

Object-Oriented Design 2024L

Stage 6 – Data Filtering

Introduction

In this stage, the task is to add a mechanism to the program that allows for searching and filtering data.

System of Commands and Queries

Your task is to expand the application with a set of additional commands:

`display` – displays data in a table format

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

```
display * from Flights
```

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

`update` – updates data

```
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` – removes selected data

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

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

`add` – adds new data

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

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

Where:

`{object_class}` – is the name of the object class: Crew, Passenger, Cargo, CargoPlane, PassengerPlane, Airport, or Flight.

`{object_fields}` – is a list of class fields separated by commas or * for all fields.

Flight: ID (uint), Origin (struct), Target (struct), TakeoffTime (DateTime), LandingTime (DateTime), WorldPosition (struct), AMSL (float), Plane (struct). The fields list Crews and Loads can be omitted.

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), Practice (uint), Role (string)

WorldPosition: Lat (float), Long (float)

{conditions} – list of conditions separated by operators '**and**' and '**or**'. A condition consists of the field name of the object class, an operator (=, <=, >=, !=), and a value. Grouping with parentheses is not required.

{key_value_list} – list of key-value pairs separated by commas with an equals sign between them, where the key is the name of the field of the given object class. For structures, it is possible to refer to the fields of that structure.

The table in the console is displayed in the format:

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}

Column headers are aligned to the left side, values are aligned to the right side. Columns are width-adjusted to the longest value in the column and always have a padding of 1 space from the column edge.

All functionalities must be implemented using object-oriented design methods, i.e., separate command parsing, query creation, query execution, and results presentation in table format.

Commands and functionality from previous stages must also continue to run correctly.

Note! The use of reflection mechanisms is strictly prohibited.

Deadline

3 weeks

All source files must be uploaded to the git repository by 22.05.2024 23:59. The project must be presented to the instructor during the class on 23.05.2024.