Zaawansowane Techniki Programowania
Praca domowa 6
Dominik Freicher
05.12.2016

OPIS ZADANIA

Celem zadania jest implementacja aplikacji klienckiej AppClient, która odpowie na pytanie jaki procent zaewidencjonowanych w bazie klientów nie posiada ważnego ubezpieczenia na wskazany model samochodu oraz dzień (np. *Jaki procent klientów nie posiadało wykupionego ubezpieczenia samochodu marki 'Range Rover' w dniu '05/26/2014'*). Poprawnie zaokrąglony wynik należy wyznaczyć w procentach z dokładnością do jednego miejsca dziesiętnego.

OPIS MECHANIZMU WYSZUKIWANIA

Do zlokalizowania klasy DbManager oraz nawiązania z nią połączenia wykorzystany został interfejs JNDI (ang. Java Naming and Directory Interface). Pozwala on klientom na odkrywanie oraz wyszukiwania danych i obiektów na podstawie nazw.

Aby nawiązać połączenie należy utworzyć obiekt InitialContext:

```
InitialContext ctx = new InitialContext();
```

Następnie należy skorzystać z metody lookup, której implementacja umożliwia odnalezienie zasobu po nazwie:

```
IDbManager obj = (IDbManager) ctx.lookup("java:global/ejb-project/
DbManager!pl.jrj.db.IDbManager");
```

Ścieżka przestrzeni nazw ma postać:

```
java:global/ejb-project/DbManager!pl.jrj.db.IDbManager
```

ALGORYTM WYZNACZANIA WYNIKU

1. Pierwszym krokiem jest pobranie danych z pliku wejściowego tj. modelu samochodu oraz daty.
2. Za pomocą języka JPQL stworzone zostaje zapytanie zwracające liczbę wszystkich klientów z bazy danych na podstawie tabeli tbcustomer

```
SELECT count(c) FROM Customer c
```

3. Następnie z użyciem natywnego zapytania pobrana zostaje z bazy danych ilość klientów, którzy nie posiadają ubezpieczenia na dany model pojazdu w podanej dacie.

4. Końcowym krokiem jest obliczenie procentowego udziału klientów nie posiadających ubezpieczenia na dany model w podanej dacie za pomocą poniższego wzoru:

$$result = \frac{\mathrm{Kn}}{\mathrm{Kw}} * 100$$

Gdzie:
Kn – liczba klientów nieposiadająca ubezpieczenia na dany model w podanej dacie
Kw – liczba wszystkich klientów

STRUKTURY DANYCH

W skład projektu wchodzą klasy `AppClient.java` `IDbManager.java` `Insurance.java` `Customer.java` `Model.java`

```
AppClient.java
private static void registerWork()
private static void fetchingData(String fileName)
private static void calculateResult()
private static void printResult()
```

Główna klasa aplikacji klienckiej, w metodzie main jako argument przekazana zostaje ścieżka do pliku, następnie wywołana zostaje metoda registerWork, której zadaniem jest zarejestrowanie zadania z użyciem metody register komponentu DbManager. Następnie wywołana zostaje metoda fetichingData, która pobiera dane z pliku. Kolejnym krokiem jest wywołanie metody calculateResult, która odczytuje dane ze źródła danych oraz wykonuje niezbędne obliczenia. Metoda printResult odpowiedzialna jest za wyświetlenie rezultatu.

`IDbManager.java` – Interfejs zawierający deklaracje metody register
`boolean register(int hwork, String album)`

`Insurance.java` – Klasa zmapowana na tabele bazy danych tbinsurance
`Customer.java` – Klasa zmapowana na tabele bazy danych tbcustomer
`Model.java` – Klasa zmapowana na tabele bazy danych tbmodel

URUCHOMIENIE

Proces kompilacji jest możliwy z użyciem komendy:
`javac -cp <app-server-modules> -Xlint AppClient.java IDbManager.java *.java`

Uruchomienie programu możliwe jest z użyciem komendy:
`java -cp <app-server-modules> AppClient <file>`