TD collectif patron de conception

Exercice 1

On souhaite développer une petite application qui mesure la **température** d'une pièce.
La température est gérée par un objet **Capteur**.
Chaque fois que la température change, plusieurs affichages doivent être mis à jour automatiquement :

1. **AfficheurNumérique** : affiche la valeur brute (ex : "Température = 22°C")
2. **AfficheurAlerte** : affiche " Chaud" si > 30°C, sinon "Normal".

On vous demande :

1. Identifier les **rôles** Subject (Observable) et Observers.
2. Dessiner un **petit diagramme de classes** du patron Observer.
3. Donner un **code Java simple** montrant l'utilisation du patron Observer.

Exercice 2

On veut afficher du texte, mais parfois on veut le **mettre en majuscules** ou **l'entourer d'étoiles**.
On doit pouvoir combiner ces décorateurs **sans modifier la classe de base**.
On vous demande :

1. Créer une classe abstraite Texte avec afficher().
2. Créer une classe concrète TexteSimple.
3. Créer deux décorateurs : Majuscules et Etoiles.
4. Faire le diagramme de classe correspondant.

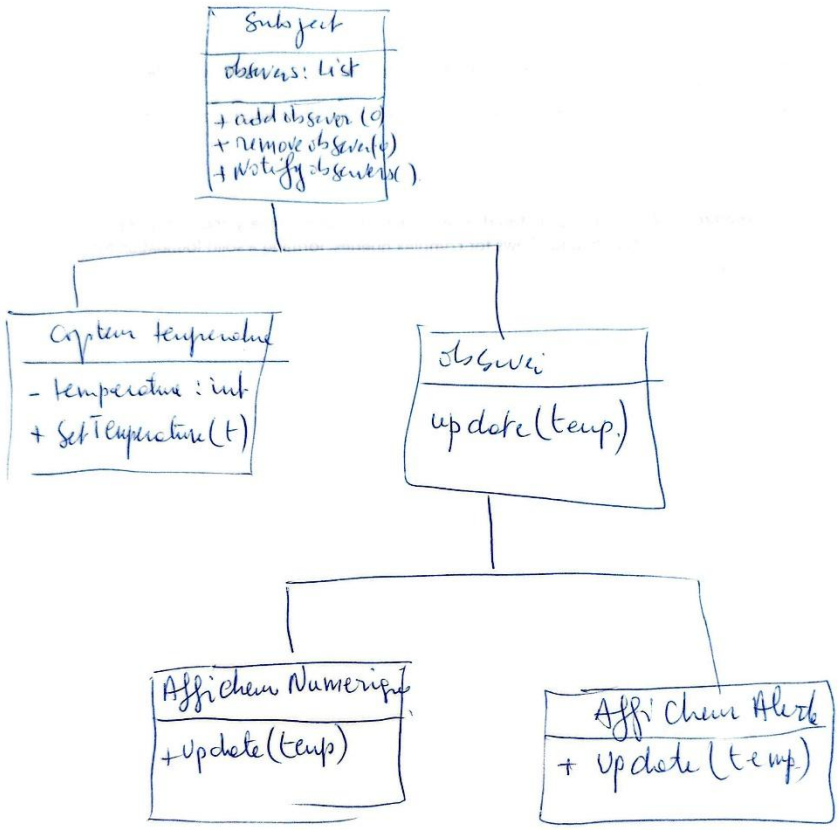
Solution

Exercice 1

1. Rôles du patron

Rôle	Classe dans l'exercice
Subject	CapteurTemperature
Observer	AfficheurNumerique et AfficheurAlerte

3. Diagramme de classes



3. Code Java corrigé

Interface Observer

```
public interface Observer {
   void update(int temp);
}
```

Classe Subject (Observable)

```
import java.util.ArrayList;
import java.util.List;

public class CapteurTemperature {

   private List<Observer> observers = new ArrayList<>();
   private int temperature;

   public void addObserver(Observer o) {
      observers.add(o);
   }

   public void removeObserver(Observer o) {
      observers.remove(o);
   }

   public void setTemperature(int t) {
      this.temperature = t;
      notifyObservers();
   }

   private void notifyObservers() {
      for (Observer o : observers) {
         o.update(temperature);
      }
   }
}
```

Observer 1 : Afficheur numérique

```
public class AfficheurNumerique implements Observer {

   @Override
   public void update(int temp) {
      System.out.println("Afficheur numérique : Température = " + temp + "°C");
   }
}
```

Observer 2 : Afficheur alerte

```
public class AfficheurAlerte implements Observer {

   @Override
   public void update(int temp) {
      if (temp > 30)
         System.out.println("Afficheur alerte : ⚠ Chaud !");
      else
         System.out.println("Afficheur alerte : Normal");
   }
}
```

Programme principal

```
public class Main {
   public static void main(String[] args) {

      CapteurTemperature capteur = new CapteurTemperature();

      Observer num = new AfficheurNumerique();
      Observer alerte = new AfficheurAlerte();

      capteur.addObserver(num);
      capteur.addObserver(alerte);

      System.out.println("---- Température = 25 ----");
      capteur.setTemperature(25);

      System.out.println("---- Température = 35 ----");
      capteur.setTemperature(35);
   }
}
```

Solution exercice 2

1. **Classe de base abstraite**

```
public abstract class Texte {
   public abstract String afficher();
}
```

2. **Classe concrète : TexteSimple**

```
public class TexteSimple extends Texte {

   private String contenu;
```

```
    public TexteSimple(String contenu) {
        this.contenu = contenu;
    }

    @Override
    public String afficher() {
        return contenu;
    }
}
```

3. Décorateur abstrait

```
public abstract class TexteDecorator extends Texte {
    protected Texte texte;
}
```

Décorateur Majuscules

```
public class Majuscules extends TexteDecorator {

    public Majuscules(Texte t) {
        this.texte = t;
    }

    @Override
    public String afficher() {
        return texte.afficher().toUpperCase();
    }
}
```

Décorateur Etoiles

```
public class Etoiles extends TexteDecorator {

    public Etoiles(Texte t) {
        this.texte = t;
    }

    @Override
    public String afficher() {
        return "*** " + texte.afficher() + " ***";
    }
}
```

Programme principal

```
public class Main {
   public static void main(String[] args) {

      Texte t1 = new TexteSimple("Bonjour");
      System.out.println(t1.afficher());  // Bonjour

      Texte t2 = new Majuscules(new TexteSimple("Bonjour"));
      System.out.println(t2.afficher());  // BONJOUR

      Texte t3 = new Etoiles(new TexteSimple("Bonjour"));
      System.out.println(t3.afficher());  // *** Bonjour ***

      Texte t4 = new Etoiles(new Majuscules(new TexteSimple("Bonjour")));
      System.out.println(t4.afficher());  // *** BONJOUR ***
   }
}
```

4. Diagramme de classes