Section: Master 1 SIGL

Faculty of Mathematics and Computer Science
Department of Computer Science

Object Analysis and Design (OAD)
Lab Session (TP) 04

Section 1. Creational Patterns

1. **Server Singleton:**
   We want to create an example of Singleton pattern, where the Singleton represents a Server with two attributes (Name and Ip address)

   Using the Singleton pattern, give a class diagram and a code implementation.

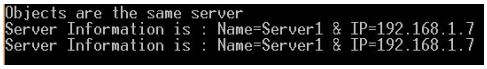


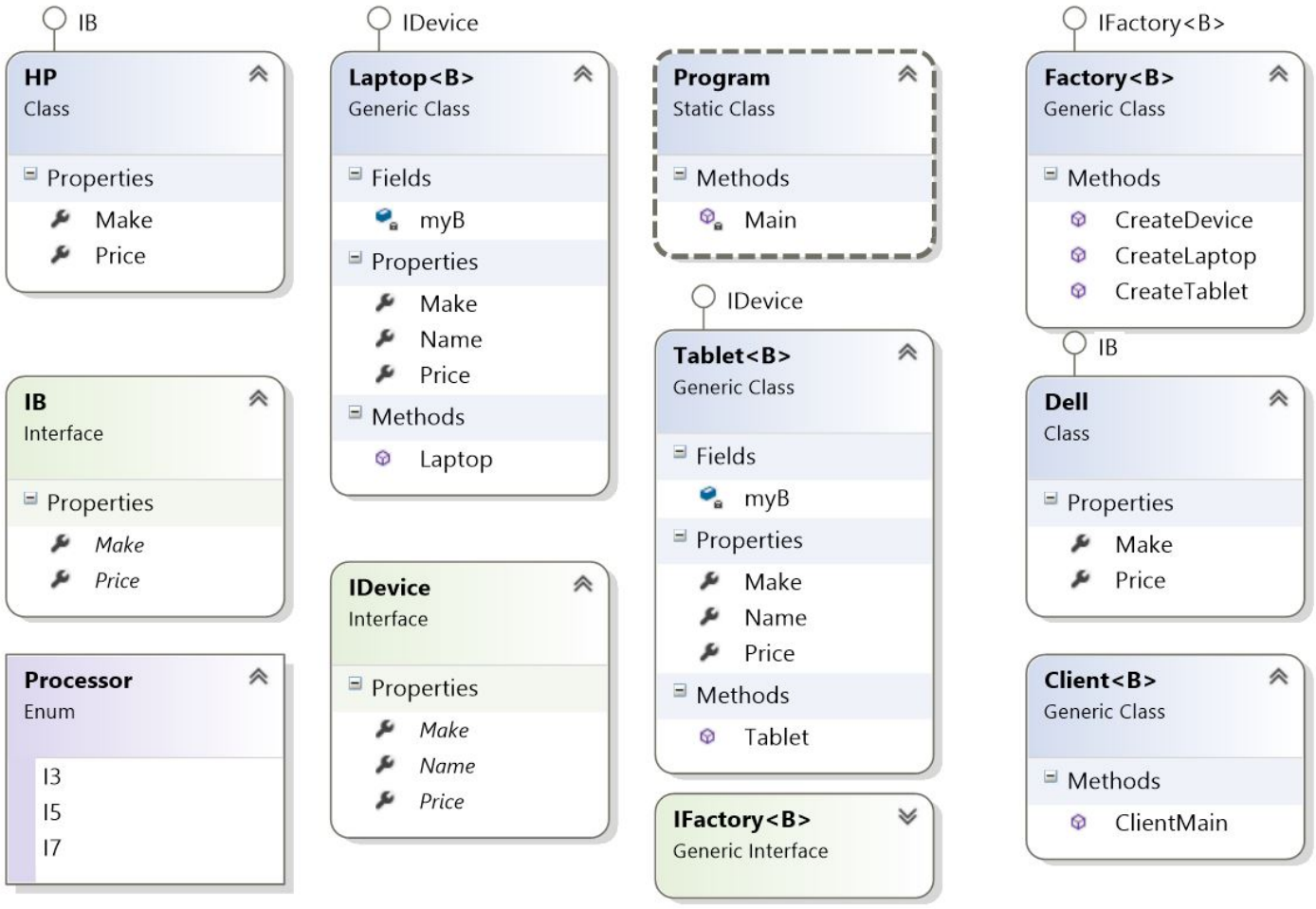
Figure 1: Singleton example output

```
Enter Technical Characteristics
CPU: I3
Ram: 2
I bought a device Tablete, of make Dell which cost 250
Enter Technical Characteristics
CPU: I5
Ram: 4
I bought a device Laptop, of make HP which cost 2000

These products have a price lower than 1000
Name:Tablete
Make:Dell
Price:250
```

2. Suppose that we have two factories of two brands *HP* and *DELL*. Both factories can create two types of devices *Tablet* and *Laptop*. Based on the characteristics specified by the customer, the factory either returns a laptop or tablet. If Ram$\leq$ 2 and Cpu =I3, then the factory would return tablet, If Ram$>$ 2 and CPU=I5 or CPU=I7, then the factory would return a laptop.

   Given the class diagram, give an implementation in C# for the output presented below using Abstract factory pattern and with the help of factory method pattern.

   Express in Lambda expression the query that returns a list of returned products that have a price lower than 1000.



3. **Vehicle Factory Method:**
We have an interface *IVehicle* that defines a function *Drive*. We have two concrete classes that implement *IVehicle*, *Bike*, *Scooter* and *DefaultVehicle*. The creation of such objects of these classes is made by the class *VehicleCreator*, through a function *CreateVeihicle*. *CreateVeihicle* returns the object based on the distance specified by the user, it returns Bike if distance<= 10, returns Scooter if 10<=distance<= 20 and returns DefaultVehicle otherwise .

Using the Factory Method pattern, give a class diagram and a code implementation of this problem.

```
Drive the Scooter : 10km
Drive the Bike : 20km
Vehicle can neither be created nore driven
```

Figure 2: Factory Method example output

Answer Key for Exam A

Section 1. Creational Patterns

1. **Server Singleton:**
   We want to create an example of Singleton pattern, where the Singleton represents a Server with two attributes (Name and Ip address)

   Using the Singleton pattern, give a class diagram and a code implementation.

Figure 3: Singleton example output

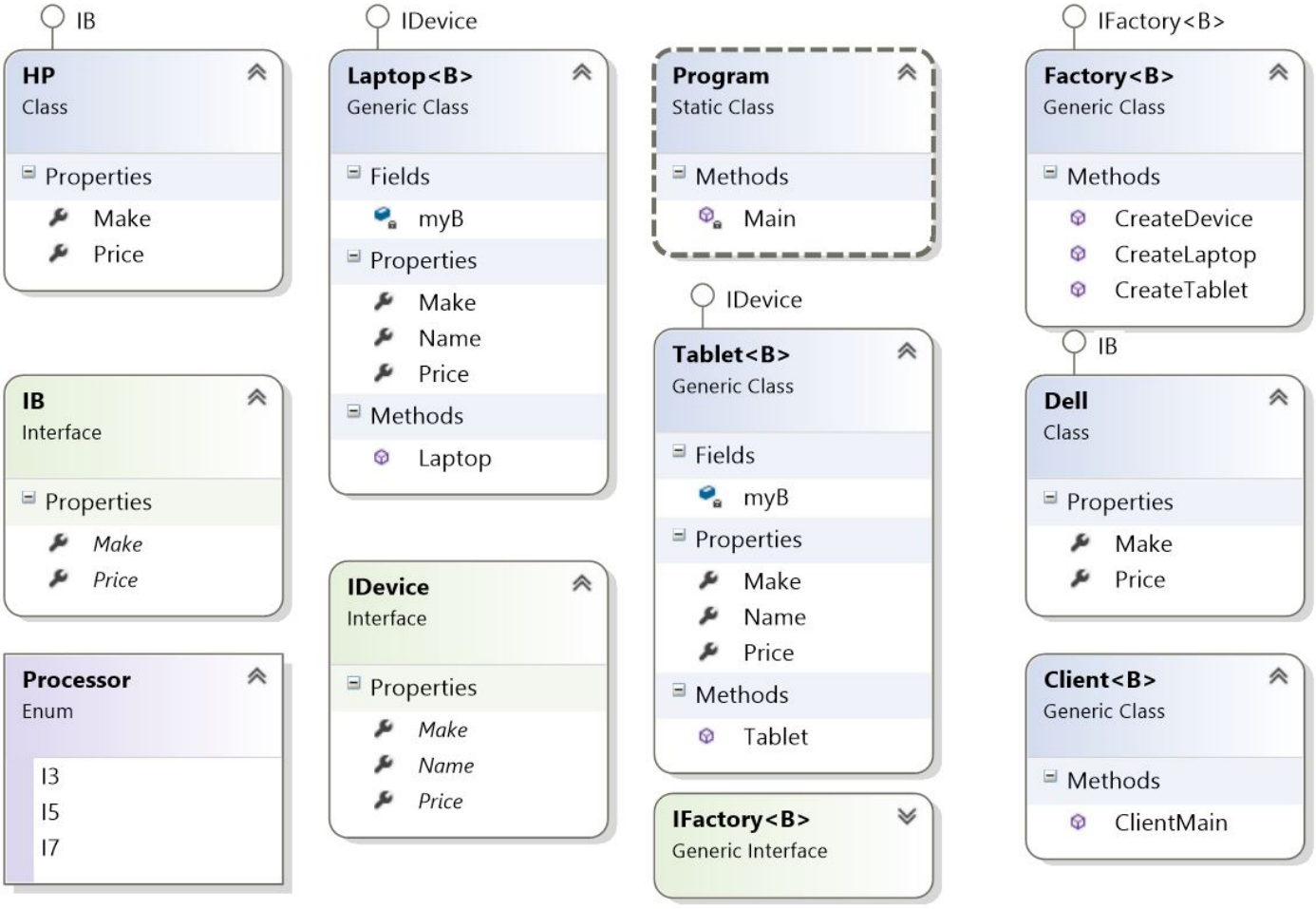
```
Enter Technical Characteristics
CPU: I3
Ram: 2
I bought a device Tablete, of make Dell which cost 250
Enter Technical Characteristics
CPU: I5
Ram: 4
I bought a device Laptop, of make HP which cost 2000

These products have a price lower than 1000
Name:Tablete
Make:Dell
Price:250
```

2. Suppose that we have two factories of two brands *HP* and *DELL*. Both factories can create two types of devices *Tablet* and *Laptop*. Based on the characteristics specified by the customer, the factory either returns a laptop or tablet. If Ram$\leq$ 2 and Cpu =I3, then the factory would return tablet, If Ram$>$ 2 and CPU=I5 or CPU=I7, then the factory would return a laptop.

   Given the class diagram, give an implementation in C# for the output presented below using Abstract factory pattern and with the help of factory method pattern.

   Express in Lambda expression the query that returns a list of returned products that have a price lower than 1000.

3. **Vehicle Factory Method:**
   We have an interface *IVehicle* that defines a function *Drive*. We have two concrete classes that implement *IVehicle*, *Bike*, *Scooter* and *DefaultVehicle*. The creation of such objects of these classes is



made by the class *VehicleCreator*, through a function *CreateVeihicle*. *CreateVeihicle* returns the object based on the distance specified by the user, it returns Bike if distance<= 10, returns Scooter if 10<=distance<= 20 and returns DefaultVehicle otherwise .

Using the Factory Method pattern, give a class diagram and a code implementation of this problem.

```
Drive the Scooter : 10km
Drive the Bike : 20km
Vehicle can neither be created nore driven
```

Figure 4: Factory Method example output