

Kotlin TD 2 POO

Exercice 1 :

Ecrire une classe implémentant une paire d'entier :

1. Définir une classe Paire dont le constructeur initialise les attributs privés de la aire.
2. Définir un deuxième constructeur, qui initialisera à 0 les composants de la paire.
3. Définir un troisième constructeur, qui initialisera une paire à l'aide d'une autre paire.
4. Définir des fonctions permettant d'accéder et de modifier chaque élément de la paire.
5. Définir une méthode affiche et une fonction main pour tester cette classe

Exercice 2 :

Un Rectangle est défini par sa longueur, sa largeur et le caractère qui est utilisé lors de son affichage (* - .). Les opérations que permet un Rectangle sont :

- getSurface : permet de retourner la surface du Rectangle,
- getPerimetre : permet de retourner le périmètre du Rectangle,
- Agrandir : prend en paramètre un entier et permet d'agrandir la longueur et la largeur en fonction de ce paramètre,
- setCaractere : prend en paramètre un caractère qui sera le caractère utilisé pour l'affichage du Rectangle,
- afficher : permet d'afficher le Rectangle, exemple pour (5,3) le rectangle à

afficher est :

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

- 1- Identifier les attributs de la classe Rectangle et donner le type adéquat de chaque attribut.
- 2- Implémenter trois constructeurs de la classe Rectangle :
 - a- Le 1er ne prend aucun paramètre, utiliser (longueur 10, largeur 5, '*')
 - b- Le 2ème prend en paramètre la longueur et la largeur,
 - c- Le 3ème prend en paramètre les trois attributs.
- 3- Implémenter les cinq méthodes de la classe Rectangle.
- 4- Implémenter la méthode main () et qui permet de tester les différentes méthodes de la classe Rectangle

Exercice 3 :

Un parc auto se compose des voitures et des camions qui ont des caractéristiques communes regroupées dans la classe Véhicule.

*Chaque véhicule est caractérisé par son matricule, l'année de son modèle, son prix.

*Lors de la création d'un véhicule, son matricule est incrémenté selon le nombre de véhicules créés.

*Tous les attributs de la classe véhicule sont supposés privés. ce qui oblige la création des accesseurs

*La classe Véhicule possède également deux méthodes abstraites démarrer() et accélérer() qui seront définies dans les classes dérivées et qui afficheront des messages personnalisés.

*La méthode ToString() de la classe Véhicule retourne une chaîne de caractères qui contient les valeurs du matricule, de l'année du modèle et du prix.

*Les classes Voiture et Camion étendent la classe Véhicule en définissant concrètement les méthodes accélérer() et démarrer() en affichant des messages personnalisés.

Travail à faire:

-Créer la classe abstraite Véhicule.

-Créer les classes Camion et Voiture.

- Implémenter la méthode main () qui permet de tester la classe Voiture et la classe Camion

Annexes :variable statique

```
class Statikvar {
    companion object {
        // static count
        var count:Int=0;

        fun getCount(){
            print("No. of objects="+count)
        }
    }
    //constructor
    constructor(){
        count++
    }
}

fun main(args: Array<String>) {
    //object 1
    Statikvar()
    // object 2
    Statikvar()
    //method called using class name
    Statikvar.getCount()
}
```