

Projektowanie obiektowe 2024L

Etap 6 – Filtrowanie danych

Wstęp

W tym etapie zadaniem jest dodanie do programu mechanizmu pozwalającego na przeszukiwanie i filtrowanie danych.

System poleceń i kwerendy

Twoim zadaniem jest rozbudowa aplikacji o zestaw kolejnych poleceń:

`display` – wyświetla dane w postaci tabeli, składnia:

```
display {object_fields} from {object_class} [where {conditions}]
```

```
display * from Flights
```

```
display ID, TakeoffTime, WorldPosition from Flights where ID > 10 and ID < 20
```

`update` – aktualizuje dane, składnia:

```
update {object_class} set ({key_value_list}) [where {conditions}]
```

```
update Flights set (WorldPosition.Lat=54.5323, WorldPosition.Long=21.453)
```

```
where ID=34 or ID=50
```

`delete` – usuwa wybrane dane, składnia:

```
delete {object_class} [where {conditions}]
```

```
delete Flights where WorldPosition.Lat > 45.0 or WorldPosition.Long < -70.0
```

`add` – dodaje nowe dane, składnia:

```
add {object_class} new ({key_value_list})
```

```
add Flights new (ID=123, WorldPosition.Lat=21.0, WorldPosition.Long=50.0)
```

Gdzie:

`{object_fields}` – to nazwa klasy obiektu: Crew, Passenger, Cargo, CargoPlane, PassengerPlane, Airport lub Flight.

`{object_fields}` – to lista pól klasy rozdzielona przecinkami lub * dla wszystkich pól.

Flight: ID (uint), Origin (struct), Target (struct), TakeoffTime (DateTime), LandingTime (DateTime), WorldPosition (struct), AMSL (float), Plane (struct). Pola list Crews i Loads można pominąć.

Airport: ID (uint), Name (string), Code (string), WorldPosition (struct), AMSL (float), CountryCode (string)

PassengerPlane: ID (uint), Serial (string), CountryCode (string), Model (string), FirstClassSize (uint), BusinessClassSize(uint), EconomyClassSize(uint)

CargoPlane: ID (uint), Serial (string), CountryCode (string), Model (string), MaxLoad (float)

Cargo: ID (uint), Weight (float), Code (string), Description (string)

Passenger: ID (uint), Name (string), Age (uint), Phone (string), Email (string), Class (string), Miles (uint)

Crew: ID (uint), Name (string), Age (uint), Phone (string), Email (string), Practise (uint), Role (string)

WorldPosition: Lat (float), Long (float)

{conditions} – lista warunków rozdzielona operatorami **and** i **or**. Warunek składa się z nazwy pola klasy obiektu operatora (=, <=, >=, !=) oraz wartości. Bez możliwości grupowania nawiasami.

{key_value_list} – lista par klucz wartość rozdzielana przecinkami ze znakiem równa się pomiędzy nimi, gdzie klucz to nazwa pola danej klasy obiektu.

W przypadku struktur istnieje możliwość odwołania się do pól tej struktury.

Tabela w konsoli wyświetlana jest w postaci:

ID	TakeoffTime	WorldPosition
1	2024-04-22 12:39:23	{52.237049, 21.017532}
995	2024-04-22 11:50:00	{38.83, -6.12}
999	2024-04-22 02:03:00	{42.68, 119.1}
1000	2024-04-22 06:43:00	{5.62, 98.77}
1001	2024-04-22 18:01:00	{51.03, 56.71}
1004	2024-04-22 04:55:00	{-48.26, 114.05}

Nagłówki kolumny wyrównane są do lewej strony, wartości wyrównane są do prawej strony. Kolumny mają szerokość dostosowaną do najdłuższej wartości w kolumnie oraz zawsze mają dopełnienie (padding) 1 spacji od brzegu kolumny.

Całość funkcjonalności należy wykonać stosując metody projektowania obiektowego, tj. należy oddzielić od siebie parsowanie komend, tworzenie kwerend, wykonywanie kwerend oraz prezentowanie wyników w postaci tabeli.

Polecenia oraz funkcjonalności z poprzednich etapów muszą również nadal poprawnie działać.

Uwaga! Obowiązuje całkowity zakaz wykorzystywania mechanizmu refleksji.

Termin wykonania

3 tygodnie

Wszystkie pliki źródłowe muszą zostać wgrane na repozytorium git do 22.05.2024 23:59. Projekt należy zaprezentować prowadzącemu w czasie zajęć 23.05.2024.