Ajout d'accessoires (aux `Montre`s)

Rappel :

```
class Montre extends Produit {
 private Mecanisme coeur;
 private ArrayList<Accessoire> accessoires;
}
```

On souhaite ajouter des accessoires .

Par exemple :

```
montre.ajouter(new Bracelet(...));
```

Ajout d'accessoires (aux `Montre`s)

```
class Montre extends Produit {

   private Mecanisme coeur;
   private ArrayList<Accessoire> accessoires;

   public void ajouter(Accessoire accessoire) {
      accessoires.add(accessoire); // nous reviendrons sur ce point
   }
}
```

Finalisation d'une première version

Essayons maintenant d'avoir une première version opérationnelle de notre code, pour le moment :

- *sans* tous les mécanismes
- *sans* copie des montres

Pour cela, il nous faut encore :

- quelques *accessoires*
- terminer la classe `Montre`
- un exemple d'utilisation dans une méthode `main()`

Quelques accessoires

Décidons par exemple que les accessoires :

- ont un nom et une valeur de base fixés au départ (sans valeur par défaut)

```
abstract class Accessoire extends Produit {
   private final String nom;

   public Accessoire(String unNom,
                     double valeurDeBase) {
      super(valeurDeBase);
      nom = unNom;
   }
}
```

Quelques accessoires

Décidons par exemple que
les accessoires :

- s'affichent en indiquant leur nom et leur prix

```
abstract class Accessoire extends Produit {
   private final String nom;

   public Accessoire(String unNom,
                     double valeurDeBase) {
      super(valeurDeBase);
      nom = unNom;
   }

   @Override
   public String toString() {
      String result = nom + " coûtant ";
      result += super.toString();
      return result;
   }
}
```

Quelques accessoires

Décidons par exemple que
les accessoires :

- que leur prix est celui d'un produit usuel

Quelques accessoires

```
class Bracelet extends Accessoire {

   public Bracelet(String unNom, double valeurDeBase) {
      super("bracelet " + unNom, valeurDeBase);
   }
}

//-------------------------------------------------------

class Fermoir extends Accessoire {
   public Fermoir(String unNom, double valeurDeBase) {
      super("fermoir " + unNom, valeurDeBase);
   }
}

//...
```

Finalisation des `Montre`s

Pour finaliser la classe `Montre`
(mais sans mécanisme) :

- contentons nous pour l'heure d'un constructeur par défaut

```
class Montre extends Produit {

  private Mecanisme coeur;
  private ArrayList<Accessoire> accessoires;

  public Montre() {
    accessoires = new ArrayList<Accessoire>();
  }
  // ...
}
```

Finalisation des Montres

Pour finaliser la classe Montre
(mais sans mécanisme) :

- décidons d'un calcul de prix :
  somme des prix des accessoires

```
// ...
   @Override
   public double prix() {
      // Au départ, le prix est le prix de base
      double prixFinal = super.prix();

      for (Accessoire acc : accessoires) {
         prixFinal += acc.prix();
      }
      return prixFinal;
   }
   //..
}
```

Finalisation des Montres

Pour finaliser la classe Montre
(mais sans mécanisme) :

- décidons d'un affichage :

```
Une montre composée de :
  * bracelet cuir coûtant 54
  * fermoir acier coûtant 12.5
  * etc.
==> Prix total : 147.9
```

```
//...
public void afficher () {
   System.out.print  ("Une montre ");
   System.out.println("composée de :");

   for (Accessoire acc : accessoires) {
     System.out.println("  * " + acc);
   }
   System.out.print  ("==> Prix total : ");
   System.out.println(prix());
}
//...
```

Un exemple de main()

```
public static void main(String[] args)
{
  Montre m = new Montre();

  m.ajouter(new Bracelet("cuir"  , 54.0));
  m.ajouter(new Fermoir ("acier" , 12.5));
  m.ajouter(new Boitier ("acier" , 36.6));
  m.ajouter(new Vitre   ("quartz", 44.8));

  System.out.println('\n' +
                     "Montre m :");

  m.afficher();
}
```

```
Montre m :
Une montre composée de :
  * bracelet cuir coutant 54
  * fermoir acier coutant 12.5
  * boitier acier coutant 36.6
  * vitre quartz coutant 44.8
==> Prix total : 147.9
```

Le code complet à ce stade (154 lignes) peut être téléchargé sur le site du cours :

`https://d396qusza40orc.cloudfront.net/java-fr/complements/Montres01.java`