Université de Biskra
Département d'informatique

Cours POO

Vidéos 12-13-14

Constructeur, Destructeur, Setters & getters, Objet *this

(Partie 3)

Pr. Laid Kahloul

L2 2020

Plan

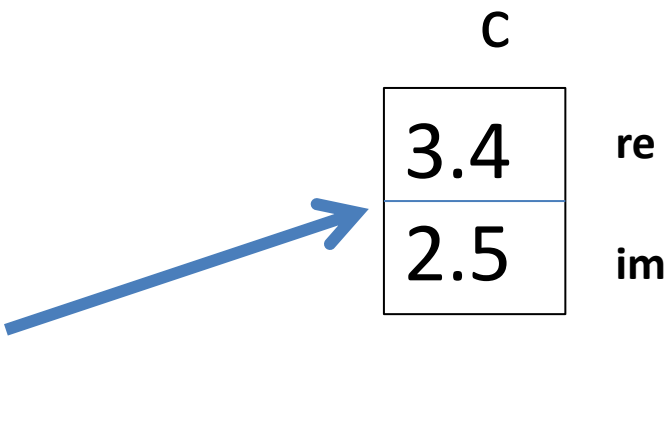
- Constructeur
- Destructeur
- Setters & getters
- Objet *this

Constructeur

Exemple: class Complexe (1)

- Exemple class Complexe:

```
class Complexe{
float im, re;
        Complexe(float im0, float re0);
};
Complexe:: Complexe(float im0, float re0){
im=im0;
re=re0;
}
void main(){
Complexe c=Complexe(3.4, 2.5);
}
```



Constructeur

Exemple: conversion de type (2)

- Il est possible de faire une converstion de type dans la POO lors de l'appel du constructeur

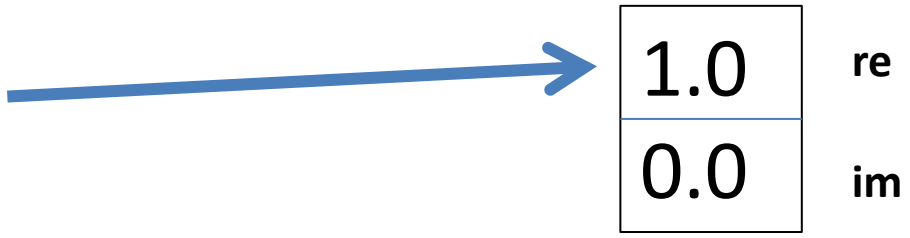
```
Complexe::Complexe(float x) {
    re = x;
    im = 0.0;
}
```

Constructeur

Exemple: conversion de type (3)

- Dorénavant, toute déclaration :

Complexe z=1.0;



Génère un complexe dont la valeur de re est 1.0

- D'où on convertie le **float** x en un complexe directement ;

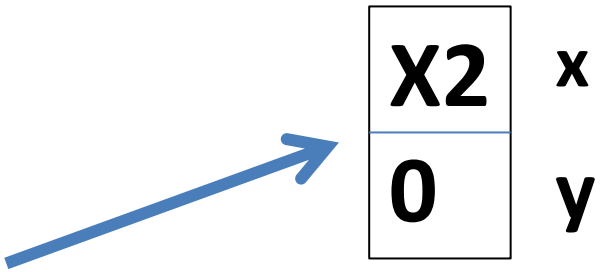
Constructeur

Initialisation d'attributs avant les accolades

- Possibilité d'initialiser les attributs avant les accolades des méthodes.

```
class C{
        int x; int y;
    public:
        C(int x2);
        };

C::C(int x2):x(x2),y(0){
    }
```

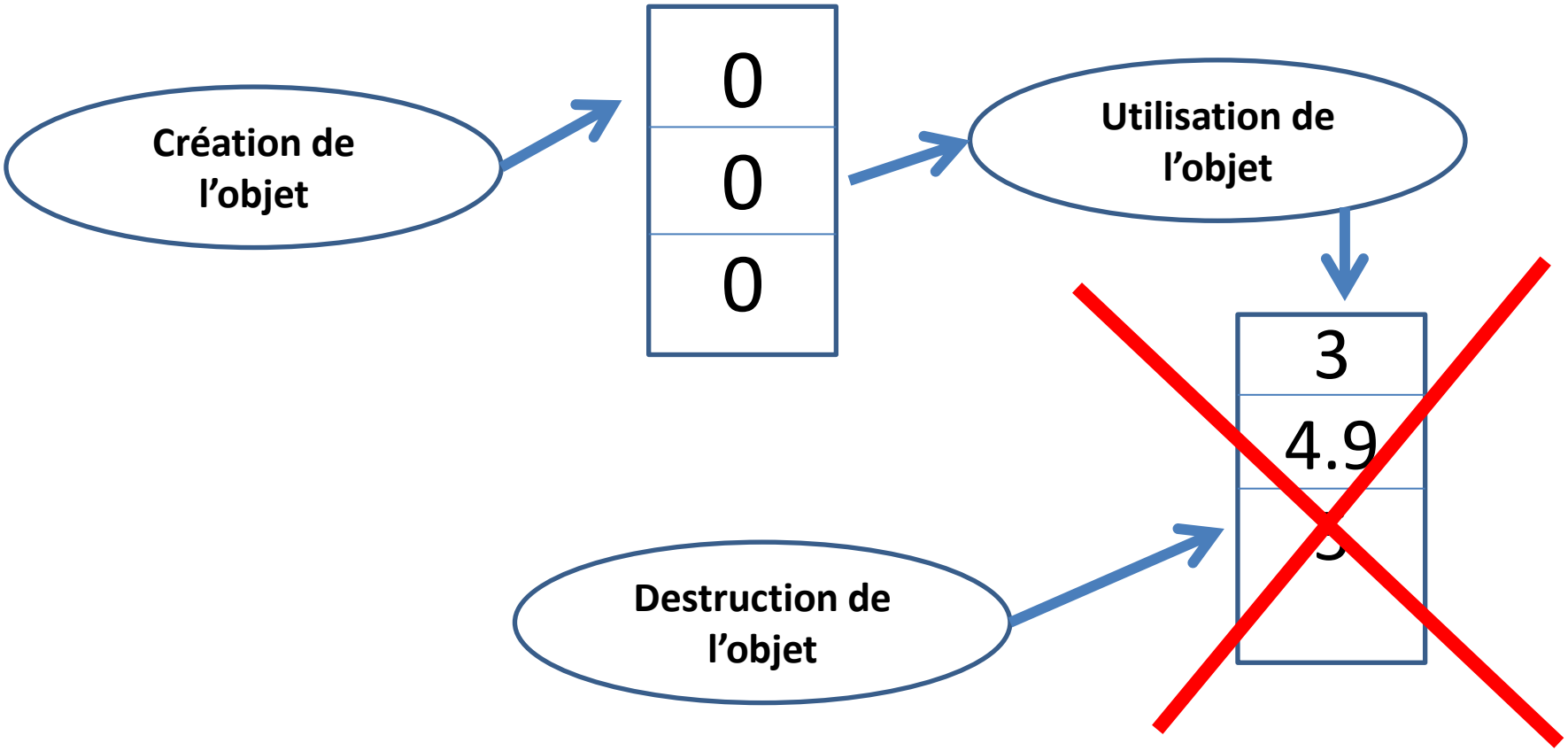


Destructeur (1)

- C’est quoi?

Une **méthode particulière** qui permet de **supprimer** des **objets** de la **mémoire** après la fin de leur utilisation

Destructeur (2)
vie d'un objet



Destructeur (3)
Dans la programmation

- La méthode destructeur est une méthode qui **peut être ajouté** dans des langages de POO comme le cas du **C++;**

- Dans certains langages de programmation comme **Java**, le programmeur n'a pas besoin d'ajouter cette méthode et déjà **ce n'est pas permis**.

Destructeur (3)
dans le C++

- Nom de cette méthode?

Si ***nom_classe*** est le nom de la classe alors le destructeur doit porter le nom:

~nom_classe()

- Le symbole ~ est obtenu par les touche:

ALT Gr+2

- Appel du destructeur: comme l'appel des autre méthode:

nom_objet.nom_destructeur();

Destructeur (4)
Exemple C++

```
class A{
    A();// constructeur
    ~A();// destructeur
};
A::A(){
// mettre ici le code d'initialisation par exemple
}
A::~A(){
}
Void main(){
        A a=A();// création de l'objet par appel du constructeur
        a.~A();//appel du destructeur
}
```

Destructeur (5): dans le C++

- Dans le C++, le programmeur peut ajouter un destructeur comme il peut ne pas l'ajouter.

- Si l'utilisateur ajoute un destructeur, **il peut le personaliser** en ajoutant des **messages** informant la suppression de l'objet.

- Mais l'utilisateur **ne doit pas mettre du code** de suppression de l'objet dans la méthode.

- **On peut l'appeler même si il n'est pas ajouter explicitement ;**

Destructeur (5): Exemple en C++

```
class A{
    A();// constructeur
    ~A();// destructeur
};
A::~A(){
cout<<"Je suis détruit";
}
Void main(){
        A a=A();// création de l'objet par appel du constructeur
        a.~A();//appel du destructeur
}
```

Destructeur (6): Exemple en C++: sans le déclarer

```
class A{
    A();// constructeur
    ~A();// destructeur
};
A::~A(){
cout<<"Je suis détruit";
}
Void main(){
        A a=A();// création de l'objet par appel du constructeur
        a.~A();//appel du destructeur
}
```

Destructeur (7): En C++: Remarque

- Si le programmeur n'appelle pas le destructeur, donc le destructeur sera **appelé de manière automatique** à la fin du programme main().