

**Travaux Dirigés/Pratiques :**

**Les exceptions**

**M9-JAVA AVANCEE & J2EE**

**Hamza Jebrane**

**2019/2020**

**Introduction :**

Une exception est une erreur se produisant dans un programme qui conduit le plus souvent à l'arrêt de celui-ci. Il vous est sûrement déjà arrivé d'obtenir un gros message affiché en rouge dans la console d'Eclipse : eh bien, cela a été généré par une exception… qui n'a pas été capturée.

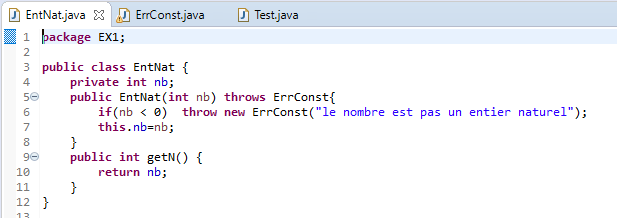
Le fait de gérer les exceptions s'appelle aussi « la capture d'exception ». Le principe consiste à repérer un morceau de code (par exemple, une division par zéro) qui pourrait générer une exception, de capturer l'exception correspondante et enfin de la traiter, c'est-à-dire d'afficher un message personnalisé et de continuer l'exécution.

**Exercice 1 :**

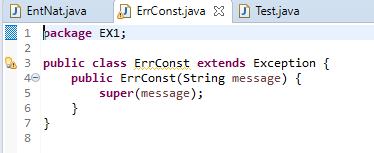
*EntNat* permettant de manipuler des entiers naturels (positifs

Ou nuls) *EntNat* disposera de :

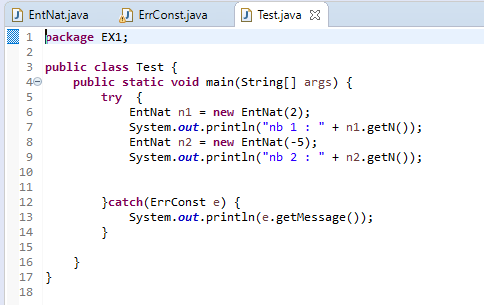
* Un constructeur à un argument de type int qui générera une exception de type *ErrConst*
* Une méthode getN fournissant sous forme d’un int, la valeur encapsulée dans un objet de type *EntNat*.

****

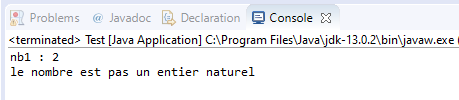
*ErrConst* une classe qui héritant de la classe Exception

****

Classe Test pour tester notre programme

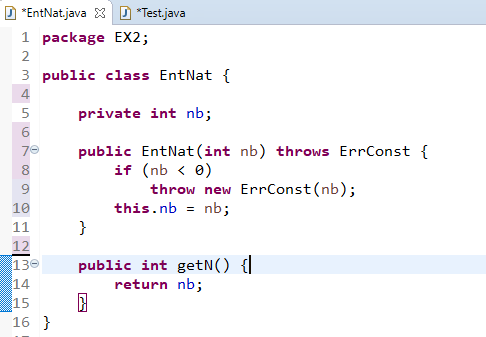
****

Exécution de Test

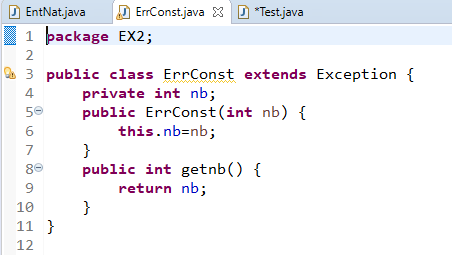


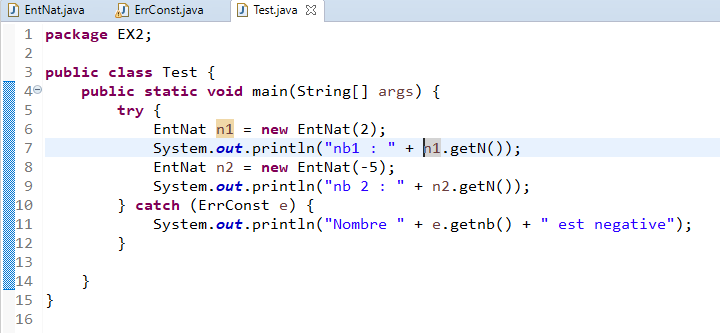
**Exercice 2 :**

Adapter la classe *EntNat* de l’exercice 1 et le programme d’utilisation de manière à disposer dans le gestionnaire d’exception du type ***ErrConst*** de la valeur fournie à tort au constructeur.

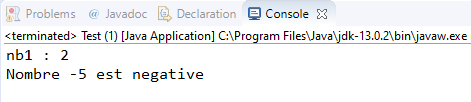
****La classe *EntNat :*

La classe *ErrConst :*

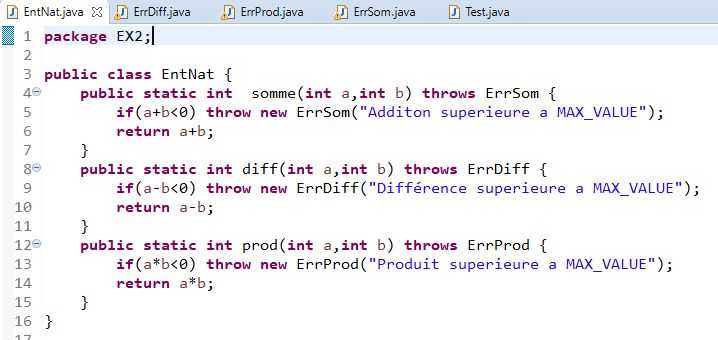
****

****Classe Test pour tester notre programme

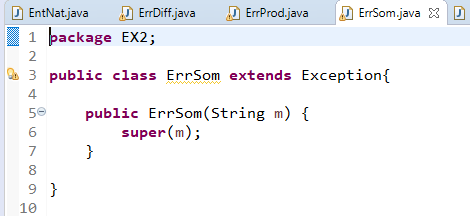
Exécution de Test

****

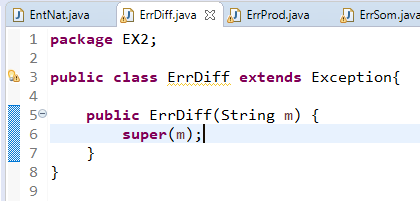
**Problème : (Synthèse du chapitre) :**

****La classe *EntNat :*

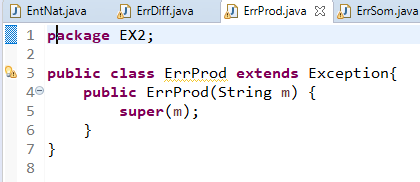
La classe *ErrSom :*

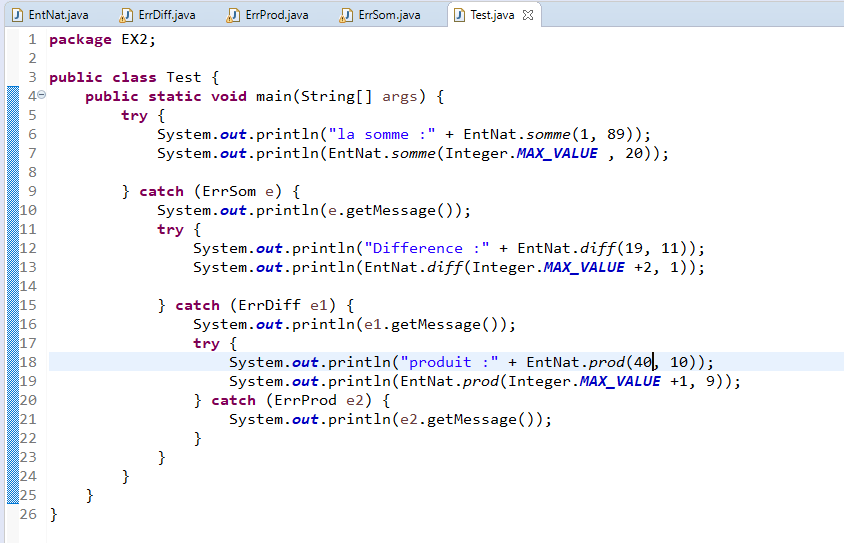
**

La classe *ErrDiff :*

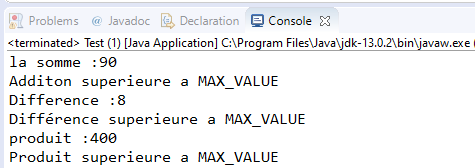
****

La classe *ErrProd :*

**

Classe Test pour tester notre programme

Exécution de Test

****