DUT MMI – IUT de Marne-la-Vallée
31/03/2015
M2202 - Algorithmique

Cours 5
Programmation objets : les principes

UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

iut Marne-la-Vallée

Philippe Gambette

Sources

- Cours d'Anne Tasso à l'IUT de Marne-la-Vallée
- *Le livre de Java premier langage*, d'A. Tasso

Plan du cours 5 – Objets

- Concept de la programmation orientée objets
- Protection des données et encapsulation
- Syntaxe Java

Plan du cours 5 – Objets

- Concept de la programmation orientée objets
- Protection des données et encapsulation
- Syntaxe Java

Concept de la programmation orientée objets

Objectif : modéliser des objets informatiques

- décrire leurs **caractéristiques**

- décrire leurs **comportements**

types d'objets Java que le programmeur peut définir

Avantages :

- code modulaire, décomposé en “classes” d'objets

- code sécurisé, possible de réutiliser certains objets dans d'autres programmes

→ simplifie la vie du programmeur

Exemples : - les actions Javascript sur les composants d'une page web

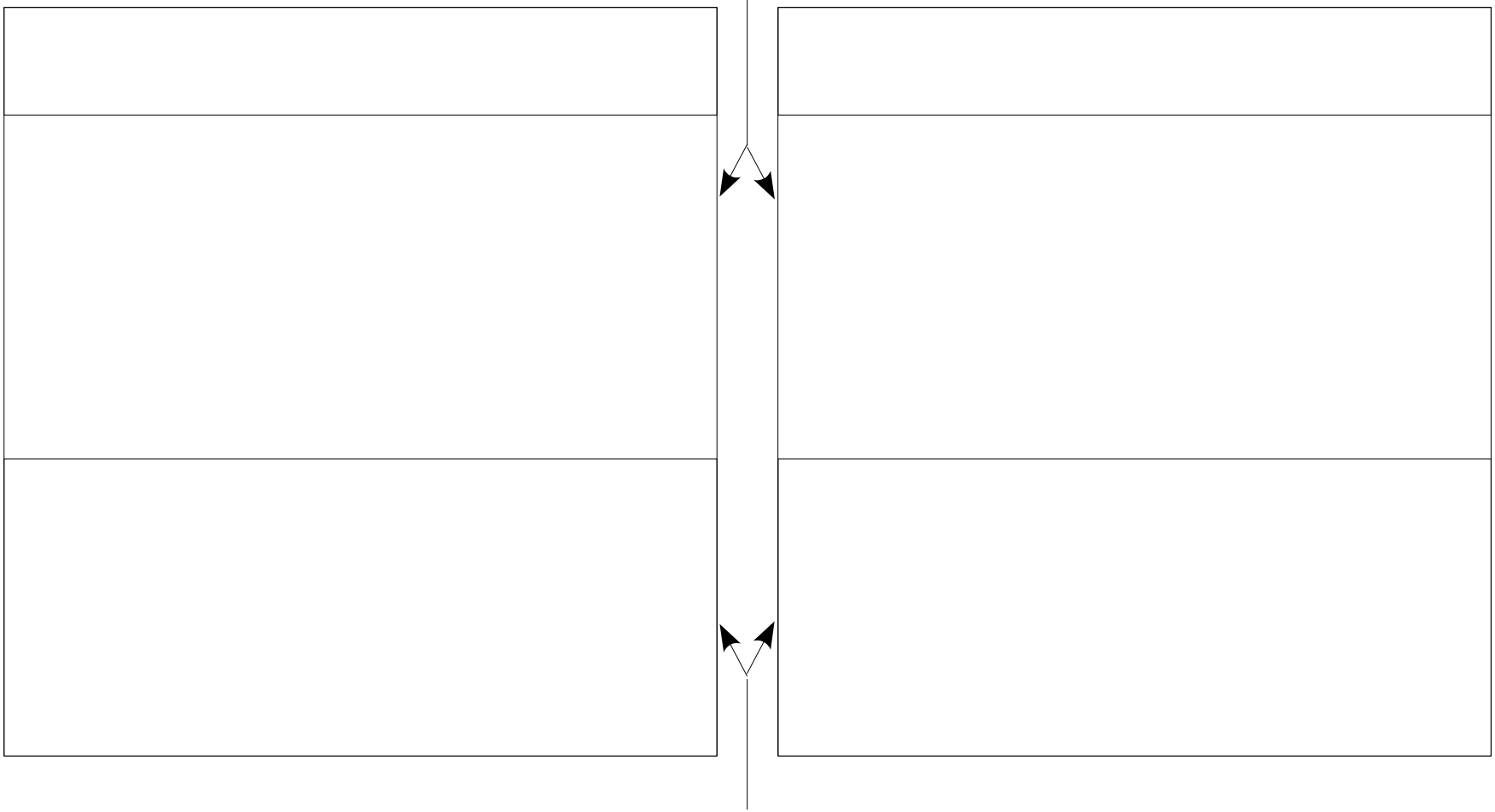
- la gestion des notes d'une classe

Modélisation objets

Quelles caractéristiques / attributs ? **“propriétés”**

Composants d'une page web

Etudiants d'une classe



Quels comportements / actions ? **“méthodes”**

Modélisation objets

Quelles caractéristiques / attributs ? **“propriétés”**

Composants d'une page web

Composant_de_page_web
type : le nom de balise attributs CSS : **width**, **margin**, etc. attributs HTML : **href**, **alt**, etc. **contenu** : le code entre la balise ouvrante et la balise fermante
affiche, **cache**, **changeCouleur**, etc.

Etudiants d'une classe

Etudiant
String **nom** String **prenom** int **id** String **typePromo** int[] **notes**
calculeMoyenne, **creeBulletin**, **afficheInfos**, **donneNote1**, etc.

Quels comportements / actions ? **“méthodes”**

Modélisation objets

La différence entre un objet et une classe ?

Modélisation objets

La différence entre l'objet et sa classe est la même qu'entre :

• qu'entre une **variable** et son **type**

• qu'entre une **occurrence** et son **entité** (en bases de données)

• qu'entre la **ligne de données d'une table**, et la **classe** qui correspond à la table (en bases de données).

Ainsi, l'objet est un élément spécifique d'une classe.

Utiliser la variable `etudiant1` comme un objet de la classe `Etudiant` :

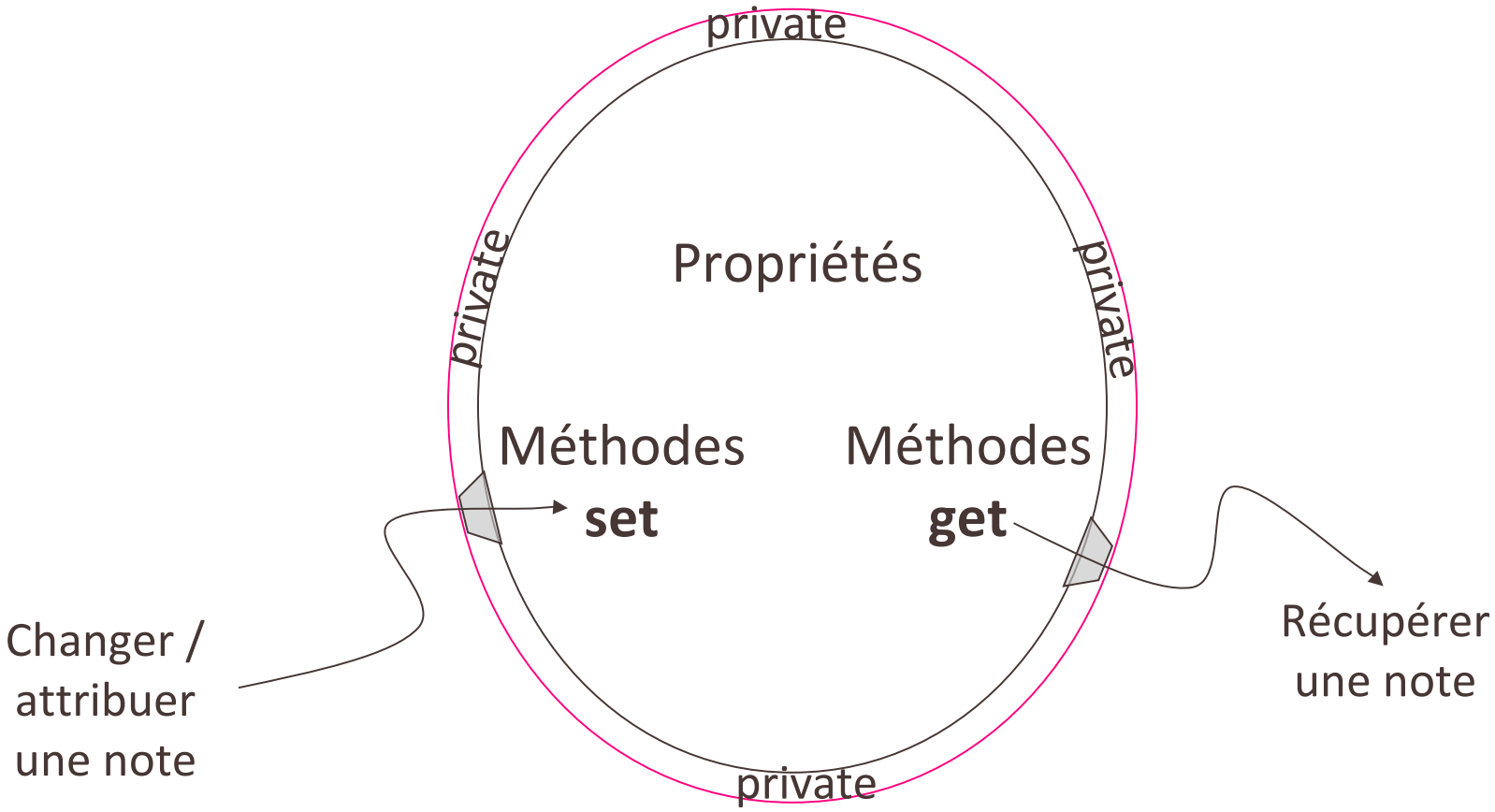
```
//Déclaration (nom de variable précédé de son type) :
Etudiant etudiant1;

//Utilisation de propriétés de la classe Etudiant :
System.out.println(etudiant1.prenom+" "+etudiant1.nom);

//Appel de méthodes de la classe Etudiant :
int moyenne = etudiant1.calculeMoyenne();
```

Plan du cours 5 – Objets

- Concept de la programmation orientée objets
- Protection des données et encapsulation
- Syntaxe Java

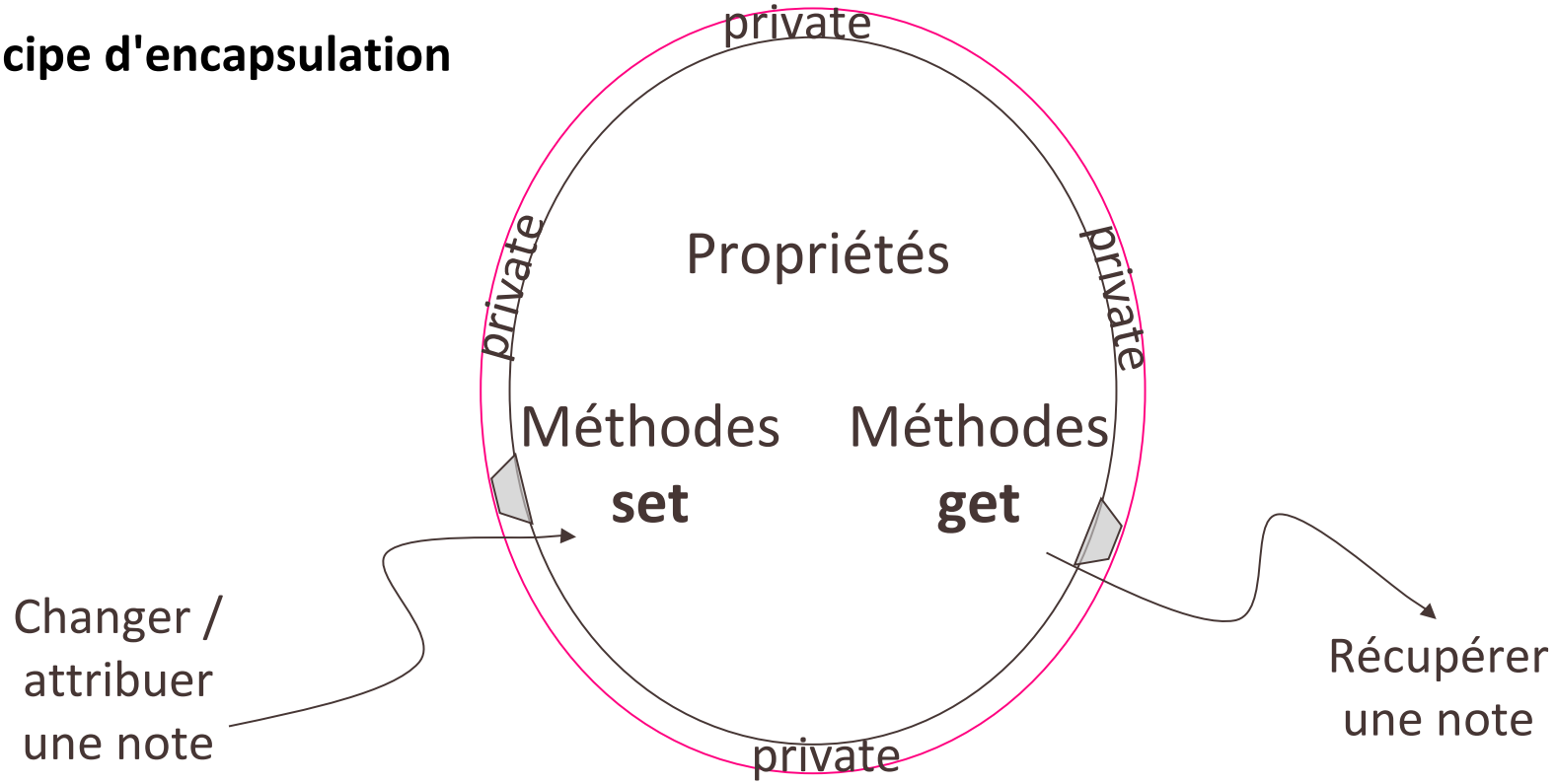
Protection des données

Principe d'encapsulation



Protection des données

Principe d'encapsulation



Utilisation d'objets de la classe `Etudiant` depuis la classe `GestionEtudiants` :

```
public static void entreNote1(String nomEtudiant){
  etudiant Etudiant1 = trouveEtudiantParNom(nomEtudiant);
  //sans encapsulation (public)
  etudiant1.note1 = 12;
  System.out.println(
    etudiant1.note1);
}
```

```
//avec encapsulation (private)
etudiant1.setNote1(12);
System.out.println(
  etudiant1.getNote1());
}
```

Protection des données

Constructeur

Sert à créer un objet

→ fournir une valeur à ses propriétés (initialisation)

Constructeur par défaut :

→ initialise les entiers à 1, les flottants à 0.0, les chaînes de caractères à Null.

Pour changer les valeurs des propriétés d'un objet par la suite :

- les modifier depuis une autre classe ?
- les modifier depuis une méthode de la classe de l'objet ?

Protection des données

Différents niveaux de protection (pour les propriétés et pour les méthodes)

→ public : accessible pour tous les objets de l'application

→ private : accessible que pour les méthodes de la même classe

→ protected : accessible que depuis les méthodes de la même classe ou d'une sous-classe

Partage d'une donnée entre objets d'une même classe :

→ static

→ sinon les données sont spécifique à un objet donné de la classe

Similaire à variable globale / variables locales, au niveau des objets plutôt que des fonctions

Plan du cours 5 – Objets

- Concept de la programmation orientée objets
- Protection des données et encapsulation
- Syntaxe Java

Syntaxe Java

Utiliser une classe depuis une autre (exemples de TP précédents)

```
public static Color couleurRGB(int r,int g,int b){

  return new Color(r,g,b);

}
```

```
public static void dessineRectanglePlein(Graphics g,
int abscisseCoin,int ordonneeCoin,int largeur, int hauteur,
Color couleur) {

  g.setColor(couleur);

  g.fillRect(abscisseCoin, ordonneeCoin, largeur,
  hauteur);

}
```