

Notions d'objet en Java

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 24 novembre 2016

Sous-routines

- Sous-routine : entrée (facultative), ensemble d'instructions, sortie (facultative)
- Fait qqch
- On peut l'appeler sans savoir comment elle procède (boîte noire)
- `Math.random()` ; `Math.sqrt(4)` ;
- Sous-routine `sqrt` statique se trouvant dans classe `Math`
- Classes regroupent des variables et sous-routines statiques
- `System.out` : variable statique `out` dans classe `System`
- Permet d'*organiser* et de *nommer*

String vers int

- Création d'un int ?

String vers int

- Création d'un int ?
 - 2
 - `int zeRoot = Math.sqrt(4);`

String vers int

- Création d'un int ?
 - 2
 - `int zeRoot = Math.sqrt(4);`
- Parfois possible de créer une instance avec la classe elle-même
- `int i = Integer.parseInt("10");`

Deux rôles des classes

- Classe : conteneur à variables et sous-routines statiques
- Mais peut aussi être considérée comme un *type* : spécification, modèle, moule à objets

La trahison des images



Classes et objets

- Classe : moule à objets, appelée type de l'objet
- Objet : instance de classe
- Exemple : classe `V` modélise une voiture
- Objet correspondant est une voiture donnée (de type `V`)
- Classe `String`, Objet : `"blah"`
- Classe `Scanner`, Objet `scanner` utilisé pour récupérer le texte entré par l'utilisateur

Opérations sur String

- `"Hello".toUpperCase();`
- `"Hello".indexOf("l");` ⇒

Opérations sur String

- `"Hello".toUpperCase();`
- `"Hello".indexOf("l");` \Rightarrow 2
- `"Hello".substring(1, 3);` \Rightarrow

Opérations sur String

- `"Hello".toUpperCase();`
- `"Hello".indexOf("l");` \Rightarrow 2
- `"Hello".substring(1, 3);` \Rightarrow "el"

Attention au test d'égalité :

✓ utiliser `s1.equals(s2)`

✗ et *pas* `s1 == s2`

Scanner

- Objet créé par `new`
- `Scanner stdin = new Scanner(System.in);`
- `String userName = stdin.nextLine();`

Licence

Cette présentation, et le code LaTeX associé, sont sous [licence MIT](#). Vous êtes libres de réutiliser des éléments de cette présentation, sous réserve de citer l'auteur.
Le travail réutilisé est à attribuer à [Olivier Cailloux](#), Université Paris-Dauphine.