D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER
Université de Toulouse
Faculté Sciences et Ingénierie

Programmation Orientée Objet

Utilisation des objets

Mathieu RAYNAL

mathieu.raynal@irit.fr

http://www.irit.fr/~Mathieu.Raynal

Où peut-on utiliser un objet ?

- Partout !
- On peut créer et utiliser un objet de la même manière qu'une variable de type int, par exemple
- On peut donc utiliser un objet dans une autre classe
- On peut passer un objet en paramètre d'une méthode …
  - … On ne passe pas un objet, mais la référence vers cet objet

Exercice 1 - La classe Point

- Créer une classe Point qui contient
  - 2 attributs
    - Les coordonnées x et y sous forme d'entiers
  - 2 constructeurs
    - Un par défaut
    - Un avec deux arguments : deux entiers pour x et y
  - 1 méthode
    - Déplacer avec 2 entiers en argument qui représente le déplacement en x et en y à effectuer

Exercice 2 – Utilisation de la classe Point dans Rectangle

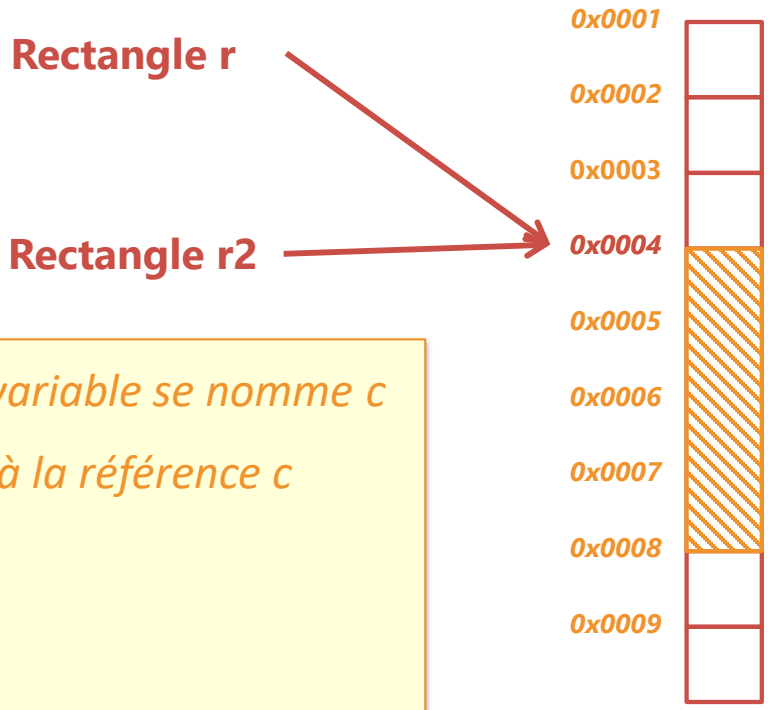
- Dans la classe Rectangle
  - Remplacer les attributs x et y par un attribut de type Point qui correspond aux coordonnées du centre du rectangle
  - Créer une méthode qui teste si un Point appartient au Rectangle
    - La méthode prend en argument un Point
    - La méthode renvoie un booléen
      - Vrai si le point est contenu dans le rectangle
      - Faux sinon

N'oubliez pas d'associer un objet à une variable …

- **Attention :** La déclaration d'une variable ne le relie à aucun objet par défaut
- Si une référence vers un objet n'est associée à aucun objet, elle aura comme valeur : **null**
- Si vous essayez d'utiliser une variable sans lui avoir affecté la référence vers un objet, une exception de type **NullPointerException** sera levée

Affectation d'une référence à une variable

- Le signe = affecte la référence vers un objet et non pas l'objet en lui-même




```
Rectangle r;            // Déclaration d'un objet : la variable se nomme c
r = new Rectangle();    // Création de l'objet associé à la référence c
Rectangle r2;
r2 = r;
r2.x = 15;
```

→ Quelle valeur contient r.x ?

Constructeur par recopie

- Le constructeur par recopie consiste à créer un objet à partir des valeurs d'un autre objet du même type, passé en paramètre du constructeur

```
public class Personne
{
    String nom, prenom;

    public Personne(){…}

    public Personne(Personne p)
    {
        this.nom=p.nom;
        this.prenom=p.prenom;
    }
}
```

Tester des variables pointant vers des objets

- **Attention** : == ou != compare les références et non pas les objets

- Chaque objet contient une méthode equals pour comparer si le contenu de deux objets est identique
  - A vous de la définir pour vos propres classes

instanceof

- Permet de tester le type de classe d'un objet

```
boolean test;
String str = « toto »
test = str instanceof String
```

- ***null*** n'est pas typé
  - instanceof renvoie toujours false pour une référence vers **null**

Exercice 3 - Compléter les classes Rectangle et Point

- Ajouter un constructeur par recopie dans les 2 classes

- Dans la classe Rectangle
  - Créer une méthode qui teste si un Rectangle est équivalent à un autre
    - Equivalent = tous les attributs caractérisant le rectangle sont égaux
    - La méthode prend en argument un Rectangle
    - La méthode renvoie un booléen
      - Vrai si les rectangles sont équivalents
      - Faux sinon