

# NFA032 : TP 4 (Objets partie 2 : création d'une classe, exceptions et méthodes statiques)

14 mars 2018

## 1 Préliminaires

Lancez Eclipse et créez un nouveau projet de nom `Tp4-032`. Vous aurez besoin de la classe `Compte` du Tp3 (elle est donnée ci-dessous) et la classe `Livre` que vous devrez écrire.

## 2 Exercice 1 : méthodes statiques et non statiques dans la classe `Compte`

Allez dans la classe `Compte` et réalisez les modifications suivantes :

1. Ajoutez le mot-clé `static` dans l'entête de la méthode `getSolde()`. Que se passe-t-il ? Expliquez la raison de ce problème et revenez à l'état du fichier avant cette modification.
2. Dans cette classe, ajoutez la méthode suivante. Que fait-elle ? Notez qu'il s'agit d'une méthode statique.

---

```
public static void afficheMessageSecurite() {  
    System.out.println("Ne donnez jamais vos codes secrets!");  
}
```

---

3. Modifiez la méthode `affiche` de votre classe comme ceci. S'agit-il d'une méthode statique ? Justifiez votre réponse. Comment se comportera cette méthode désormais ? Expliquez la syntaxe de l'appel au point 1.

---

```
public void affiche(){  
    afficheMessageSecurite(); // 1  
    System.out.println("Numero:_" + numero);  
    System.out.println("Solde:_" + solde);  
    System.out.println();  
}
```

---

4. Ajoutez dans votre paquetage la classe suivante pour tester les comptes et exécutez ce programme. Expliquez la syntaxe de l'appel fait au point 2.

---

```
public class TestMethStatiques {  
    public static void main(String[] args) {  
        // Appel à la méthode qui affiche les consignes de sécurité  
        // Notez que cet appel est fait sans qu'il existe aucun objet Compte.  
        Compte.afficheMessageSecurite(); // 2  
    }  
}
```

```

    Compte c1 = new Compte(1, 100.00);
    c1.affiche();
    // affiche(); // 3
}
}

```

---

5. Enlevez les commentaires devant la ligne au point 3 ci-dessus : une erreur se produit. Expliquez pourquoi.
6. Enlevez le mot-clé `static` dans l'entête de la méthode `afficheMessageSecurite()` : une erreur se produit. Dans quel fichier ? Expliquez pourquoi.
7. Ajoutez cette nouvelle méthode statique dans la classe `Compte`. Elle lit au clavier les données d'un nouveau compte à créer, puis renvoie en résultat l'objet compte correspondant. (a) Expliquez pourquoi cette méthode est statique (en d'autres termes, pourquoi ce code compile) ; (b) ajoutez dans le main donné plus haut un appel à cette méthode destiné à créer un deuxième compte à mettre dans une variable `c2`. Testez votre programme.

```

public static Compte SaisirCompte() {
    Terminal.ecrireString("Numero_du_nouveau_compte?_");
    int n = Terminal.lireInt();
    Terminal.ecrireString("Solde_initial?_");
    double i = Terminal.lireDouble();
    return new Compte(n,i);
}

```

---

### 3 Exercice 2

Dans cette exercice on abordera (sur plusieurs séances) les objets nécessaires pour une application qui gère le stock de livres en vente dans une librairie. Dans votre projet Tp4, créez un nouveau paquetage nommé *livres*. Dedans ajoutez une classe nommée `Livre` pour modéliser les livres de la librairie et une classe `TesteLivres` en respectant les consignes données plus bas.

1. Un livre possède un titre, un auteur, un prix (double) et un nombre d'exemplaires (int) en stock. Toutes les variables d'instance devront être `private`. *Le prix et le nombre d'exemplaires d'un livre ne devront jamais être négatifs*. Cette propriété de *cohérence de l'état interne* doit être garantie à tout moment pour un `Livre` : le constructeur ne doit créer d'objets qui violent cette propriété et toute modification de l'état interne doit maintenir cette propriété.
2. Ajoutez un constructeur pour initialiser toutes les variables. Votre constructeur doit lever `IllegalArgumentException` en cas de valeurs incohérentes pour l'initialisation (voir point précédent).
3. Ajoutez des méthodes accesseurs et une méthode pour afficher les données d'un livre.
4. Ajoutez une méthode pour modifier le prix d'un livre. *Attention à ne pas rendre l'état interne incohérent*.
5. Ajoutez et complétez le code pour ces deux méthodes, dont on vous donne les entêtes et une description du comportement attendu. Vous devrez déclarer l'exceptions `NbExemplairesInsuffisant` dérivée de `RuntimeException`. Déclarez-là dans un nouveau fichier.

---

```

1  /**
2   * Ajoute n au nombre d'exemplaires de ce livre.
3   * @throws IllegalArgumentException si n est négatif.
4   */
5   public void ajoutExemplaires(int n) {
6       // completer
7   }
8   /**
9   * Substrait n au nombre d'exemplaires de ce livre si le nombre courant
10  * est supérieur ou égal a n et retourne le prix a payer pour n exemplaires.
11  * @throws IllegalArgumentException si n est négatif.
12  * @throws NbExemplairesInsuffisant si le nombre courant est plus petit que n.
13  */
14  public double retireExemplaires(int n) {
15      // completer
16  }

```

---

6. Ajoutez dans ce paquetage une classe `TesteLivres` avec le code suivant. Assurez vous qu'il compile avant de rendre votre devoir !

---

```

1  public class TestLivres {
2      public static void main(String[] args) {
3          Livre b1 = new Livre("Richard_Russo", "Le_déclin_de_l'empire_Whiting", 18.5, 3 );
4          b1.affiche();
5          String a = b1.getAuteur();
6          b1.ajoutExemplaires(4);
7          b1.affiche();
8          System.out.println(b1.retireExemplaires(2));
9          b1.affiche();
10         System.out.println(b1.retireExemplaires(10));
11         b1.affiche();
12     }
13 }

```

---

7. Ajoutez du code dans ce main permettant de déclarer un tableau de 5 livres que vous devez initialiser en interrogeant l'utilisateur. Pour ce faire vous devez au préalable ajouter dans la classe `Livre` une méthode statique de même esprit que `saisirLivre()` de l'exercice 1. Vous l'utiliserez ensuite pour lire les données d'un livre et le créer. N'oubliez pas que la création d'un livre peut échouer, vous devez donc traiter cette éventualité : ce sera à la méthode **qui appelle `saisirLivre()`** de traiter les échecs.
8. Dans votre classe `TesteLivre` ajoutez une méthode statique `retireDeTableau` qui étant donné un tableau de livres, un titre, un nom d'auteur et une quantité n, retourne le prix à payer pour n exemplaires de cet ouvrage s'il existe en quantité suffisante dans le tableau. Le tableau devra être modifié en retirant le nombre d'exemplaires demandé. Si le livre n'existe pas ou s'il n'y a pas assez d'exemplaires, votre méthode doit lever des exceptions.
9. Ajoutez dans votre méthode main du code pour lire un titre, auteur et quantité n, et afficher ensuite le prix à payer si le le livre existe dans votre tableau. Vous devrez utiliser la méthode `retireDeTableau` et gérer les exceptions éventuellement levées.
10. Vous finirez votre main par l'affichage de tous les livres du tableau une fois le livre retiré.

## **4 A rendre**

Vous rendrez un fichier zip avec le contenu de votre projet Tp4. Consultez l'énoncé du Tp1 pour savoir comment créer un zip à l'aide d'Eclipse.