

TP N° 1 : CLASSES ET OBJETS

Exercice 1

1. Définir une classe Point comprenant 2 attributs abscisse et ordonnee ainsi qu'un constructeur permettant d'initialiser l'objet.
2. Dans une fonction main(), instancier un point p0 de coordonnées (-1,0).
3. On souhaite afficher à l'écran les coordonnées du point p0. Que faut-il ajouter à la classe Point ?
4. Modifier l'abscisse du point p0 avec la valeur 0. Que faut-il ajouter à la classe Point ? Afficher de nouveau à l'écran les coordonnées du point p0.
5. Dans le main, créer un tableau de 5 points de coordonnées (1,1), (2,2), ..., (5,5) et les afficher à l'écran.
6. Dans la classe Point ajouter une méthode distance(p:Point) qui permet de calculer la distance entre le point courant et le point passé en paramètre. Tester cette méthode dans le main.

La trace d'exécution obtenue est la suivante :

```
p0 : point (-1,0)
p0 : point (0,0)
tableau de points :
point (1, 1) point (2, 2) point (3, 3) point (4, 4) point (5, 5)
la distance entre le point (0,0) et le point (3,3) est 4.242640687119285
```

Exercice 2 : classe Complexe

1. Définir une classe Complexe permettant de décrire et manipuler un nombre complexe.
 - Préciser les attributs utilisés
 - Créer un constructeur qui construit un complexe à partir de ses parties réelles et imaginaires
 - Créer un autre constructeur qui construit (par copie) un complexe à partir d'un nombre complexe passé en paramètre
 - Créer la méthode String toString() qui renvoie une chaîne qui décrit le nombre complexe du type « 2 + i 3 ».
2. Tester votre classe avec la fonction main() ci-dessous :

```
fun main(){
    val c1=Complexe()
    val c2=Complexe(1.0,2.0)
    println(c1)
    println(c2)
}
```

```
0.0 + i 0.0
1.0 + i 2.0
```

La trace d'exécution obtenue est la suivante :

3. Écrire puis tester les méthodes suivantes, en précisant les signatures (c'est-à-dire les paramètres d'entrée et de sortie ainsi que leurs types)
- `im` : qui renvoie la partie imaginaire d'un nombre complexe
 - `re` : qui renvoie la partie réelle d'un nombre complexe
 - `module` : qui renvoie le module d'un nombre complexe
 - `argument` : qui renvoie l'argument principal d'un nombre complexe (attention si la partie réelle vaut 0, l'argument vaut $\pm\pi/2$ en fonction du signe de la partie imaginaire).

Exercice 3 : des calculs sur les complexes

On souhaite ajouter des méthodes permettant de faire des calculs sur les nombres complexes. Ajouter ce qu'il faut à la classe `Complexe` puis tester afin de pouvoir :

1. ajouter 2 nombres complexes $z1+z2 = (x1+x2)+i(y1+y2)$
2. multiplier 2 nombres complexes $z1*z2 = (x1*x2-y1*y2)+i(x1*y2+x2*y1)$
3. mettre à la puissance n un nombre complexe (en utilisant la multiplication de 2 nombres complexes)
4. calculer l'inverse d'un nombre complexe non nul $1/z = \frac{x}{x^2+y^2} - i \frac{y}{x^2+y^2}$
5. calculer le quotient de 2 nombres complexes (en utilisant les opérations précédentes) $z1/z2 = \frac{x1*x2-y1*y2}{x2^2+y2^2} + i \frac{x2*y1-x1*y2}{x2^2+y2^2}$

Vérifier avec $z1=1+i2$ et $z2=2+i1$

$$z1+z2 = 3+i3$$

$$z1*z2 = 0+i5$$

$$z^5 = 41-i38$$

$$1/z1 = 0,2 - i0,4$$

$$z1/z2 = 0,8 - i0,6$$

Exercice 4 : des constantes importantes

Quand on manipule des complexes, certaines constantes sont importantes, comme

$$i = 0 + i \quad \text{et} \quad j = -\frac{1}{2} + i\sqrt{3}/2$$

1. Créer les constantes de classe dans la classe `Complexe` (nommée `I` et `J`)
2. Vérifier en utilisant les fonctions et constantes définies précédemment que $J^3 = 1$. L'appel dans le main est de la forme :

`Complexe.J.puissance(3)`