

# C# HQC - September 2017 - Travel Agency System

---

## General Description

---

A month ago you were given the task to implement a Travel Agency System (you remember, right? :)). Now you need to bring it to the next level! Using the knowledge you gained for writing high quality code you need to refactor the project following the SOLID principles, design patterns and best practices in general.

## Implement the following

---

- Refactor the **Engine** class using the SOLID principles, so that you can write unit tests for the class
- Extract the 3 collections from the **Engine** class to another class (i.e. **Database** class with **IDatabase** interface), so that the **Engine** is no more used as a database
- Refactor all **Command** classes using the SOLID principles, so that you can write unit tests for these classes
- Refactor the **CommandParser** class using the SOLID principles, so that you can write unit tests for the class
- Replace the Singleton "hand-written" implementation in **Engine** and **TravellerFactory** classes with one using an IoC container
- Write unit tests for **CreateTicketCommand** class
- Write unit tests for **CommandParser** class
- Add functionality for displaying in console "**The Engine is starting...**" before the Engine starts and "**The Engine worked for {number} milliseconds.**" after the Engine finished working - all without modifying the **Engine** and **Startup** classes (use **Decorator** design pattern, rather than simply calling `Console.WriteLine` in **Main** method :))

## Example

---

When you are done, use the input below to fully test your application.

### Input

```
createbus 10 0.7
createtrain 300 0.4 3
createairplane 250 1 true
createairplane 250 2.7 true
createtrain 80 0.4 3
listvehicles
createjourney Sofia vTurnovo 300 0
createjourney Sofia vTurnovo 3 0
createjourney vTurnovo Sofia 300 3
listjourneys
createticket 0 30
createticket 1 100
listtickets
exit
```

### Output

The Engine is starting...  
Vehicle with ID 0 was created.  
Specified argument was out of the range of valid values.  
Parameter name: A train cannot have less than 30 passengers or more than 150 passengers.  
Vehicle with ID 1 was created.  
Specified argument was out of the range of valid values.  
Parameter name: A vehicle with a price per kilometer lower than \$0.10 or higher than \$2.50 cannot exist!  
Vehicle with ID 2 was created.  
Bus ----  
Passenger capacity: 10  
Price per kilometer: 0.7  
Vehicle type: Land  
#####  
Airplane ----  
Passenger capacity: 250  
Price per kilometer: 1  
Vehicle type: Air  
Has free food: True  
#####  
Train ----  
Passenger capacity: 80  
Price per kilometer: 0.4  
Vehicle type: Land  
Carts amount: 3  
Journey with ID 0 was created.  
Specified argument was out of the range of valid values.  
Parameter name: The Distance cannot be less than 5 or more than 5000 kilometers.  
Failed to parse CreateJourney command parameters.  
Journey ----  
Start location: Sofia  
Destination: vTurnovo  
Distance: 300  
Vehicle type: Land  
Travel costs: 210.0  
Ticket with ID 0 was created.  
Failed to parse CreateTicket command parameters.  
Ticket ----  
Destination: vTurnovo  
Price: 240.0  
The Engine worked for 3152 milliseconds.