D[i] | Département Informatique

UNIVERSITÉ TOULOUSE III PAUL SABATIER — Université de Toulouse

Faculté Sciences et Ingénierie

Programmation Orientée Objet

Mathieu RAYNAL

mathieu.raynal@irit.fr

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

D[i] | Département Informatique

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

Programmation Orientée Objet

Structurer avec des packages

Visibilité d'une classe et de ses membres

Des membres particuliers

Déclarer une classe dans un package

- Utilité d'un package
  - Regroupe un ensemble de classes sous un même espace de nommage.
  - Permet de différencier 2 classes qui porteraient le même nom dans 2 projets différents

- Déclaration d'un package
  - C'est la première instruction du fichier
  - Avant la déclaration de classe
  - L'instruction package indique à quel package appartient la classe.

```
package tp;
```

Déclarer une classe dans un package

- Noms des packages en minuscules
- On peut réaliser des « sous packages »
  - nom1.nom2.nom3…

```
package fr.univ-tlse3.bioinfo;
```

- Par défaut une classe appartient au package anonyme

Accès aux classes hors de leur package

- Hors d'un package, le nom des classes contenues dans celui-ci est :
  - Soit **nompackage.NomClasse**
  - Soit
    - avant la déclaration de la classe, on déclare **import nompackage.NomClasse;**
    - On peut ensuite utiliser le nom de la classe sans le préfixer du nom du package
- import nompackage.*; permet d'utiliser toutes les classes du package
- Pour accéder au package par défaut dans un autre fichier, il faut faire **import** *;

D[i] | Département Informatique

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

Programmation Orientée Objet

Structurer avec des packages

Visibilité d'une classe et de ses membres

Des membres particuliers

La visibilité d'une classe ou d'un élément

- Une classe peut
  - Être publique *(public)*
  - Avoir la visibilité par défaut *(rien de marqué)*

- Un membre d'une classe peut
  - Être privé *(private)*
  - Avoir la visibilité par défaut *(rien de marqué)*
  - Être protégé *(protected)*
  - Être publique *(public)*

- Ce mot clé (public, private ou protected) se place
  - Devant le type pour un attribut
  - Devant le type de retour pour une méthode

La visibilité d'une classe ou d'un élément

- Classé par ordre de restriction décroissante :
  - private
    - visible que depuis sa propre classe
    - Même pas visible dans les sous-classes
  - Par défaut *(aucun mot clé utilisé)*
    - Visible que dans son package
  - protected
    - Visible à l'intérieur de son package
    - A l'extérieur de son package, visible que dans les sous classes de sa classe
  - public
    - Visible partout

Visibilité des attributs

- La valeur d'un attribut peut avoir une incidence sur d'autres attributs
  - Par exemple, la longueur d'un rectangle a une incidence directe sur le périmètre et la surface.

- Il est donc préférable de mettre les attributs d'une classe en **private** pour qu'ils ne puissent pas être modifier n'importe comment et n'importe où

Accès aux attributs d'une classe

- Si un attribut est marqué **private**, il est de coutume de créer des méthodes dites
  - **getter** pour connaitre la valeur de cet attribut
  - **setter** pour modifier la valeur de cet attribut

```
public class MaClass{
  private int somme;

  public void setSomme(int s){
     somme = s;
     …
  }

  public int getSomme(){
     return somme;
  }
}
```

Exercice

- Dans la classe Rectangle
  - Ajouter des attributs privés ***surface*** et ***perimetre***
  - Modifier les méthodes calculerSurface et calculerPerimetre pour mettre à jour ces 2 attributs
  - Mettre en privé les attributs longueur et hauteur
  - Créer les getters et setters de ces attributs de manière à manipuler correctement les rectangles
  - Créer également les getters pour surface et perimetre

D[i] | Département Informatique

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

Programmation Orientée Objet

Structurer avec des packages

Visibilité d'une classe et de ses membres

Des membres particuliers

Des membres **static**

- Un membre static est commun à toutes les instances de la classe
  - Le mot static se place avant le type de l'attribut ou avant le type de retour d'une méthode
- Pas besoin d'instancier un objet pour les utiliser
- Pour les utiliser : nom de la classe . nom du membre
- La classe **Math** n'a que des membres statiques

Exercice

- Ajoutez un attribut static nb de type entier à la classe Rectangle
- Incrémentez cet attribut à chaque création d'un nouvel objet Rectangle
- Si le nom du Rectangle est donné par défaut, créez le nom de la forme suivante : « R » suivi de la valeur de nb

Des attributs déclarés **final**

- Les attributs précédés de **final** sont initialisés avec une valeur qui ne peut pas être modifiée par la suite

- Souvent couplé au mot clé static, il permet de définir des « constantes »