

Praca domowa 09 – entity

Szlachetka Miłosz
Nr albumu. 114014

1.Cel zadania

W zadaniu należało nawiązać połączenie z bazą danych i pobrać z niej niezbędne informacje dotyczące ocen, a następnie wyznaczyć o ile % więcej/mniej ocena studenta z danego kursu jest większa/mniejsza od mediany ocen z tego samego kursu.

2.Struktura programu

class TblCourses implements Serializable - entity bean opatrzony adnotacjami JPA reprezentujący tabelę Tbl_Courses. Klasa posiada m.in. pola: id, courseName, tblStudentcourseCollection oraz metody dostępne do nich. Pole tblStudentcourseCollection służy do odnoszenia się do obiektów klasy TblStudentcourse. Klasa nadpisuje również metody hashCode(zwracającą identyfikator obiektu), equals(służącą do porównywania dwóch obiektów), toString(zwracającą String reprezentujący obiekt).

class TblStudents implements Serializable - entity bean opatrzony adnotacjami JPA reprezentujący tabelę Tbl_Students. Klasa posiada m.in. pola: id, firstName, lastName, tblStudentcourseCollection oraz metody dostępne do nich. Pole tblStudentcourseCollection służy do odnoszenia się do obiektów klasy TblStudentcourse. Klasa nadpisuje również metody hashCode(zwracającą identyfikator obiektu), equals(służącą do porównywania dwóch obiektów), toString(zwracającą String reprezentujący obiekt).

class TblStudentcourse implements Serializable - entity bean opatrzony adnotacjami JPA reprezentujący tabelę Tbl_StudentCourse. Klasa posiada m.in. pola: tblStudentcoursePK, mark, tblStudents, tblCourses oraz metody dostępne do nich. Pola tblStudents oraz tblCourses służą do odnoszenia się do obiektów klas TblStudents oraz TblCourses. Klasa **TblStudentcourse** nadpisuje również metody hashCode(zwracającą identyfikator obiektu), equals(służącą do porównywania dwóch obiektów), toString(zwracającą String reprezentujący obiekt).

class TblStudentcoursePK implements Serializable - klasa opatrzona adnotacjami JPA reprezentujący klucz podstawowy tabeli Tbl_StudentCourse. Klasa posiada pola: studentId, courseId (reprezentują klucze obce tabel Tbl_Courses, Tbl_Students) oraz metody dostępowe do nich. Klasa nadpisuje również metody hashCode(zwracającą identyfikator obiektu), equals(służącą do porównywania dwóch obiektów), toString(zwracającą String reprezentujący obiekt).

class AppMain - klasa, która implementuje algorytm wartości procentowej, która obrazuje czy ocena danego studenta z określonego kursu jest mniejsza/większa od mediany ocen z danego kursu i o ile. Metody tej klasy to:

void getDataFromFile(String filePath) - Odczytuje dane z pliku i zapisuje je w zmiennych statycznych klasy. Odczytanymi wartościami są nazwa kursu, imię oraz nazwisko studenta.

Integer getMark(String courseName, String firstName, String lastName) - Łącząc się z bazą danych odczytuje ocenę z kursu o nazwie courseName dla studenta o imieniu firstName oraz nazwisku lastName. Odczytana z bazy danych ocena jest następnie zwracana.

List<Integer> getAllMarks(String courseName) - Łącząc się z bazą danych odczytuje wszystkie oceny z kursu o nazwie courseName. Oceny są umieszczane w liście, a następnie zwracane.

double findMedian(List<Integer> marks) - dla zadanej listy ocen marks znajduje medianę i zwraca ją. Jeśli ilość ocen w liście marks jest parzysta to medianą jest średnia arytmetyczna dwóch środkowych elementów listy. Przy nieparzystej liście medianą jest środkowa liczba w liście.

double calculateResult(double median, double mark) - oblicza procentową wartość mówiącą o tym o ile procent większa/mniejsza jest ocena "mark" od mediany "median". W przypadku gdy mediana jest równa ocenie zwracana jest wartość 0.

void main(String[] args) - sprawdza czy do programu został podany argument będący ścieżką do pliku. Jeśli tak to wywołuje metodę **getDataFromFile** pobierając nazwę kursu oraz imię i nazwisko studenta. Jeśli komplet danych został pobrany poprawnie to pobiera ocenę z bazy danych określoną odczytanymi z pliku danymi (metoda **getMark**). Później pobierane są wszystkie oceny z danego kursu o nazwie pobranej z pliku. Następnie obliczana jest mediana wszystkich ocen z danego kursu i wyznaczana jest procentowa wartość mówiąca o tym o ile procent większa/mniejsza jest ocena danego studenta z danego kursu od mediany z tego kursu. Uzyskany wynik jest formatowany i wyprowadzany na wyjście jako wynik.

3. Łączenie się z bazą danych

Aby połączyć się z bazą danych utworzono fabrykę typu `EntityManagerFactory`, która udostępnia metodę `createEntityManager` tworzącą instancje klasy typu `EntityManager`. Aby utworzyć wcześniej wspomnianą fabrykę wykorzystano metodę: `Persistence.createEntityManagerFactory`, do której należało przekazać (jako argument) nazwę kontekstu (`Persistence Unit Name`). Każda instancja klasy `EntityManager` (utworzona przy użyciu fabryki `EntityManagerFactory`) reprezentuje połączenie z bazą danych. W programie, będącym przedmiotem tego zadania, obiekt klasy `EntityManager` był tworzony za każdym razem gdy zachodziła potrzeba interakcji z bazą. Liczba takich interakcji w tym przypadku wyniosła 2. Każda interakcja zawarta była w transakcji. Zapytania do bazy danych zostały napisane w języku JPQL. Tworzenie zapytania do bazy było wykonywane z użyciem metody `createQuery`. Do pobrania wyników zapytania użyto metody `getResultList`. Po zakończeniu operacji na bazie danych, połączenie z nią zostawało zwalniane (zamykane) aby bez potrzeby nie okupować połączenia.

4. Algorytm

Początkowo otwierany jest plik o podanej w argumencie programu ścieżce dostępowej, z którego odczytywane są kolejno: nazwa kursu, imię oraz nazwisko studenta. Jeśli dane zostały odczytane poprawnie to nawiązywane jest połączenie z bazą danych, a następnie pobierana jest z niej(z bazy) ocena z kursu o nazwie pobranej z pliku dla studenta zidentyfikowanego imieniem i nazwiskiem, również pobranym z pliku. Kolejnym krokiem jest pobranie wszystkich ocen dla kursu o tej samej nazwie co poprzednio. Oceny te są składowane w liście, a następnie sortowane w kolejności rosnącej. Po posortowaniu następuje proces wyznaczania mediany z posortowanej listy liczb(ocen). Jeśli lista zawiera nieparzystą ilość ocen to medianą jest ocena będąca środkowym elementem listy. Jeśli natomiast lista zawiera parzystą ilość elementów to medianą jest średnia arytmetyczna dwóch elementów znajdujących się w jej środkowej części. Mając ocenę studenta z konkretnego kursu oraz medianę dla tego samego kursu, wyliczana jest procentowa wartość mówiąca o tym o ile większa/mniejsza jest ocena danego studenta od wyznaczonej mediany. Jeśli ocena oraz mediana mają tą samą wartość to zwracany jest wynik 0. W innym przypadku wynik obliczany jest przy użyciu wzoru:

$$(\text{mark} - \text{median}) / \text{median} * 100.0;$$

Uzyskany rezultat jest następnie wyświetlany jako wartość procentowa.