

# Développement Logiciel dans l'environnement JAVA

**Nom:** Khouni

**Prénom:** Mohamed Yahya

## Exercice 1

Un nombre complexe est défini par  $z=a+ib$ , où  $a$  et  $b$  sont des réels et  $i$  un nombre imaginaire tel que  $i^2=-1$ , soient  $z_1=a_1+ib_1$  et  $z_2=a_2+ib_2$  deux nombres complexes, leurs somme est définie par  $z=z_1+z_2=a_1+a_2+i(b_1+b_2)$  et leurs produit par  $z=z_1*z_2=a_1*a_2-b_1*b_2+i(a_1*b_2+a_2*b_1)$

**Question :** Ecrire avec Java une classe **Complexe** qui offre :

- un constructeur sans paramètre qui construit le complexe  $0+0i$ ,
- un constructeur qui prend 2 réels  $a$  et  $b$  et qui construit le complexe  $a+ib$ ,
- les setters/getters appropriés
- l'addition et la multiplication

```
public class Complexe {  
    private double a;  
    private double b;  
  
    public Complexe() {  
        this.a = 0;  
        this.b = 0;  
    }  
  
    public Complexe(double a, double b) {  
        this.a = a;  
        this.b = b;  
    }  
  
    public double getA() {  
        return a;  
    }  
  
    public void setA(double a) {  
        this.a = a;  
    }  
}
```

```

public double getB() {
    return b;
}

public void setB(double b) {
    this.b = b;
}

public Complexe addition(Complexe z) {
    double newA = this.a + z.a;
    double newB = this.b + z.b;
    return new Complexe(newA, newB);
}

public Complexe multiplication(Complexe z) {
    double newA = this.a * z.a - this.b * z.b;
    double newB = this.a * z.b + z.a * this.b;
    return new Complexe(newA, newB);
}
}

```

## Exercice 2

Une liste de complexes est une structure en mesure d'accueillir des nombres complexes sans limite de taille. Elle offre les opérations suivantes :

- Le constructeur qui construit une liste vide
- L'Ajout qui permet d'ajouter un nombre complexe au début de la liste
- La suppression qui permet de supprimer le complexe au début de la liste
- La recherche qui permet de retrouver le complexe qui se trouve en position  $i$  de la liste, où  $i$  est un entier donné
- La produit qui permet de calculer le produit de tous les complexes de la liste

Question : Ecrire avec Java une classe **Liste** qui offre les caractéristiques ci-dessus, on pourra pour cela introduire une classe **Noeud**

```

public class Noeud {
    private Complexe valeur;
    private Noeud suivant;

    public Noeud(Complexe valeur) {
        this.valeur = valeur;
        this.suivant = null;
    }
}

```

```

public Complexe getValeur() {
    return valeur;
}

public void setValeur(Complexe valeur) {
    this.valeur = valeur;
}

public Noeud getSuivant() {
    return suivant;
}

public void setSuivant(Noeud suivant) {
    this.suivant = suivant;
}
}

```

```

public class Liste {
    private Noeud tete;

    public Liste() {
        this.tete = null;
    }

    public void ajouter(Complexe c) {
        Noeud nouveau = new Noeud(c);
        nouveau.setSuivant(tete);
        tete = nouveau;
    }

    public Complexe supprimer() {
        if (tete == null) {
            return null;
        }
        Complexe valeur = tete.getValeur();
        tete = tete.getSuivant();
        return valeur;
    }

    public Complexe rechercher(int i) {
        if (i < 0) {
            return null;
        }
        Noeud courant = tete;
    }
}

```

```

    int position = 0;
    while (courant != null && position < i) {
        courant = courant.getSuivant();
        position++;
    }
    if (courant == null) {
        return null;
    }
    return courant. getValeur();
}

public Complexe produit() {
    if (tete == null) {
        return null;
    }
    Complexe resultat = new Complexe(1, 0);
    Noeud courant = tete;
    while (courant != null) {
        resultat = resultat.multiplication(courant.getValeur());
        courant = courant.getSuivant();
    }
    return resultat;
}
}

```

```

public class Main {
    public static void main(String[] args) {
        Complexe z1 = new Complexe();
        Complexe z2 = new Complexe(3, 4);
        Complexe z3 = new Complexe(1, 2);

        Complexe somme = z2.addition(z3);
        Complexe produit = z2.multiplication(z3);

        Liste liste = new Liste();
        liste.ajouter(z1);
        liste. ajouter(z2);
        liste.ajouter(z3);

        Complexe recherche = liste.rechercher(1);
        Complexe supprime = liste.supprimer();
        Complexe produitListe = liste.produit();
    }
}

```