Ecrire la classe d'une exception dite métier

- Une exception métier (custom exception) dérive de la classe Exception
- Cela implique qu'elle doit définir les constructeurs qui prennent un message en argument et,optionnellement, un objet Throwable
- On peut aussi définir le constructeur sans paramètre qui appelle le constructeur avec message
- Éventuellement, redéfinir la méthode public String getMessage() qui retourne le message passé en argument de ce constructeur

Exemple

```
public class SoldeInsuffisantException extends
Exception {
   // Constructeur avec un message en argument
   public SoldeInsuffisantException( String message) {
    // Appel du constructeur de Exception
    super( message);
 }
}
```

Remarque sur les exceptions non-contrôlées

- Il est possible de définir une exception de type RunTime, c'est à dire dont le traitement par l'appelant n'est pas obligatoire
- Il faut définir une classe d'exception métier qui hérite de la classe RunTimeException
- Il ne faut surtout pas déclarer cette exception dans la clause throws de la méthode susceptible de la lancer

Propagation des exceptions

- Un aspect intéressant des exceptions est la possibilité de les chaîner
- Le principe repose sur l'idée de ne pas traiter une exception, mais de la faire remonter à l'appelant

```
public versement( long montant, Compte
beneficiaire) throws SoldeInsuffisantException {
  try {
    debit( montant);
    beneficiaire.crediter( montant);
  } catch( SoldeInsuffisantException e) {
    throw e; }}
```


Propagation automatique

- En Java, toute exception non "catchée" sera automatiquement remontée à l'appelant
- L'exemple précédent peut devenir plus simplement :

```
public versement( long montant, Compte beneficiaire) throws
SoldeInsuffisantException {
   debit( montant);
   beneficiaire.crediter( montant);
}
```

La pile d'appel (StackTrace)

- On lève une exception
  - if(...)
    - throw new SoldeInsuffisantException(« aa »);
- Lorsque l'exception est levée, une « photographie » est prise sur la succession des appels qui ont mené à cette instruction throw
- C'est ce que l'on appelle la pile d'appels
- Il est possible de l'afficher: printStackTrace
- Elle permet d'identifier l'endroit dans le programme d'où provient l'exception

Stratégie de gestion des exceptions

- Il est possible de choisir sa stratégie, en gérant par exemple les exceptions au coup par coup, ou au contraire en ne gérant rien et en propageant vers l'appelant du plus haut niveau

Escalade des exceptions

- Elle suit la logique de spécialisation.
- Les exceptions de bas niveau sont renvoyées sous la forme d'exceptions métier de haut niveau.
- L'avantage est que l'on va cacher l'implémentation de bas niveau, qui est susceptible de changer, sans pour autant remettre en question les traitements de haut niveau (par exemple, passage d'un mécanisme de base de données vers un autre).

END