JAVA
FONDAMENTAUX

Présenté par :

Xavier TABUTEAU

Les exceptions

Les exceptions c'est la gestion des erreurs.
Dans un langage non objet on va traiter les erreurs de cette façon :

```
retour = fonction1(parametre x);

si (retour == erreur) {
        traiter l'erreur;
}
```

En langage objet, non traitons l'erreur via une classe nommée Exception.

- Une exception est un objet instancié de la classe Exception.
- Il faut informer le système de ce qui déclenche la levée d'une exception.
- Il faut savoir récupérer et traiter une exception.

Toutes les exceptions levées dans une méthode et qui n'ont pas été traitées localement, doivent être ajoutées à la signature de la méthode via le mot-clé « throws » qui permet de déléguer la responsabilité des erreurs à la méthode appelante.

Traitements des exceptions

La récupération se fait en encapsulant le code risquant de lever une exception avec les instructions try / catch / finally.

Syntaxe d'un try catch :

```
try {
        instructions pouvant provoquer une exception;
}
catch (Exception1 e) {
        instructions si une exception est survenue.
}
catch (Exception2 e) {
        instructions si une exception est survenue.
}
...
finally {
        instruction à exécuter qu'une exception soit survenue ou non.
}
```

Remarque : le finally est très utile pour libérer des ressources (ex : mémoires ou fichiers).

Traitements des exceptions

Une alternative à l'utilisation de plusieurs blocs « catch » consiste à utiliser instanceof :

Syntaxe avec instanceof :

```
try {
        instructions pouvant provoquer une exception;
}
catch (Exception e) {
        if (e instanceof Exception1)
                instructions si Exception1 est survenue.
        if (e instanceof Exception2)
                instructions si Exception2 est survenue.
}
...
finally {
        instruction à exécuter qu'une exception soit survenue ou non.
}
```

Traitements des exceptions avec ressource

Le traitement d'exception avec ressource permet de libérer automatiquement la ressource utilisée à la fin du « try / catch ». Il n'est pas nécessaire d'avoir un bloc finally pour libérer la ressource. La ressource s'indique à la ligne du try comme une création d'objet classique. Cet objet disparaîtra à la fin du « try / catch ».

Syntaxe d'un try avec ressources :

```
try (<type> obj = new UneClasse(<type> param, ...) {
        instructions pouvant provoquer une exception;
}
catch (Exception1 e) {
                instructions si Exception1 est survenue.
}
…
```

Levée des exceptions

La signature d'une méthode doit inclure la déclaration des diverses exceptions levées dans le code qui ne sont pas traitées localement, et ceci se réalise avec l'instruction « throws ».

Syntaxe d'une levée d'exceptions :

```
<Visibilité> <Type> nomMethode(<Type> param) throws Exception1, Exception2 {
        instructions;
        if (condition 1) {
                throw new Exception1("Message de l'exception 1");
        }
        if (condition 2) {
                throw new Exception2("Message de l'exception 2");
        }
        ...
}
```

Exceptions personnalisées

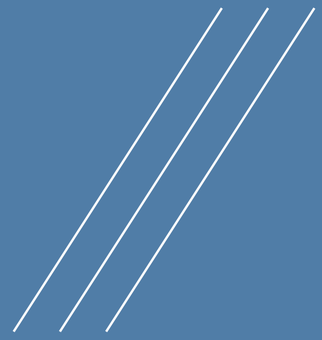
Pour des besoins applicatifs spécifiques, il est possible de créer soit même sa gestion d'exceptions en Java.

Les classes d'exceptions personnalisées doivent hériter de la classe Exception, ou d'une classe qui hérite de cette classe.

La gestion de ces exceptions est identique aux autres.

ExceptionPerso.java

JAVA
FONDAMENTAUX



Présenté par
Xavier TABUTEAU