywords—*wczytywanie danych z pliku, algorytm wczytywania, optymalizacja wydajności*

I. ZADANIE 1 (WCZYTYWANIE DANYCH)

Zapoznaj się z danymi umieszczonymi w lokalizacji *airlineroute/algorithms/resources*. Są one podzielone na 3 pliki tekstowe: *airports.dat* (dane o lotniskach), *airlines.dat* (dane o liniach lotniczych) oraz *routes.dat* (dane o kursach realizowanych przez poszczególne linie lotnicze).

Napisz algorytm, który po podaniu ścieżki do każdego z plików wczyta ich zawartość do lokalnej bazy danych. Bazę reprezentuje obiekt zawierający trzy listy.

- Obiektowe reprezentacje danych (*Airport.java*, *Airline.java* i *Route.java*) są już zaimplementowane i dostarczone w pakiecie *com.asid.algorithms.entity*.
- Obiekt reprezentujący bazę danych (*InmemoryDataBase.java*) jest już zaimplementowany i dostarczony w pakiecie *com.asid.algorithms.database*.
- Należy zapoznać się z relacjami istniejącymi pomiędzy obiektami *Airport*, *Airline* i *Route* oraz metodami dostępnymi w klasie *InmemoryDataBase.java*.
- Logikę wczytującą poszczególne rekordy z plików do bazy danych należy zaimplementować tak, aby mogła zostać ona uruchomiona przez wywołanie metod z interfejsu *com.asid.algorithms.database.IDataLoader*.
- Jeśli wiersz w pliku tekstowym zaczyna się znakiem #, to powinien zostać pominięty.
- Przed otwarciem połączenia do pliku i dodaniem danych do bazy należy zadeklarować której reprezentacji interfejsu *List* będzie używać baza danych do przechowywania obiektów. Należy posłużyć się jedną z dwóch stworzonych w zadaniu 1 struktur, np. *inmemoryDataBase.setAirports(new CustomLinkedList())*.
- Dane należy wczytać w następującej kolejności. Najpierw plik *airports.dat*, następnie *airlines.dat*, a na końcu *routes.dat*. Podczas tworzenia obiektu *Route* należy upewnić się, że powiązane z nim obiekty *Airport* i *Airline* istnieją już w bazie. W przeciwnym wypadku obiekt *Route* nie powinien zostać dodany.
- Tożsamość obiektu *Route* stanowi klucz biznesowy złożony z *airline + origin + destination*. Przed dodaniem takiego obiektu do bazy danych należy upewnić się, że obiekt z takimi samymi wartościami nie istnieje. W przeciwnym razie duplikat nie powinien zostać dodany.

II. PYTANIA

- 1) Kiedy warto zastosować listę powiązaną a kiedy tablicową [1] [2]?

LITERATURA

- [1] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition. [in Polish: Algorytmy]*. 2011. 1
- [2] James Ross Simon Harris. *Algorytmy. Od podstaw*. 2006. 1