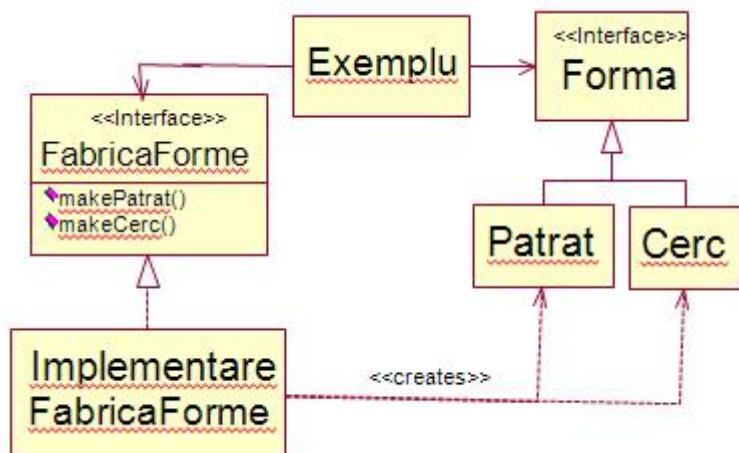


## Laborator 6

1. Fie fabrica de forme prezentata in curs:



```

public interface FabricaForme {
    public Forma make (String FormaName) throws Exception
}
public class ImplementareFabricaForme implements FabricaForme {

    public Forma make(String FormaName) throws Exception {
        if (FormaName.equals("Cerc")) return new Cerc();
        else if (FormaName.equals("Patrat")) return new Patrat();
        else throw new Exception("FabricaForme cannot create " + FormaName);
    }
}

.....
private FabricaForme Fabrica;
Fabrica = new ImplementareFabricaForme();
Forma s = Fabrica.make("Cerc");
    
```

**tema 1.** Pornind de la aceasta sa se construiasca doua fabrici de obiecte (50 min)

- fabrica de nave spatiale una cub, una piramidala si una sferica care sa fie desenate pe canvas. Navele sa primeasca si un parametru pentru culoarea de desinare
- o fabrica de "foc" in care sa se obtina trei tipuri de foc pentru navele antarioare linear – mitraliera, blast – o minge de foc si rachete - mai multi clindri.

2. API-ul Java oferă două interfețe pentru a ușura implementarea acestui șablon: Obsever și Observable. Exemplul următor arată cum pot fi folosite aceste interfețe. Pornind de la acesta sa se implementeze un observer general pentru navele spatiale anterioare care sa permita ca fiecare nava sa primeasca informatii complete despre celelalte nave (viata, foc, directie foc, tip foc, putere nava, clasa navei etc) (50 minute)

```

import java.util.Observable;

public class MessageBoard extends Observable
{
    private String message;
    public String getMessage() { return message;}
    public void changeMessage(String message)
    {
        this.message = message;
        setChanged();
        notifyObservers(message);
    }
    public static void main(String[] args)
    {
        MessageBoard board = new MessageBoard();
        Student bob = new Student();
        Student joe = new Student();
        board.addObserver(bob);
        board.addObserver(joe);
        board.changeMessage("More Homework!");
    }
}

```

```

import java.util.Observable;
import java.util.Observer;

class Student implements Observer
{
    public void update(Observable o, Object arg)
    {
        System.out.println("Message board changed: " + arg);
    }
}

```

**Tema acasa:** Pornind de la exemplul din curs sa se creeze un adapter pentru fiecare din tipurile de functii pentru foc ale navelor astfel incat oricare nava sa poata sa foloseasca oricare tip de foc (cand face upgrade de exemplu)