

Notion d'objet

- A côté des langages impératifs, fonctionnels, logiques, ...
- Les objets: des composants logiciels à composer



- Des entités autonomes, encapsulées, communicantes et interagissantes, réutilisables et construites suivant les besoins

Notion d'objet (suite)

- Un objet possède des composants et peut servir à effectuer certaines actions...
- **Attributs** et **méthodes**

Notion d'objet (suite)

- Un objet possède des composants et peut servir à effectuer certaines actions...

- **Attributs et méthodes**

Exemple

Un nombre complexe:

- est composé de sa partie réelle et sa partie imaginaire,
- peut être initialisé
- peut fournir des résultats de certaines opérations
- *Rmq. le produit est effectué par un objet de la classe des « Complexe », prend en paramètre un autre Complexe et rend comme résultat un autre Complexe*

Complexe	
Entier	a
Entier	b
initialiser (Complexe z) :	
conjugué () : Complexe	
produit (Complexe z) : Complexe	
module () : Réel	

Un programme en Java

Exemple

```
class ProgrammePrincipal {
    public static void main ( String arg[ ] ) {
        Complexe z1 = new Complexe(1, 2); // z1 est le complexe 1+2i
        Complexe z2 = new Complexe(2, 2); // z2 est le complexe 2+2i
        z1 = z1.ajouter(z2); // z1 devient le complexe 3+4i
        System.out.println ( z1.module() ) ; // affiche 5
        z1.afficher(); // affiche 3+4i
    }
}
```

Complexe
Réel a
Réel b
Complexe (entier r, entier i) :
conjugué () : Complexe
ajouter (Complexe z) : Complexe
module () : Réel
afficher () :

Un programme en Java

Exemple

```
class Complexe{
  int a;
  int b;

  Complexe (int r, int i) {
    a = r;
    b = i;
  }
  public Complexe ajouter (Complexe z){
    a = a+z.a;
    b = b+z.b;
    return this;
  }
  public int module (){ . . . }
  public void afficher(){
    System.out.println(a+" "+b+"i");
  }
}
```

```
public class ToutPetitTest {
  public static void main(String[] a) {
    Complexe z1 = new Complexe(1, 2);
    Complexe z2 = new Complexe(2, 2);
    z1 = z1.ajouter(z2);
    System.out.println (z1.module());
    z1.afficher();
  }
}
```

Complexe

Entier a

Entier b

Complexe (entier r, entier i) :

conjugué () : **Complexe**

ajouter (**Complexe z**) : **Complexe**

module () : Réel

afficher () :

Exécution d'un programme par communication entre des objets

Définition

Le module d'un nombre complexe $|z|$ est la distance qui sépare l'origine du repère complexe au point M d'affixe z.
De plus, pour $z = x + iy$, on a :

$$|z| = \sqrt{x^2 + y^2} = \sqrt{z\bar{z}}$$

- Calcul du module

- Exemple. Si z est un complexe :

```
Complexe z1 = z . conjugué() ;
```

- Exemple. Si z est un complexe :

```
Réel m = (z . produit (z . conjugué()) ) . racine() ;
```

- Résultat de plusieurs envois de messages à différents objets.

- *Rmq. problème de typage parce que la racine d'un complexe n'est pas forcément un réel... (cf. plus loin)*

Création d'un objet

- Pour pouvoir exécuter une action, tout objet doit avoir été **déclaré** et **créé**

- Exemple. Déclaration et Création d'un complexe :

```
Complexe z1 = new Complexe (3, 4) ;
```

- Autre exemple de création d'un complexe :

```
Complexe z2 = new Complexe (z1) ;
```

Rmq. identité et dualité vs égalité et différence

- Encore un autre mode de création d'objet

```
Complexe z3 = z . conjugué() ;
```

Un programme en Java

Exemple en Java

```
class ProgrammePrincipal {
    public static void main ( String arg[ ] ) {
        Complexe z1 = new Complexe(1, 2);
        Complexe z2 = new Complexe(2, 2);
        z1 = z1.ajouter(z2);
        System.out.println ( z1.module ) ; // affiche 5
        z1.affiche(); //affiche 3+4i si la méthode d'affichage est bien faite
    }
}
```

Un programme en Java

Exemple en Java

```
class ProgrammePrincipal {  
    public static void main ( String arg[ ] ) {  
        Complexe z1 = new Complexe(1, 2);  
        Complexe z2 = new Complexe(2, 2);  
        z1 = z1.ajouter(z2);  
        System.out.println ( z1.module ) ; // affiche 5  
        z1.affiche(); //affiche 3+4i si la méthode d'affichage est bien faite  
    }  
}
```

le nom de la classe principale qui exécute le programme

sa méthode main

la création d'un complexe

l'exécution d'une méthode par un objet

l'exécution d'une méthode par le même objet
celle-ci ne renvoie pas de résultat

Petit travail pratique : premier programme en Java

```
class ProgrammePrincipal {  
    public static void main ( String arg[ ] ) {  
        Complexe z1 = new Complexe(1, 2);  
        Complexe z2 = new Complexe(2, 2);  
        z1 = z1.ajouter(z2);  
        z1.affiche();//affiche 3+4i si la méthode  
            d'affichage est bien faite  
    }  
}
```

Travail demandé :

- charger le fichier **Complexe.java**
- créer le fichier **ProgrammePrincipal.java** (contenant le programme ci-dessus) en utilisant un éditeur de texte
- compiler le programme principal par la commande : `javac ProgrammePrincipal.java`
- exécuter le programme principal par la commande : `java ProgrammePrincipal`
- *ajouter une méthode pour créer le conjugué d'un complexe*
- *afficher le complexe conjugué d'un complexe*