

## TP n° 1

## M02 : Prise en main de l'environnement de programmation

Dans votre répertoire utilisateur, créez un répertoire M02. Ensuite, pour chaque TP, vous créez dans ce répertoire un répertoire TPX, où X est le numéro du TP. Par exemple, aujourd'hui vous devez créer un répertoire TP1. Dans ce répertoire, vous créez un répertoire par exercice.

### Exercice 1 [Compilation d'un programme]

Écrivez une classe HelloWorld qui affiche le texte "Hello World!" sur l'écran. Compilez-la et exécutez-la.

### Exercice 2 [Dessins d'étoiles]

1. Écrivez une procédure `affiche_etoiles( nb )` qui affiche une ligne de `nb` étoiles
2. Écrivez une procédure `affiche_blancs( nb )` qui affiche une ligne de `nb` espaces blancs
3. Écrivez une procédure `affiche_char( c )` qui affiche un caractère `c`
4. Modifiez les procédures `affiche_etoiles( nb )` et `affiche_blancs( nb )` de façon à utiliser la fonction `affiche_char( c )`
5. Écrivez une procédure `nouvelle_ligne()` qui fait passer à la ligne suivante
6. Écrivez une procédure `carre( n )` qui affiche un carré de côté `n` :

```
*****
*   *
*   *
*   *
*   *
*****
```

7. Écrivez une procédure `pyramide( n )` qui affiche une pyramide de hauteur `n` :

```
      *
     ***
    *****
   *****
  *****
```

### Exercice 3 [Un livre]

Le but de cet exercice est de créer une classe Livre et d'implémenter ses méthodes.

1. Créez une classe Livre ayant comme attributs le titre du livre (String), le nom de l'auteur (String) et un nombre de pages (int).
2. Tous ces attributs doivent être privés. Écrivez les accesseurs associés (méthodes `set` et `get`).
3. Écrivez une méthode `clone()` qui permet de dupliquer un objet de la classe Livre. Votre classe Livre implémente donc l'interface `Cloneable`.
4. Écrivez une méthode `toString()` qui retourne une chaîne de caractères (String) en vue d'afficher l'état d'un objet.

5. Écrivez un constructeur qui permet de créer un objet de la classe `Livre` en passant comme paramètres le titre du livre, le nom de l'auteur et le nombre de pages.

#### Exercice 4 [La médiathèque]

Reprennez le système conçu en TD de conception orientée objet : les classes `Livre` (qui sera une extension de la classe `Livre` de l'exercice précédent), `CD` et `DVD`.

Un `CD` et un livre sont consultables sur place : les deux classes correspondantes ont donc un attribut de type booléen spécifiant si le livre est en cours de consultation. Elles implémentent toutes les deux une interface `Consultable`, qui définit une méthode `consulter()`.

Les trois classes définissent une méthode `emprunter()`. En cela elles implémentent toutes les trois une interface `Emprunable`. L'implémentation de la méthode `emprunter()` est spécifique à chaque classe.

Les attributs et méthodes en commun entre les trois classes peuvent être regroupés dans une classe abstraite `Document`. Les documents ont notamment une date d'emprunt.

Attention, les classes définies dans cet exercice doivent implémenter l'interface `Cloneable` et implémenter une méthode `toString()`. Leurs attributs seront privés.

1. Complétez la classe `Livre` de l'exercice précédent, implémentez les classes `CD` et `DVD` ainsi que la classe abstraite `Document` et les deux interfaces `Consultable` et `Emprunable`.
2. Un client de la médiathèque est défini par sa date de naissance (son âge est utilisé pour vérifier les restrictions d'emprunts), son nom et son prénom. Définissez et implémentez une classe `Client`.
3. Afin de retrouver quel client a emprunté un document, on inclut une référence vers ce client dans les attributs de ce document. Ajoutez cette référence et les méthodes associées.
4. Les documents de la médiathèque sont stockés dans un objet de classe `Mediatheque` sous forme d'une collection. Choisissez la collection que vous utiliserez et définissez la classe `Mediatheque`.
5. Les objets de la classe `Client` doivent stocker les références vers les documents empruntés. Choisissez la collection que vous utiliserez pour les stocker et définissez les méthodes associées dans la classe `Client`.