Cours de POO avec Java (Partie 1)

Module GL et POO
2LMD,
2013
Kahloul Laid

Objectif

Apprendre le minimum nécessaire pour écrire un code java et l'exécuter avec le minimum d'outils (javac, java, JDK, JVM).

Plan du cours

- **C'est du C avancé**: types de base, casting, entrée sortie, …;
- **Les Classes java**: attributs, méthodes (signature & implémentation), méthode main, visibilité des attributs et méthodes;
- **Les Objets java**: instance par clonage, instance statique, méthode main **static**;
- **Fonctionnement**: javac, java, JVM, exemple concret.

Types de données de base

byte	un entier signé sur 8 bits	de –128 à +127
short	un entier signé sur 16 bits	de –32 768 à +32 767
int	un entier signé sur 32 bits	de –2 147 483 648 à 2 147 483 647
long	un entier signé sur 64 bits	de l'ordre de (±) 9 milliards de milliards
float	un réel sur 32 bits	de l'ordre de 1.4E-45 à 3.4E38
double	un réel sur 64 bits	de l'ordre de 4.9E-324 à 1.79E308
boolean	true ou false	
char	un caractère Unicode sur 16 bits	entier positif entre 0 et 65 535

Transtypage: casting

Pour changer le type d'une valeur val, on utilise la syntaxe:

(type)valeur

Exemple:

```
Int x=10;
Long y=x*(long) 2.5f;
Long z=(long) (x*2.5f);
```

Instructions d'entrée/sortie de base

- Entrée:

```
int System.in.read (); // lire un octet
```

- Sortie:

```
System.out.println();
```

Classe (1): attributs

```
class nom_classe{

//Liste des attributs

//Liste des méthodes (signature et implémentation)

}
```

Exemple:

```
class cercle{
//Liste des attribut
    int x; int y; int r;
// liste des méthodes
                    }
```

Classe (2): méthode(signature)

Exemple:

```
class cercle{
  // liste des méthodes;
     int x; int y; int r;
  // liste des méthodes;
     void initialiser(int x0, int y0, int r0);
          }
```

Classe (2): méthode(implémentation)

Exemple:

```
class cercle{
 // liste des méthodes;
        int x; int y; int r;
 // liste des méthodes;
        void initialiser(int x0, int y0, int r0){
          x= x0;
          y= y0;
          r= r0;
        }
 }
```

Classe (3): méthode principale

- Pas de notion de **programme main** comme dans le langage C !!!!

- Mais on peut avoir une **méthode main** dans une classe, d'où l'exécution du programme java commence.

Classe (4): méthode principale

Exemple:

```
class cercle{
 // liste des attributs;
        int x; int y; int r;
 // liste des méthodes;
        void initialiser(int x0, int y0, int r0){
        x= x0; y= y0; r= r0;
        }

        void main(){
        //afficher le rayon
        System.out.print(r);
        }

 }
```

Classe (5): Visibilité (portée) des attributs et des méthodes: **public** vs **privé**

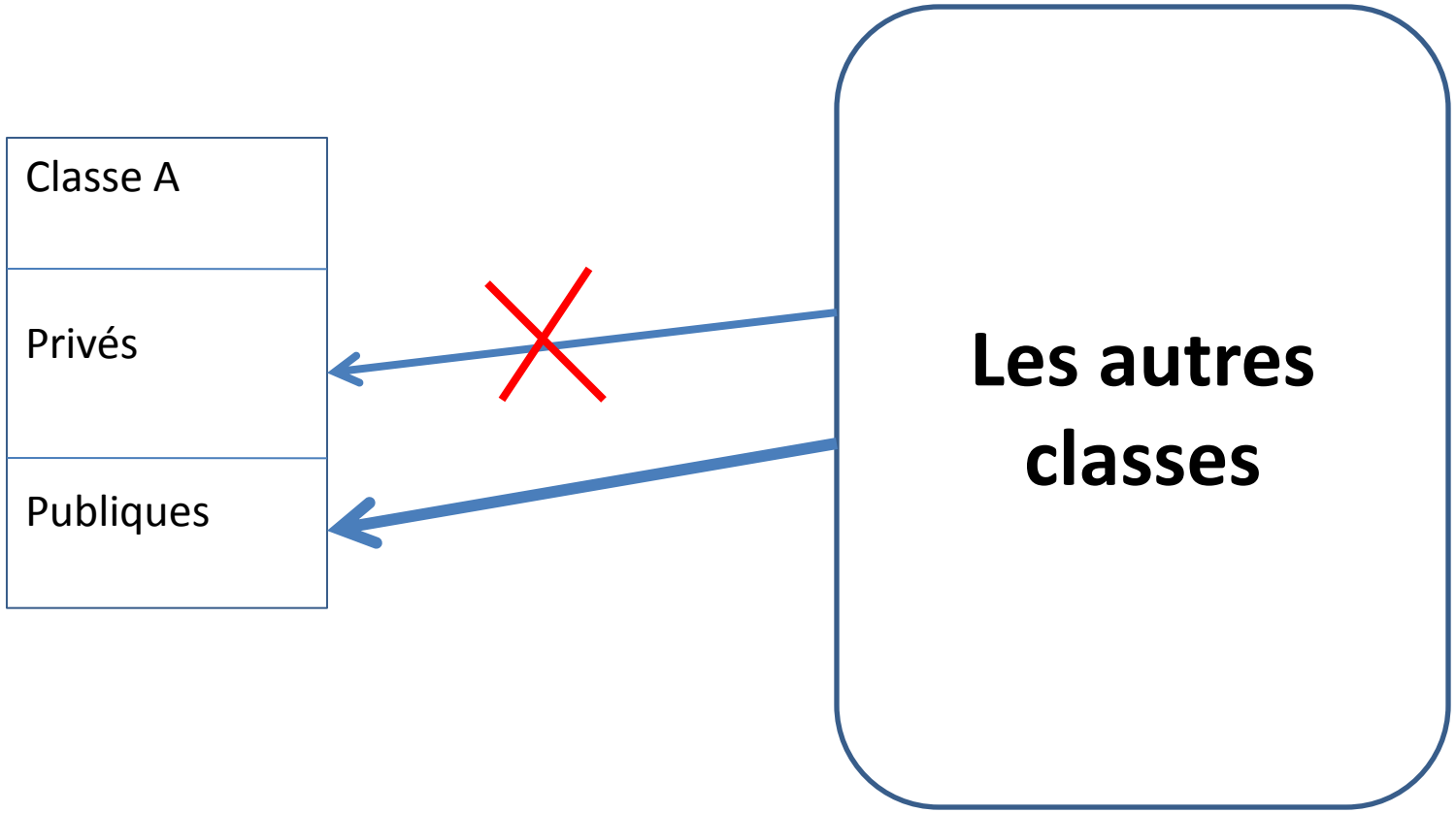
Visible par tout:

- un attribut ou une méthode **publique** est visible par tout => accessible par tout. On peut l'utiliser sans avoir (unknown identifier)
- L'attribut où la méthode doit être précédé par le mot clé **public**

Invisible hors sa classe:

- un attribut ou une méthode **privé** est visible uniquement dans sa classe=> inaccessible hors la classe.
- L'attribut où la méthode doit être précédé par le mot clé **private**
- Par défaut , tous les attributs et les méthodes sont privés

Classe (6): Visibilité (portée) des attributs et des méthodes



L’objet (1)

- Objet= **instance** d’une classe;
- L’exécution du programme **doit créer** des **objets** (au moins un objet);
- Programme java=**ensemble de classes**, au moins une de ces classes doit être **la classe principale**, d’où l’exécution doit démarrer.

Objet (2): instance par **déclaration** ou instance **statique**

- Le programmeur peut créer un objet par une déclaration (dans des **lieux spécifiques** dans son programme java)

  exemple:

  cercle c=**new** **cercle**();// crée un objet cercle

- Le programmeur peut exécuter son programme **sans créer un objet par déclaration**

Objet (3): méthode main **statique**

- Ceci exige que la méthode main doit être statique (**static**); ça veut dire ne nécessite pas le clonage.

On doit précéder la méthode main par le mot clé **static**

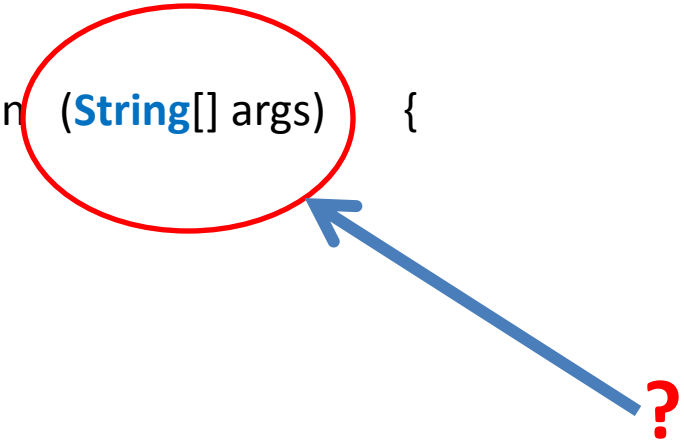
Objet (4): méthode main statique

Exemple:

```
class cercle{
  // liste des attributs;
        int x; int y; static int r;
  // liste des méthodes;
        void initialiser(int x0, int y0, int r0){
        x= x0; y= y0; r= r0;
        }

        public static void main (String[] args)  {
        //afficher le rayon
        System.out.print(r);
        }

  }
```



Fin de cette partie

Questions????