

Programmation Orientée Objet

Mathieu RAYNAL

mathieu.raynal@irit.fr

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

Programmation Orientée Objet

Enumération

Exception

Créer une énumération

- Se déclare comme une classe
 - Mot clé **enum** à la place de **class**
 - Hérite de la classe **java.lang.Enum**
- Contient une série de valeurs constantes
 - ni le type, ni la valeur réelle de chaque constante n'est précisé.
- Les valeurs sont classées par ordre de déclaration
- Elles ont un indice
 - Commence à la valeur 0 pour la première valeur déclarée

```
public enum TypeObjet
{
    RECTANGLE,
    CARRE,
    TRIANGLE,
    ELLIPSE,
    CERCLE
}
```

Utilisation d'une énumération

- Les valeurs sont comparables avec les opérateurs classiques
- Utilisable avec l'instruction switch

```
TypeObjet t = TypeObjet.CARRE;
```

```
if(t == TypeObjet.CARRE)
{
    // Traitement ...
}
```

```
for(TypeObjet type : TypeObjet.values())
    System.out.println(type);
```

```
switch(t)
{
    case RECTANGLE:
        // Traitement ...
        break;
    case CARRE:
        // Traitement ...
        break;
    ...
}
```

Compléter une énumération

- Les énumérations peuvent avoir des attributs et des méthodes

```
public enum TypeObjet {  
    RECTANGLE("un rectangle"),  
    CARRE("un carré"),  
    TRIANGLE("un triangle"),  
    ELLIPSE("une ellipse"),  
    CERCLE("un cercle");  
  
    private String nom;  
  
    private TypeObjet(String nom){  
        this.nom = nom;  
    }  
  
    public String getNom(){  
        return nom;  
    }  
}
```

Exercice 1 – Utilisation d'une énumération

- Ecrire une classe Personne
 - Deux attributs de type String pour le prénom, et le nom
 - Un attribut de type Genre pour le genre (HOMME|FEMME)
 - Définir l'énumération GENRE
 - Un Constructeur par défaut où les attributs seront remplis par l'utilisateur au moyen de la classe Saisie
 - Proposer les différents choix pour le genre
 - Une méthode affichage qui affiche un message du type
« Bonjour **M.** BARRIOT, vous êtes **le bienvenu** dans cette classe ! »
« Bonjour **Mme** FICHANT, vous êtes **la bienvenue** dans cette classe ! »

Programmation Orientée Objet

Enumération

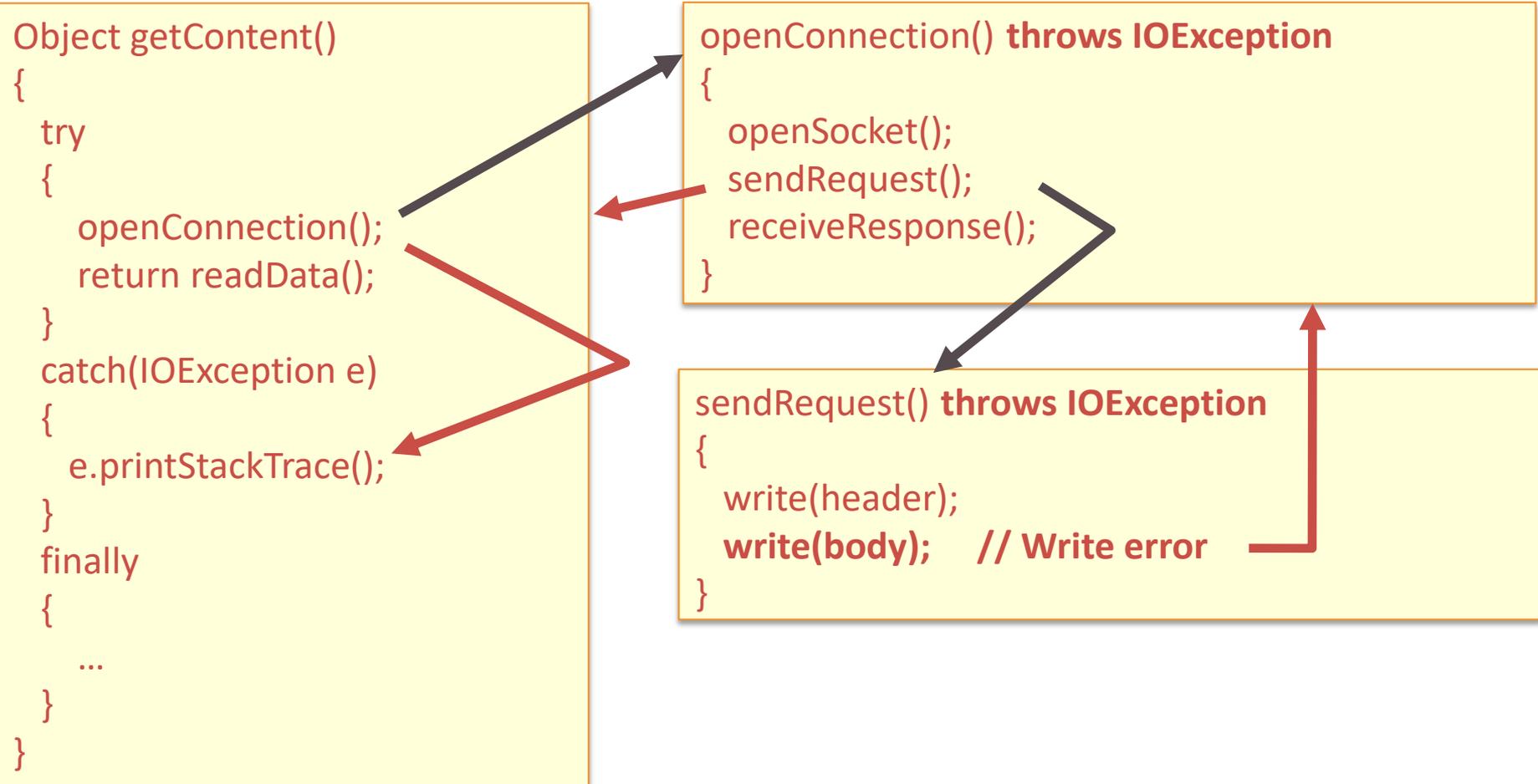
Exception

Gestion des exceptions

- Dans certains cas, une méthode peut renvoyer une exception
 - Permet de séparer le flot de contrôle classique de celui lié à un comportement exceptionnel.
- L'appel à une méthode pouvant lever une exception doit :
 - Soit avoir un bloc ***try/catch*** ;
 - Soit propager ***throws*** ;
- Un bloc ***finally*** peut être posé à la suite d'un bloc ***catch***. Il est toujours exécuté.

Gestion des exceptions

- Lors d'une exception il faut remonter jusqu'à trouver un bloc *try/catch*



Exemple d'affichage d'une exception

- Affichage lors de l'appel à la méthode `printStackTrace()`

java.lang.NullPointerException

```
at diamant.tp.dess.tableau.Tableau.getLine(Tableau.java:231)
at diamant.tp.dess.tableau.Gestionnaire.valueChanged (Gestionnaire.java:139)
at javax.swing.DefaultListSelectionModel.fireValueChanged
    (DefaultListSelectionModel.java:187)
at javax.swing.DefaultListSelectionModel.fireValueChanged
    (DefaultListSelectionModel.java:167)
at javax.swing.DefaultListSelectionModel.fireValueChanged
    (DefaultListSelectionModel.java:214)
at javax.swing.DefaultListSelectionModel.changeSelection
    (DefaultListSelectionModel.java:402)
```

- Dans une méthode main
 - Pour chaque attribut passé en paramètre du programme, affichez le préfixe constitué des 3 premières lettres de la chaîne de caractères
 - Méthode substring de la classe String
 - Attention au cas où la chaîne de caractères fait moins de 3 caractères ...