Ministère de l'enseignement supérieur et de la recherche scientifique
Institut supérieur des études technologiques de Kélibia

TD2 POO
Classes et objets partie 1

Classes : DSI21 *Enseignante : Safa MAHOUACHI*

Exercice 1 : questions à choix multiples

Q1

a) Il existe deux types de données en Java
b) En Java, toutes les variables sont de type objet
c) `System.out.print` est un objet
d) Les objets possèdent des attributs et fonctionnalités

Q2

a) Le mot réservé **`new`** est utilisé pour créer des nouveaux objets
b) On peut créer un objet sans déclarer de variables
c) Il faut déclarer une variable avant de créer un objet
d) On doit fournir des paramètres afin de pouvoir créer des objets
e) Certains objets ne nécessitent pas de paramètres pour être créés

Q3

a) L'appel à une méthode d'un objet utilise **`.`**
b) L'appel à une méthode d'un objet utilise **`::`**
c) Pour appeler une méthode, il ne faut pas créer d'objets
d) On ne peut pas appeler une méthode sans créer un objet

Q4

a) On peut créer un objet sans créer une classe
b) Un objet est un pointeur vers un espace mémoire
c) Un objet est une variable de type simple
d) Le garbage collector supprime les objets ayant des attributs nuls
e) instanceof permet de tester si deux objets sont égaux

Exercice 2

Définir une classe « Point » comportant :

1. Deux attributs x et y de type double représentant les coordonnées d'un point,
2. Une méthode « affiche » pour afficher les valeurs des coordonnées d'un point,
3. Une méthode « translate » qui permet de faire une translation d'un point de « dx » et de « dy ».
4. Ajouter à la classe « Point » un constructeur ayant deux arguments permettant d'initialiser les valeurs de x et y,
5. Ecrire la méthode main qui permet de créer un point de coordonnées 2.6 et 8, affiche ces valeurs, puis effectue une translation de ce point de 1 et de 3.

Exercice 3

1. Ecrire une classe Personne qui contient :
   - Deux attributs nom et prenom de type String,
   - Un attribut genre de type char,
   - Un constructeur qui permet de créer un objet Personne qui s'appelle « Julien Dupont »
2. Ecrire la méthode main qui permet de créer un objet Personne sans initialiser les attributs,
3. Selon-vous, est ce que c'est possible ? Pourquoi ? Proposer une solution pour pouvoir répondre à la question 2.

Exercice 4

Que fournit le programme suivant :

```
class A {
        int nb=20, decal;
        A (int coeff) {
                nb*=coeff;
                nb+=decal;
        }
        void affiche() {
                System.out.println("nb="+nb+", decal="+decal);
        }
        public static void main(String[] args) {
                A a = new A(4);
                a.affiche();
        }
}
```

Exercice 5

Relevez l'erreur commise dans la définition de classe suivante :

```
class Test {
       final float x;
       final int n=10;
       final int p;
       Test(float r) {
              x=r;
       }
}
```

Exercice 6

1. Définir une classe nommée « Cercle » contenant un attribut de type float nommé « rayon » et un attribut constant « pi »
2. Définir les méthodes suivantes :
    a. « float surface() » qui permet de retourner la surface d'un cercle,
    b. « float permietre() » retourne le périmètre d'un cercle,
    c. « void affiche() » qui permet d'afficher le rayon d'un cercle, son périmètre et sa surface,
    d. La méthode « main » dans laquelle on crée et affiche les informations sur un cercle de rayon « rayon=3.8 ».
3. Ajouter un constructeur qui permet d'initialiser l'attribut rayon.
4. Ajouter la méthode « void modifer(float r) » qui permet de modifier la valeur du rayon d'un cercle.

Exercice 7

1. Créer une classe Etudiant qui contient :
    - un attribut de type `String` nommé `nom` ;
    - un constructeur qui a un paramètre de type `String` servant à initialiser le nom de l'étudiant ;
    - une méthode sans paramètre et qui ne renvoie rien, nommée `travailler`, qui écrit à l'écran, si le nom de l'étudiant a pour nom toto :
      `toto se met au travail !`
    - une méthode sans paramètre et qui ne renvoie rien, nommée `seReposer`, qui écrit à l'écran, si le nom de l'étudiant a pour nom toto :
      `toto se repose`
2. Créer parallèlement à la classe Etudiant une classe TestEtudiant contenant une méthode main qui :
    - crée un étudiant (instance de la classe `Etudiant`) en lui donnant un nom écrit directement dans le fichier source ;
    - invoque la méthode `travailler` de l'étudiant créé ;
    - invoque la méthode `seReposer` de l'étudiant créé.
3. Ajouter à la classe Etudiant une méthode comparer qui permet de tester la conformité de deux étudiants, sachant qu'on considère deux étudiants sont identiques dans le cas où ils ont le même nom. Ceci doit être fait avec deux façons ;
    i. méthode « comparer » prend un seul étudiant en paramètre
    ii. méthode « comparer » prend deux étudiants en paramètre

~Bon travail~