Année 2008-2009

esil

Les objets Java

Nicolas Baudru
mél : nicolas.baudru@esil.univmed.fr
page web : nicolas.baudru.perso.esil.univmed.fr

Programmation procédurale ou orientée objet (OO)

esil

Spécification du programme :

- Un cercle ou un carré s'affiche sur une interface graphique.
- Lorsque l'utilisateur clique sur un bouton la forme affichée doit tourner.
- Lors de la rotation d'une forme, le système jouera une mélodie spécifique à cette forme.

Approche procédurale :

Que doit faire ce programme ?

De quelles procédures ai-je besoin ?

Approche OO :

Quelles sont les entités de ce programme ?

Quels sont les acteurs clés ?

Approche procédurale :

Les procédures tourner() et jouerSon() :

```
tourner(typeForme) {
  //faire tourner la forme
}

jouerSon(typeForme) {
  // utiliser typeForme
  // pour déterminer
  // quel son jouer
}
```

Approche OO :

Les formes Cercle et Carre :
il faut donc créer une classe pour chaque forme.

Cercle
tourner()
jouerSon()

Carre
tourner()
jouerSon()

La spécification du programme change :

- Un cercle ou un carré ou un triangle s'affiche sur une interface graphique.

Approche procédurale :

Qu'est-ce qui change ?

Il faut modifier jouerSon() :

```
jouerSon(typeForme) {
  if(forme != triangle){
    // utiliser typeForme
    // pour déterminer
    // quel son jouer
  }
  else
    // jouer le son
    // du triangle
}
```

Approche OO :

Qu'est-ce qui change ?

il faut créer une nouvelle classe pour le triangle. Pas besoin de toucher au reste du code : flexibilité et extensibilité.

Triangle
tourner() jouerSon()

La spécification était imprécise :

- Un carre et un cercle tournent autour de leur centre.
- Un triangle tourne autour d'un de ses sommets !

Approche procédurale :

Qu'est-ce qui change ?

Il faut modifier tourner(), et peut-être de manière importante

Il faut ensuite recompiler et retester l'ensemble du programme...

Approche OO :

Qu'est-ce qui change ?

Il suffit de modifier la méthode tourner() du triangle.

Il faut ensuite recompiler et retester la classe Triangle seulement...

Au final, on obtient 3 classes :

Cercle
tourner() (centre) jouerSon()

Carre
tourner() (centre) jouerSon()

Triangle
tourner() (sommet) jouerSon()

Pour ne pas dupliquer certains morceaux de code, les caractéristiques communes peuvent être extraites et placées dans une classe plus abstraite nommée Forme :

Forme
tourner() (centre) jouerSon()

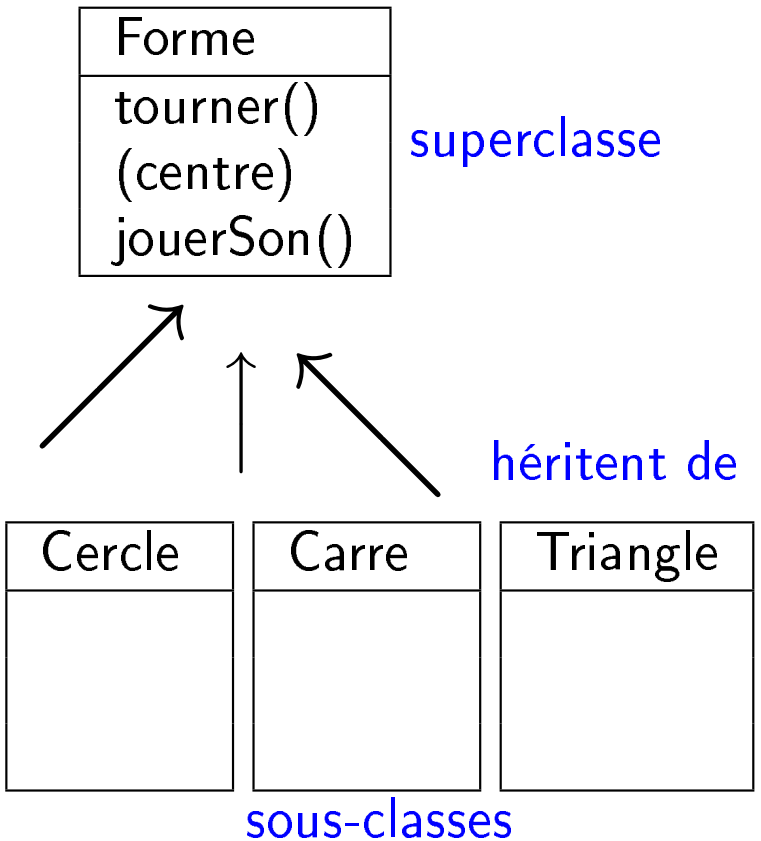
Cercle

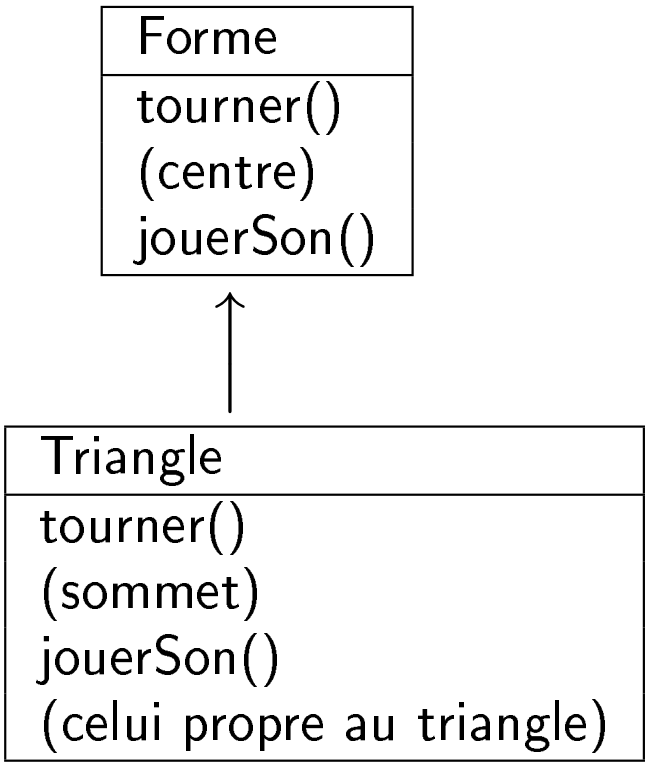
Carre

Triangle

Notion d'héritage : sous-classes et superclasse

Les sous-classes Carre, Cercle et Triangle héritent de la superclasse Forme. Les sous-classes héritent des méthodes de la superclasse. Ainsi il ne reste qu'un seul exemplaire de tourner() et jouerSon() à maintenir.





Il suffit alors de redéfinir des métodes spécifiques lorsque cela est nécessaire

Comment le programme sait quelle méthode utiliser ?

Un programme est un ensemble de classes. Une des classes doit comporter une méthode main() qui sert à lancer le programme.

Toutes ces classes sont souvent regroupées dans une archive Java, un fichier .jar. Dans ce fichier jar, vous devez inclure un manifeste, c.a.d. un fichier texte qui définit dans quelle classe se situe la méthode main() de votre application.

C'est ce fichier jar que vous livrerez à vos utilisateurs.

Lorsque vous concevez une classe, il faut penser aux objets qui seront créés à partir de cette classe :

Qu'est ce qu'un objet sait ? Les variables d'instance définissent l'état de l'objet

Qu'est ce qu'un objet fait ? Les méthodes définissent le comportement de l'objet

Exemple :

Chien
taille poids race nom
aboyer()

Machine à café
niveauEau niveauCafé
caféLong() caféCourt() eau()

Téléphone cellulaire
variables d'instance ?
méthodes ?

Différences entre classe et objet

Une classe n'est pas un objet. Une classe est un patron d'objet.

La classe explique à la machine virtuelle comment produire un objet d'un type donné. Une classe définit les caractéristiques générales d'un objet. Un objet est une instance d'une classe, c.a.d. un objet créé à partir d'une classe, en affectant aux variables d'instance des valeurs qui lui sont propres.

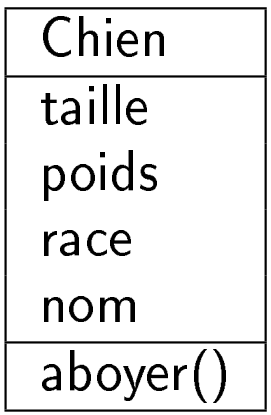
Chien
taille poids race nom
aboyer()

Une classe Chien

58 cm
25 kg
berger allemand
Fritz

55 cm
31 kg
labrador
Noireau

plusieurs objets Chien




```
public class Chien {
   int taille;     // |                    |
   int poids;      // | Les variables      |
   String race;    // |  d'instance        |
   String nom;     // |                    |

   void aboyer(){ // la méthode aboyer()
      System.out.println("Ouah! Ouaf!");
   }
}
```

Les points importants :

- Comment créer un objet Chien ?

```
Chien monChien = new Chien();
```

L'opérateur new vous permet de créer un objet d'une classe donnée.

- Comment manipuler les instances et les méthodes de l'objet créé ?

```
monChien.taille = 55; // monChien mesure 55cm
monChien.aboyer();  // monChien aboie
```

L'opérateur (.) vous donne accès à l'état et au comportement d'un objet.

- Où créer l'objet Chien ?

```
public class TestChien {
  public static void main(String[] args){
    Chien monChien = new Chien();
    monChien.taille = 55;
    monChien.aboyer();
  }
}
```

1 Spécification du programme :

Nous allons écrire un petit programme correspondant au jeu suivant. L'ordinateur choisit un nombre compris entre 0 et 9, et un joueur essaie de le deviner.

Quelles classes sont nécessaires ?

2 Quelles classes sont nécessaires ?

Joueur
nombre
deviner()

Jeu
j1
commencer()

TestJeu
main()

3 L'idée :

- la classe TestJeu démarre l'application. Elle contient donc la classe main().
- la méthode main() crée un objet Jeu et appelle sa méthode commencer().
- la méthode commencer() de l'objet Jeu est l'endroit où tout le jeu se déroule. Elle crée un objet Joueur puis choisit un nombre aléatoirement. Elle demande ensuite au joueur de deviner ce nombre.
- La méthode deviner() de l'objet Joueur permet à un joueur de deviner un nombre entre 0 et 9, puis de le stocker dans sa variable nombre.

Un programme avec des objets

```
public class Joueur {
  int nombre = 0; // le nombre proposé

  public void deviner() {
    nombre = (int) (Math.random() * 10);
    System.out.println("Je␣propose" + nombre);
  }
}

public class TestJeu {
  public static void main(String[] args) {
    Jey jeu = new Jeu();
    jeu.commencer();
  }
}
```

esil

```
public class Jeu {
  Joueur j1;

  public void commencer() {
    j1 = new Joueur();

    int nombreCible =  (int) (Math.random()*10);
    System.out.println("Ordi␣choisit␣" + nombreCible);

    while(true) {
      j1.deviner();
      System.out.println("j1␣a␣choisit" + j1.nombre);

      if(j1.nombre == nombreCible) {
        System.out.println("j1␣a␣trouvé");
        break;
      }
      else { System.out.println("j1␣n␣a␣pas␣trouvé");}
    }
  }
}
```

Résultat de notre programme

Ci-dessous une des exécutions possibles de notre application :

```
% javac Joueur.java
% javac Jeu.java
% javac Testjeu.java
% java TestJeu
Ordi choisit 4
j1 a choisit 3
j1 n a pas trouvé
j1 choisit 8
j1 n a pas trouvé
j1 choisit 4
J1 a trouvé
```

Qui suis-je ?

une classe, un objet, une méthode ou une variable d'instance ?

- Je résulte de la compilation d'un fichier .java
- Je me comporte comme une recette de cuisine
- J'aime faire des choses
- Je peux changer lors de l'exécution
- Je peux avoir plusieurs méthodes
- Je sers à créer des objets
- Je représente un état
- Je me trouve dans un objet

Qu'avons nous vu jusqu'ici ?

esil

- Pourquoi la conception objet est plus efficace que la conception procédurale ?
- La différence entre une classe et un objet.
- Comment définir une classe, une variable d'instance, une méthode.
- Comment créer une instance d'une classe (i.e. un objet).
- Comment invoquer une méthode d'un objet.
- Une première approche de la notion d'héritage.