

Langages de programmation

TP9

Objectifs

- Classes et objets (suite)
- La relation entre les classes

Théorie

L'association

- C'est une simple relation d'utilisation

```
class Adresse {  
    private String rue;  
    private int nombre;  
}
```

```
class Compte {  
    private int numero;  
    private float solde;  
}
```

```
class Personne {  
    private String nom;  
    private String prenom;  
    private Adresse adresse;  
    //Si la personne peut avoir plusieurs comptes  
    private ArrayList< Compte > comptes = new ArrayList<>();  
}
```

Problèmes en classe

1. Créez un projet avec le nom LivreAuteur. Vous avez la classe LivreAuteur avec la méthode “main”.

a) Mettez en œuvre la classe Auteur, dans le projet LivreAuteur. Un auteur est caractérisé par:

- Code unique (private, integer)
- Nom de famille (private, string)
- Prénom (private, string)
- Pays de naissance (private, string)

Décidez les plus appropriés constructeurs et méthodes pour la classe.

Problèmes en classe

b) Chaque auteur peut écrire jusqu'à 10 livres. Le livre se caractérise par:

- Titre (private, string)
- Prix de vente (private, double)
- Année de publication (private, integer)
- Nombre de pages (private, integer)
- ISBN (private, string)
- Domaine (string; par exemple, "les mathématiques")

Mettez en œuvre la classe Livre, dans le projet LivreAuteur.

Problèmes en classe

c) Mettez en œuvre les méthodes suivantes dans la classe Auteur:

- Ajoutez un nouveau livre
- Affichez les livres
- Modifiez le domaine d'un livre avec un ISBN donné
- Trouvez le livre avec le plus grand nombre de pages
- Calculez le prix moyen des livres
- Calculez le prix moyen des livres publiés au cours des deux dernières années

// année en cours

```
int annee = new GregorianCalendar().get(GregorianCalendar.YEAR);
```

d) Dans la méthode “main” de la classe LivreAuteur, déclarez 5 livres et 2 auteurs, chacun avec 3 livres écrits et testez les méthodes mises en œuvre.

Devoir

1. Écrivez une classe `Item`. Un article est identifié par un nom sous forme de chaîne de caractères, un prix (flottant) et enfin un identifiant unique (entier positive ≥ 1). On peut également donner une description à l'objet mais c'est optionnel. Le code suivant vous montre la création de deux articles :

```
public static void main ( String [] args) {  
    Item i1 = new Item ( "Vodka" , "Une bouteille de Vodka" , 5483918746738L, 75.5);  
    Item i2 = new Item ( "Filet de citron 200g" , 5473664615361L, 3.25);  
}
```

Ajoutez également deux méthodes `getPrice` a la classe :

- `public double getPrice();`
- `public double getPrice (int q);`

La première méthode renvoie simplement le prix de l'article, tandis que la seconde renvoie le prix pour `q` fois l'article en tenant compte du fait que si on commande plus de 5 fois un article, on reçoit 5% de réduction sur le prix total.

Ecrivez aussi une classe `TestItem` afin de tester la classe `Item`. Attention : deux items ne peuvent avoir le même identifiant!

Devoir

2. Mettez en œuvre la classe **Produit**. Un produit a:

- Code unique (integer)
- Prix (double)
- La date de production (GregorianCalendar "jj / mm / aa")
- Période de validité - en jours (integer)
- Catégorie (string; EX.: «aliment»)

Mettez en œuvre les constructeurs et les méthodes nécessaires. Chaque **boutique** peut vendre jusqu'à 50 produits. Une boutique contient une gamme de produits et les méthodes suivantes:

- Ajoutez un nouveau produit
- Supprimez un produit en utilisant son index dans le tableau
- Modifiez le prix d'un produit, étant donné le code unique
- Affichez tous les produits à partir d'une certaine catégorie
- Affichez tous les produits qui sont périmés
- Affichez le produit le moins cher
- Calculez la moyenne des prix des produits fabriqués au cours des trois dernières années

Devoir

Créez une classe de test et de déclarez 2 boutiques.

Ajoutez 6 produits de la même catégorie pour la première boutique: 3 produits sont périmés et 3 sont valables. Ajoutez encore 2 produits des différentes catégories pour la première boutique. Testez ensuite les méthodes.

```
Livre l1 = new Livre();
```

```
l1.setDatePublication(new GregorianCalendar(2012,12,29));
```

Pour afficher dans un format agréable:

```
SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yy");
```

```
String dp = formatter.format(datePublication.getTime());
```

Note: Si vous ne réussissez pas à travailler avec **GregorianCalendar**, vous pouvez mettre la date de production seulement en années