Informatique 1

7. Classes et objets

Dr Pierre-André Mudry
pierre-andre.mudry@hevs.ch

Rev 1.69

Un exemple de code

Objectifs

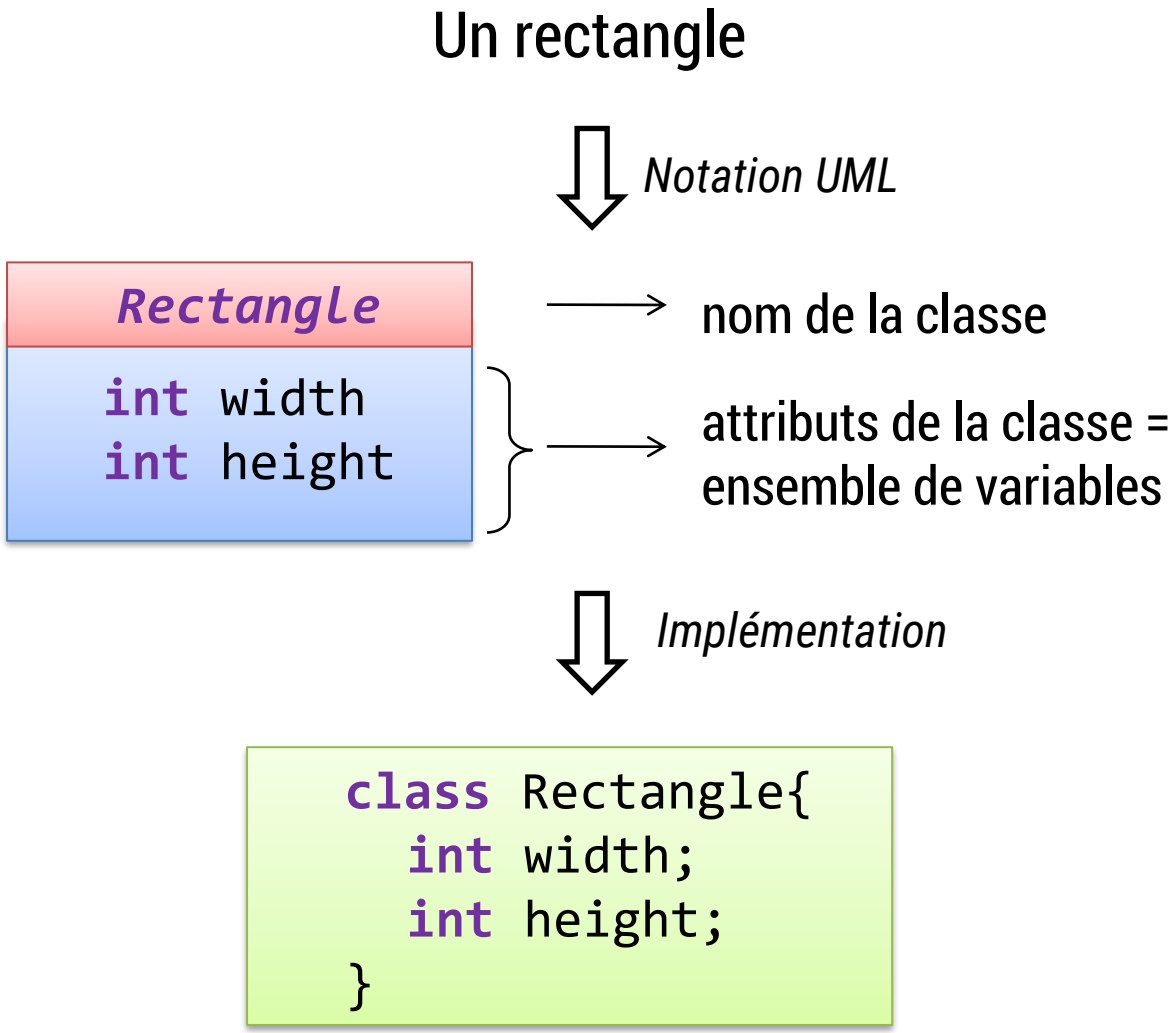
Découvrir classes et objets

- 7.1 Classes et objets
- 7.2 Constructeurs
- 7.3 Visibilité des membres
- 7.4 Modificateur `static`
- 7.5 Autoboxing des types

Commençons par le début

7.1 CLASSES ET OBJETS

Créer une classe : 1er exemple



Définition de classe

Une classe est un **type généralisé** que l'on peut **définir nous-même**.

Définition

Attributs

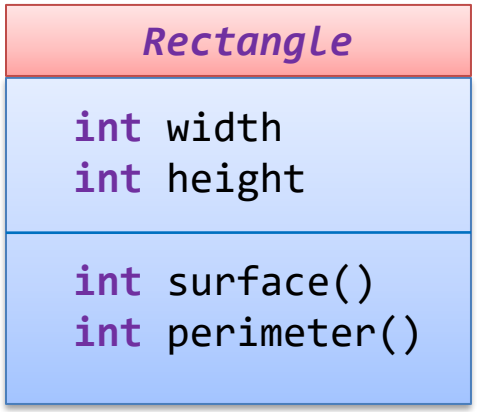
- Variables internes

Méthodes

- Fonctions en relation avec la classe

interface de la classe

Implémentation



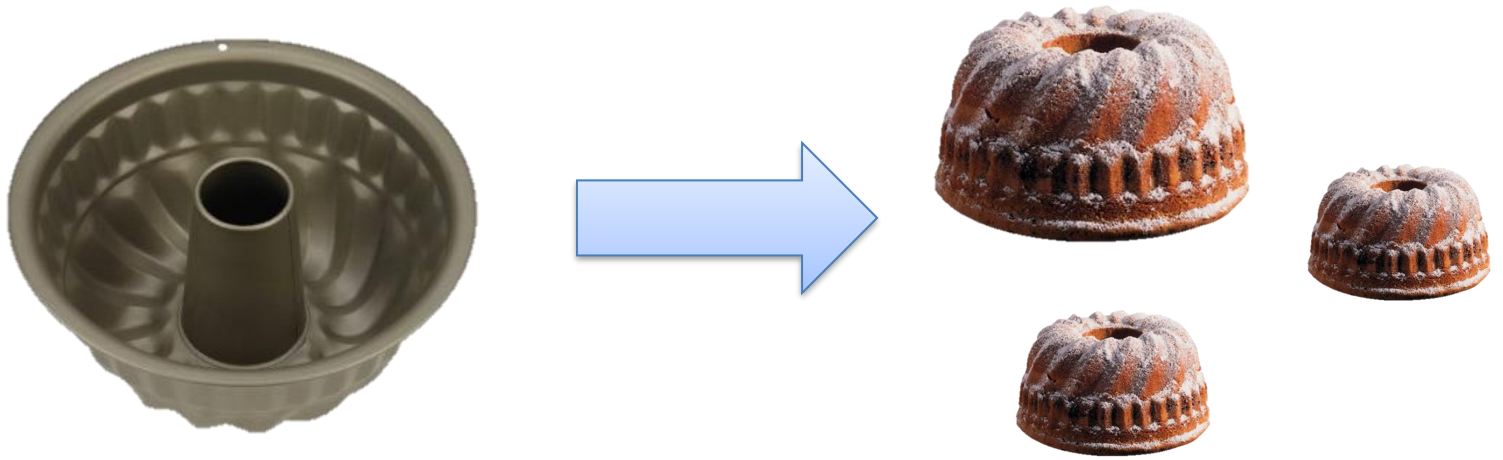

```
public class Rectangle{
   int width;
   int height;

   int surface(){
      return width * height;
   }

   int perimeter(){
      return 2*width + 2*height;
   }
}
```

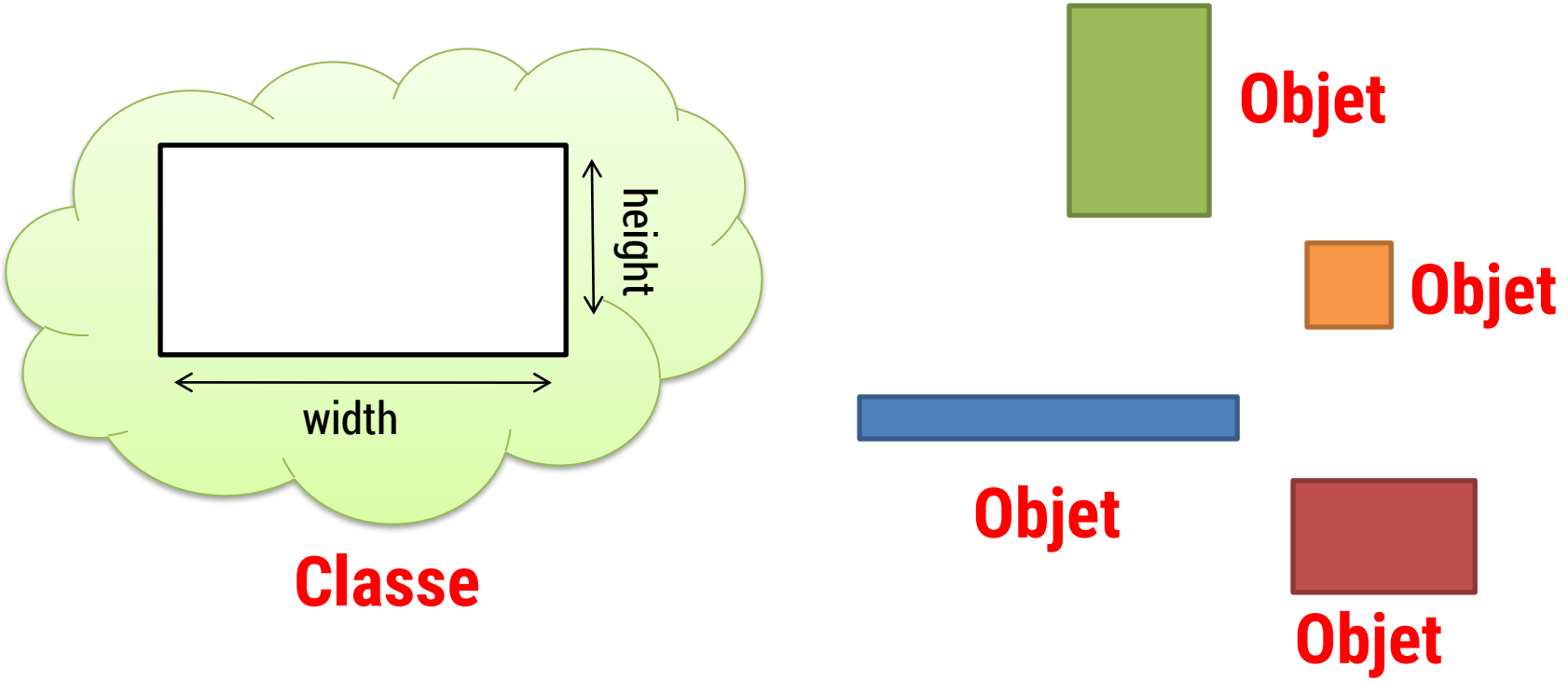
Un peu de vocabulaire…

- Classe = moule
  - Une fois moulé (on dit *instancié*), l'objet possède attributs et méthodes du moule.
  - L'instance de la classe s'appelle un *objet* .
  - Si change le moule, instances différentes.

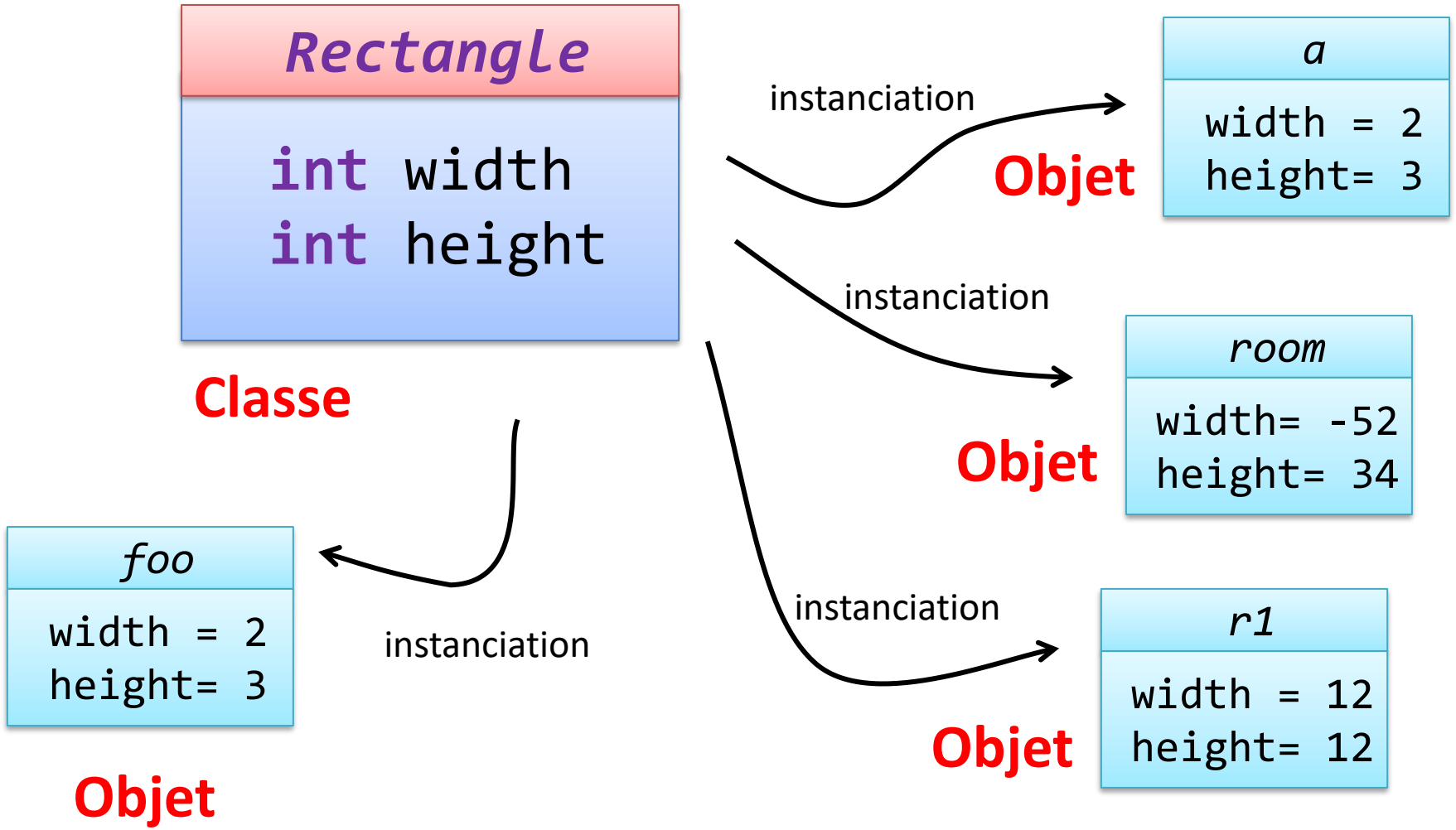


Classes et objets (1)

- Classe `Rectangle` = le modèle de rectangle
- Objets = des rectangles particuliers



Classes et objets (2)



Instanciation (I)

- Classe = type abstrait → création objets comme variables.
- Syntaxe similaire

`className identifier`

Exemple :

```
Rectangle aRectangle;
Rectangle bathRoom;
```

Instanciation (II) – Opérateur **new**

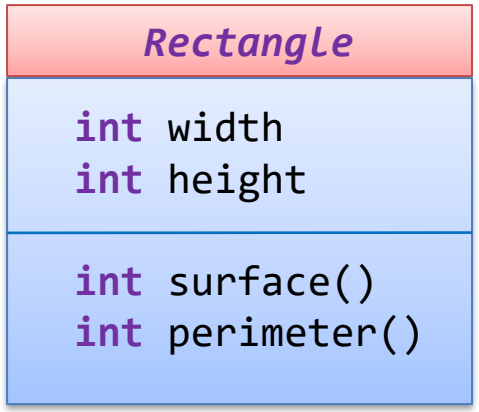
- Une fois le type de l'objet prévu, il faut encore le créer.
- La création de l'objet lui-même se nomme l'*instanciation*.
- Pour réserver la place mémoire, on utilise l'opérateur **new**

```
Rectangle room;
room = new Rectangle();
```

← déclaration
← instanciation

Accès aux membres

- Pour accéder aux membres → opérateur '.'



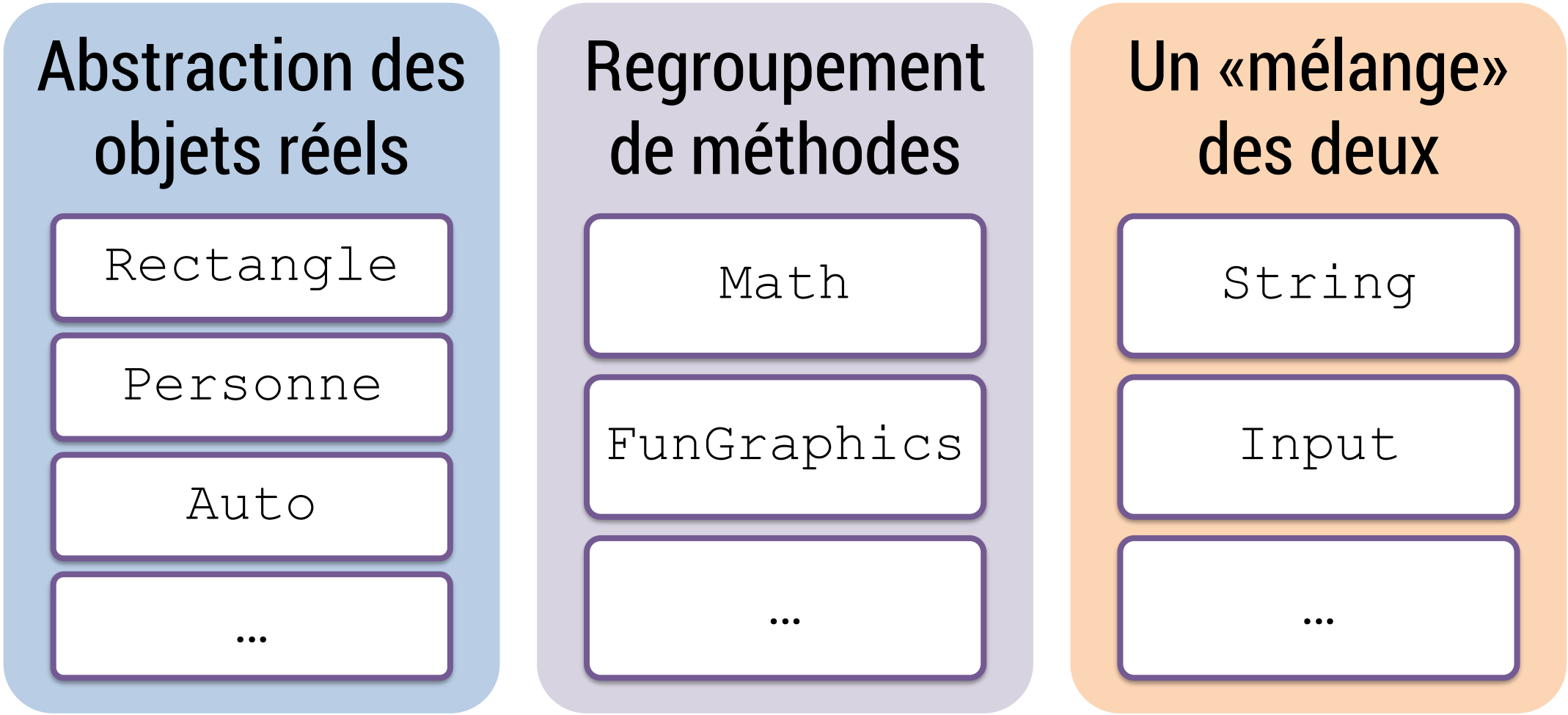

```
Rectangle room;
room = new Rectangle();

room.witdh = 10;
room.height = 4;
int s = room.surface();
```

Les classes

- Variable = tiroir avec un nom et taille.
- Une classe = une armoire modulaire
  - Peut contenir des *variables*
    - Nombre pas limité
    - Type quelconque (également d'autres classes)
  - Peut contenir des *méthodes*

Usage des classes



- Origine: le langage, d'autres codeurs, nous

Rappel (cours 3)

Pour l'instant

- **Une classe = un type**
- **Un objet = une variable de ce type**

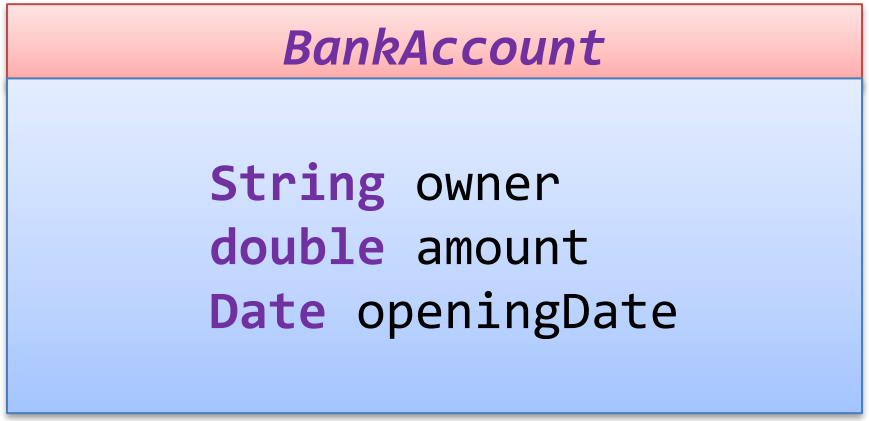
- *Exemple* : la classe **`String`** sert à contenir des chaînes de caractères

```
String name = "Henri";
```

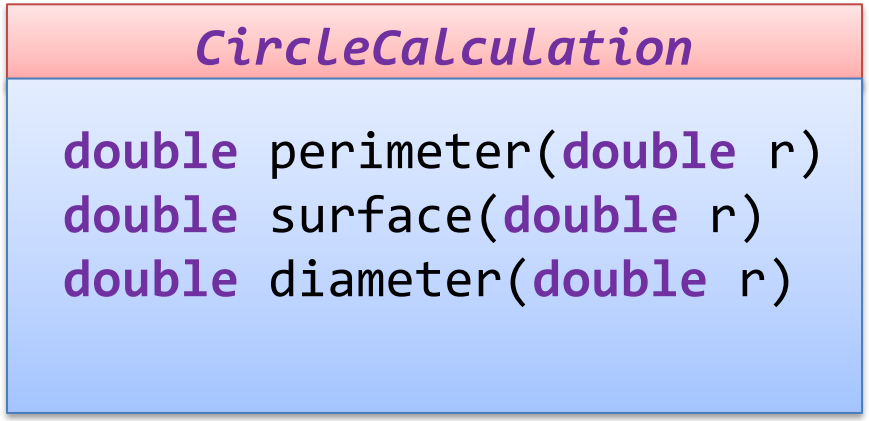
Classe — Objet — Val. initiale

Des classes multi-usages

Exemple 1 : variables seules



Exemple 2 : méthodes seules



Exemple 3 :

Exemple 4 :

Deux conventions importantes

- Nom des classes : doit toujours commencer par des majuscules
- Valeurs par défaut : si font du sens
  - Attributs avec un type primitif (`int`, `char`, etc…) →initialisés automatiquement à 0
  - Attributs qui sont des objets → initialisés à `null`