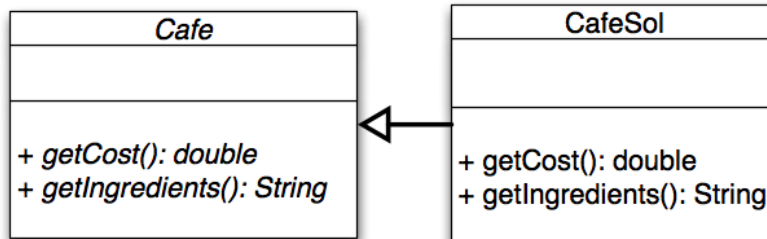


**Examen PROP**  
**Quadrimestre Tardor**  
**Curs 2014-15**

**Problema 1 (7 punts):**

Tenim les següents classes:



on **Cafe** és una classe abstracta amb un parell de mètodes abstractes (**getCost()** i **getIngredients()**). Volem ampliar aquest domini amb més classes (per exemple, les classes **Llet**, **Crema**, **Xocolata** i d'altres) de manera que sigui senzill afegir ingredients al café. La intenció és fer servir el sistema de manera que un programa com aquest:

```
public class Main {
    public static void main(String[] args) {
        Cafe c = new CafeSol();
        System.out.println("Cost: " + c.getCost() +
                           "; Ingredients: " + c.getIngredients());

        c = new Llet(c);
        System.out.println("Cost: " + c.getCost() +
                           "; Ingredients: " + c.getIngredients());

        c = new Xocolata(c);
        System.out.println("Cost: " + c.getCost() +
                           "; Ingredients: " + c.getIngredients());

        c = new Crema(c);
        System.out.println("Cost: " + c.getCost() +
                           "; Ingredients: " + c.getIngredients());

        c = new Xocolata(c);
        System.out.println("Cost: " + c.getCost() +
                           "; Ingredients: " + c.getIngredients());
    }
}
```

hauria de generar la següent sortida:

```
Cost: 1.0; Ingredients: Cafe
Cost: 1.5; Ingredients: Cafe, Llet
Cost: 1.7; Ingredients: Cafe, Llet, Xocolata
Cost: 2.4; Ingredients: Cafe, Llet, Xocolata, Crema
Cost: 2.6; Ingredients: Cafe, Llet, Xocolata, Crema, Xocolata
```

Es valorarà molt positivament que el codi es reutilitzi al màxim.

1.1 (1 punt).- Implementa les classes **Cafe** i **CafeSol**.

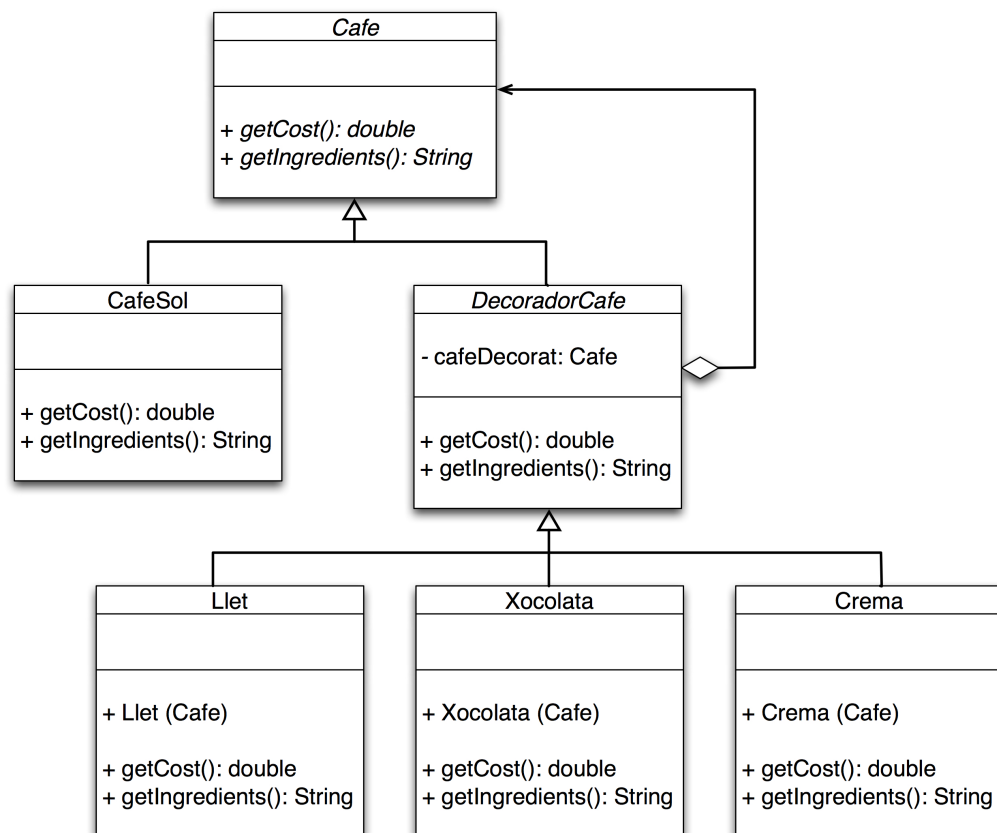
Solució:

```
abstract class Cafe {  
    abstract public double getCost();  
    abstract public String getIngredients();  
}
```

```
class CafeSol extends Cafe {  
    public double getCost() {  
        return 1;  
    }  
  
    public String getIngredients() {  
        return "Cafe";  
    }  
}
```

1.2 (3 punts).- Afegeix al diagrama les classes **Llet**, **Crema** i **Xocolata**. Fes-ho pensant que la redundància en el codi ha de ser mínima. Si penses que fan falta, fes servir classes auxiliars. Cal ser precís en l'especificació de les relacions entre les classes (poseu-les totes).

Solució:



1.3 (3 punts).- Implementa les classes afegides, és a dir, les classes **Llet**, **Crema** i **Xocolata** i les classes auxiliars que hagi utilitzat.

Solució:

```
abstract class DecoradorCafe extends Cafe {
    private Cafe cafeDecorat;

    public DecoradorCafe(Cafe cafeDecorat) {
        this.cafeDecorat = cafeDecorat;
    }

    public double getCost() {
        return cafeDecorat.getCost();
    }

    public String getIngredients() {
        return cafeDecorat.getIngredients();
    }
}

class Llet extends DecoradorCafe {
    public Llet(Cafe cafeDecorat) {
        super(cafeDecorat);
    }

    public double getCost() {
        return super.getCost() + 0.5;
    }

    public String getIngredients() {
        return super.getIngredients() + ", Llet";
    }
}
```

i **Crema** i **Xocolata** són iguals que **Llet**.

Cal tenir en compte algunes coses amb les que podríem ser flexibles:

- La classe **DecoradorCafe** no cal que sigui abstracta. Altra cop, és convenient (ja que així no podem instanciar-la) però no imprescindible.

## Problema 2 (3 punts):

Digueu si cadascuna de les següents afirmacions és certa o falsa. NO cal justificar la resposta. (*Cada resposta correcta suma 0,5 punts i cada resposta incorrecta resta 0,5 punts*).

1) Podem instanciar les classes-paràmetre de les classes genèriques dins d'aquestes.

Fals

2) Una classe abstracta no es pot instanciar

Cert

3) Cal que una classe abstracta tingui un o més mètodes amb el qualificador **abstract**

Fals

4) Una instància d'una classe és també instància de les seves superclasses.

Cert

5) Una instància d'una classe és també instància de les seves subclasses.

Fals

6) En Java s'hereda tot, no és possible eliminar o esborrar ni atributs ni mètodes.

Cert