Exceptions

Pourquoi : Une méthode ne peut pas effectuer le travail demandé.

* > On lance une exception spécifique au probleme
* > Voir la JavaDoc pour des exemples

Object - > Throwable - > Exception - > IOException

- > RunTimeException

Pour lancer une exception on utilise le Throw new Exception

Pour attraper une exception on utilise le bloc try, catch()

Quand une méthode est susceptible de lancer une exception, on le spécifie dans l’en-tête de la méthode. Exemple : public String getString() throws new MathException

Les unchecked exceptions > pas d’obligation de les déclarer, ni de les attraper

Exemples : NullPointerException, IllegalArgumentException

Les checked exceptions > déclaration et traitement obligatoire

Exemples : Tout le reste

Une image contenant texte

Description générée automatiquement

Méthodes d’une exception :

* String getMessage() : Envoie le message de l’erreur
* printStackTrace() : Trace des exceptions

Quand lancer une exception ? :

* Il ne faut pas en abuser
* Les utiliser pour éviter les fins brutes
* On les utilise pour signaler un probleme dans une méthode
* On préfère parfois envoyer false ou null lors d’une exception

On peut traiter plusieurs exceptions en même temps : utilisation de plusieurs blocs catch

Bloc finally : instruction executée avant de sortir d’une méthodefinally{...} toujours entièrement exécuté avant de quitter la méthode, sauf si exception lancée dans le finally ouSystem.exit.

Un bloc finally{...} après le dernier bloc catch d’un ensemble try{...}catch(...){...}.