Abstract Factory

## Introduction

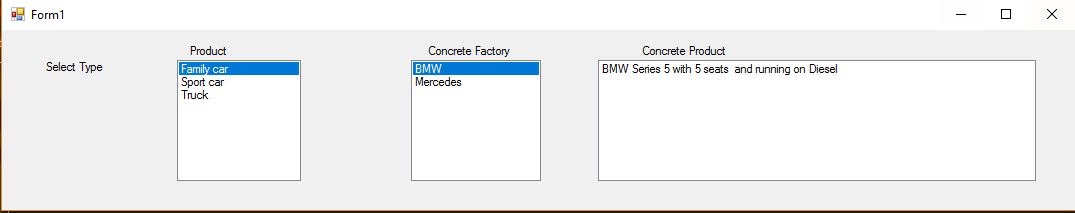
The main goal of the assignment was to create an application that would implement the “Abstract Factory” design pattern. The application is a car “dealership” where the user can select the type of the cat, the car maker and get the information about it.

## Implementation

The abstract factory design pattern provides an interface for creating families of related or dependent objects without specifying their concrete class. In our application we have three interfaces for car types: <<IFamilyCar>>, <<ISportCar>> and <<ITruck>>. Those interfaces are created in the main interface <<iFactory>> which represents a car factory that makes family cars, sport cars and trucks.

## User Interface

The following screenshot shows the user interface. By selecting the type and the maker of the car, the system will display the information about the vehicle.



* Reusability
  + The code is modular; it makes it easy to change the name of objects or properties.
* Extensibility
  + Adding new Factories or product is also easy as you just have to add classes.
* Maintainability
  + As we specified earlier, the code is modular so we can easily figure out what is going wrong in the app.

