**Exercices de revision**

**Exercice 1: Notions OO**

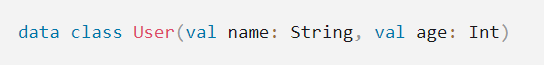
1. Déclarer la classe Point ayant deux attributs privés x et y et les getters
2. Ajouter des setters á cette classe
3. Instancier deux points P1(20,20) et P2(30,20)
4. Afficher les 2 points au terminal comme suit Point(x=….,y= …..)
5. Ajouter une méthode distance(P1 :Point, P2:Point) qui calcule la distance entre deux points
6. Calculer la distance entre P1 et P2

**Exercice 2: Les collections**

1. Declarer une liste de Points **listPoint** contenant initialement P1 et P2
2. Ajouter le point P3(40,50) á cette liste
3. Afficher la liste
4. Filtrer les points dont l´abscisse est <=30
5. Chercher s´il existe un point dont la distance est > 20 par rapport un point racine P(0,0)

**Exercice 3 : data classes**

1. Declarer la classe User



1. Instancier cette classe et afficher línstance
2. Qu´est ce qu´une classe data et que contient comme méthodes par défaut

**Exercice 4: héritage**

1. Déclarer une classe **Compte** avec deux attributs privés numero, proprietaire et solde
2. Ajouter les deux méthodes **debiter(solde)** **crediter(solde)**
3. Déclarer les deux classes **CompteEpargne** et **CompteCourant** ayant les memes attributs et méthodes que Compte
4. Pour la metode debiter, le solde du compte courant peu etre en négatif (-1000) alors que l´épargne ne peut pas etre négatif
5. Ajouter la methode **crediter\_pourcentage()** qui permet d`ajouter un pourcentage **1% 1000** au solde compte d´épargne