

Notions d'objet en Java

Olivier Cailloux

LAMSADE, Université Paris-Dauphine

Version du 12 décembre 2020

Deux rôles des classes

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

Rôle 1 : Conteneur statique

- Classes regroupent des variables et sous-routines statiques
- Permet d'*organiser* et de *nommer*
- `Math.sqrt(Math.PI)` ;
- Méthode `sqrt` statique dans classe `Math`

Rôle 2 : Moule à objets

- Classe : moule à objets, appelée type de l'objet
- Objet : instance de classe
- Exemple : classe Car modélise une voiture
- Objet correspondant est une voiture donnée (de type Car)
- Classe déjà utilisée dans ce cours ?

Rôle 2 : Moule à objets

- Classe : moule à objets, appelée type de l'objet
- Objet : instance de classe
- Exemple : classe Car modélise une voiture
- Objet correspondant est une voiture donnée (de type Car)
- Classe déjà utilisée dans ce cours ? String, Objet : "blah"

Création d'objets

① Via *constructeur* : new + nom de la classe

- `Random r = new Random();`
- `IllegalArgumentException exc1 = new
IllegalArgumentException("Must be positive.");`
- `IllegalArgumentException exc2 = new
IllegalArgumentException("Must be negative.");`

② ???

Création d'objets

- ❶ Via *constructeur* : `new` + nom de la classe
 - `Random r = new Random();`
 - `IllegalArgumentException exc1 = new
IllegalArgumentException("Must be positive.");`
 - `IllegalArgumentException exc2 = new
IllegalArgumentException("Must be negative.");`
- ❷ Via une méthode
 - `Instant now = Instant.now();`
 - `String ten = String.valueOf(10);`
- ❸ Via une variable statique : `DateTimeFormatter formatter
= DateTimeFormatter.ISO_DATE`
- ❹ Syntaxe particulière pour instances de `String` : `String s =
"Coucou";`

Usage des objets

- Classes définissent des méthodes
- `String s1 = "a";`
- `String s2 = "Coucou";`
- Différents objets, mais de même type
- Méthodes non statiques ont un comportement différent en fonction de l'instance
- `s1.toUpperCase() ⇒ "A"`
- `s2.toUpperCase() ⇒ "COUCOU"`
- `e1.getMessage() ⇒ "Must be positive."`
- `e2.getMessage() ⇒ "Must be negative."`

Égalité

Attention au test d'égalité :

- ✓ utiliser `s1.equals(s2)` (teste l'égalité sémantique)
- ✗ et *pas* `s1 == s2` (teste l'égalité des références)

Scanner

- `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.