|  |
| --- |
| isil |
| RAPPORT #1 |
| TP EXCEPTION |
|  |
| **BOUTAKOURT Youssef** |
| **2019/2020** |

**Encadré par :**

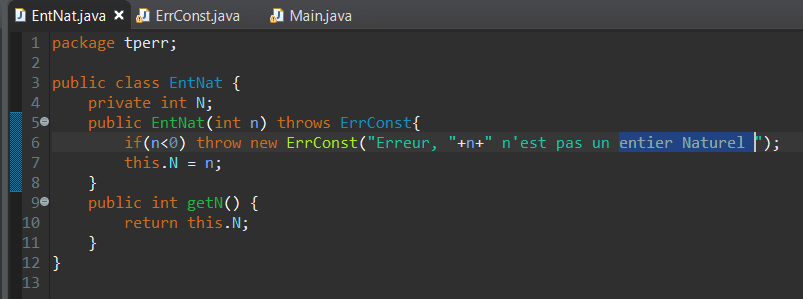
**Mr. KARAMI Fahd**

INTRODUCTION

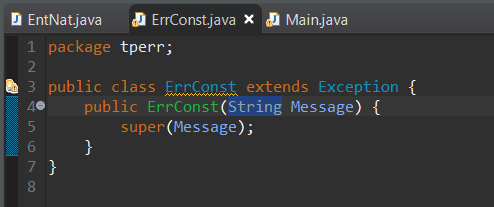
Dans le cadre d’appliquer les connaissances acquises lors les séances de cours, j’ai réalisé un ensemble d’exercices TP.

## Exercice 1 : Déclenchement et Traitement d’une exception

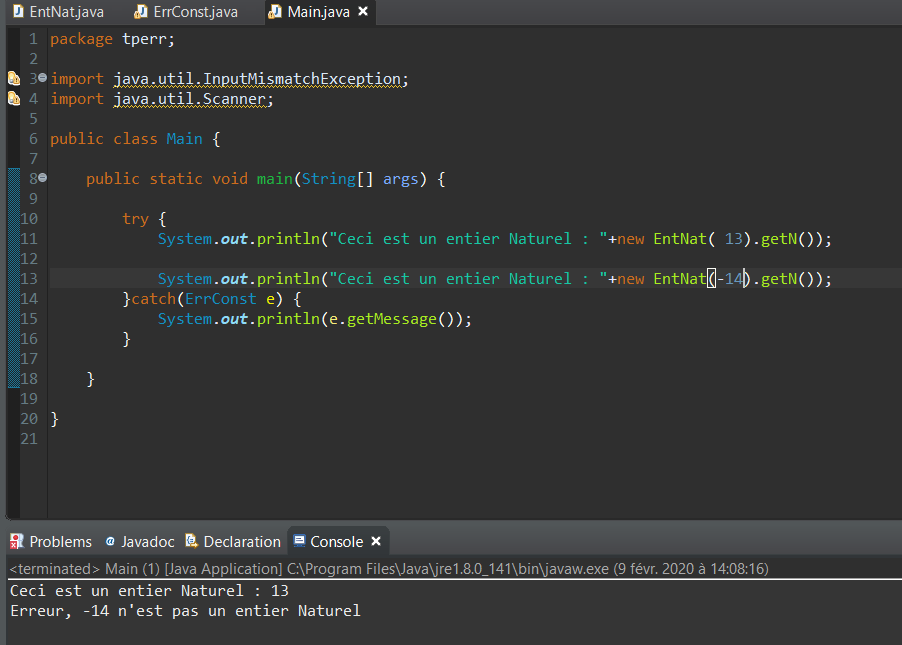
La classe EntNat : j’ai créé une exception avec message



La Classe ErrConst : j’ai créé une classe qui dérive d’une class exception avec un message comme paramètre.

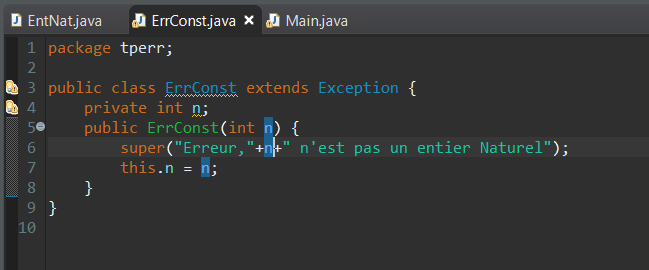


Et Finalement la classe test, l’entier naturel ne déclenche pas l’exception qu’on a géré.

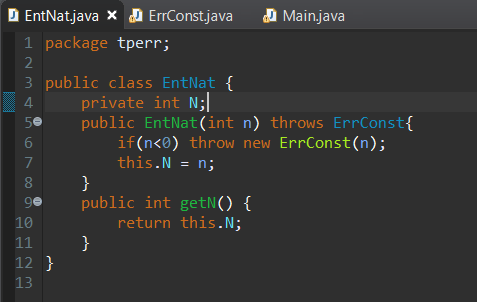


## Exercice 2 : Transmission d’information au gestionnaire

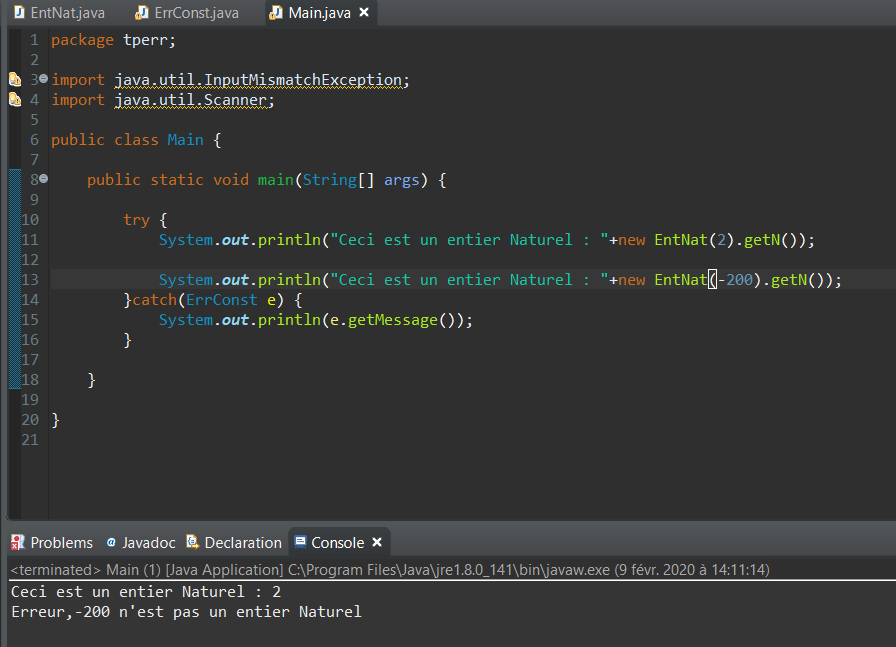
La classe ErrConst avec l’entier comme paramètre.



La classe EntNat en gérant l’exception.

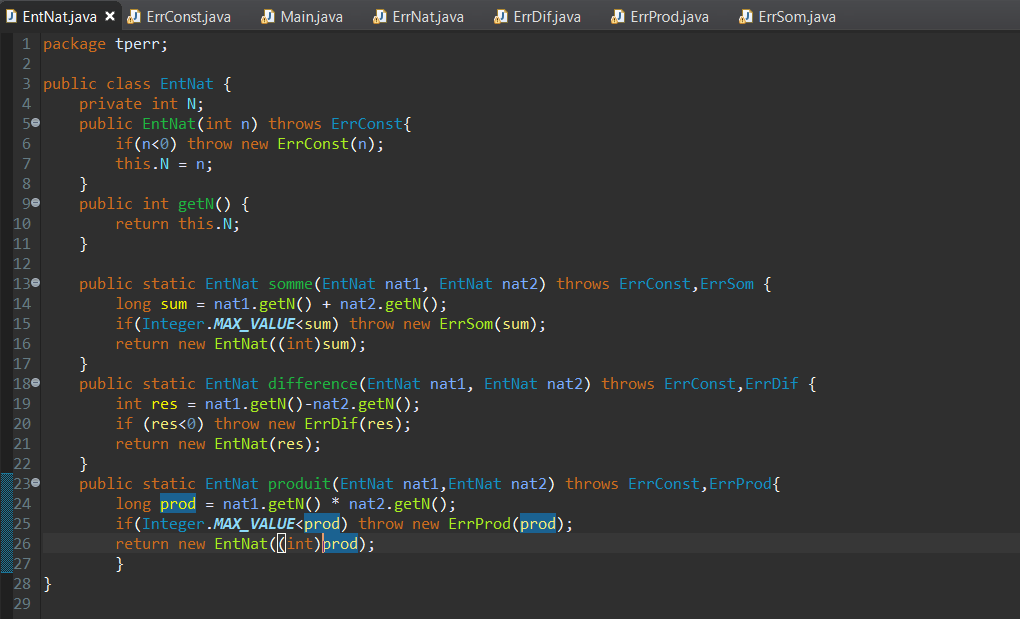


La classe main pour tester :

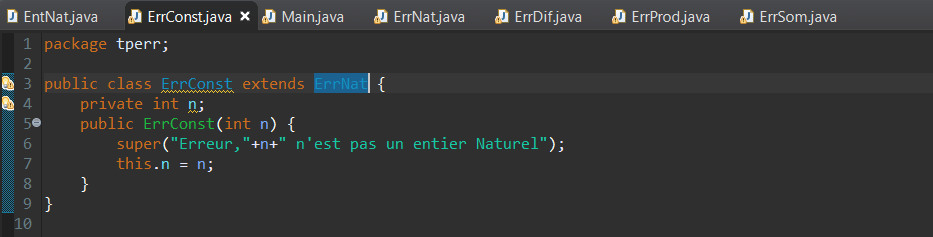


## Problème : Synthèse du chapitre

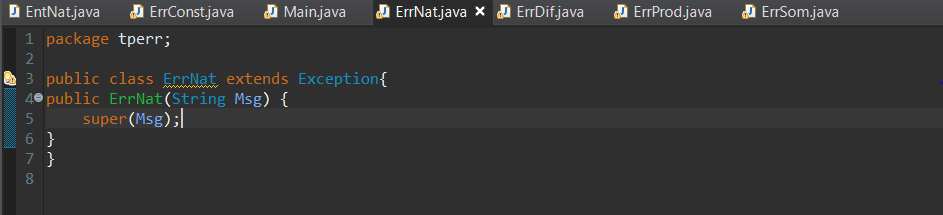
La classe EntNat avec les nouveaux méthodes et nouveaux exceptions :



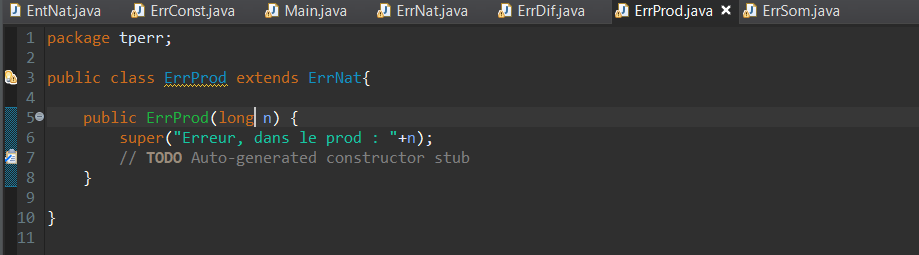
La classe ErrConst qui dérive du classe ErrNat (exception) :



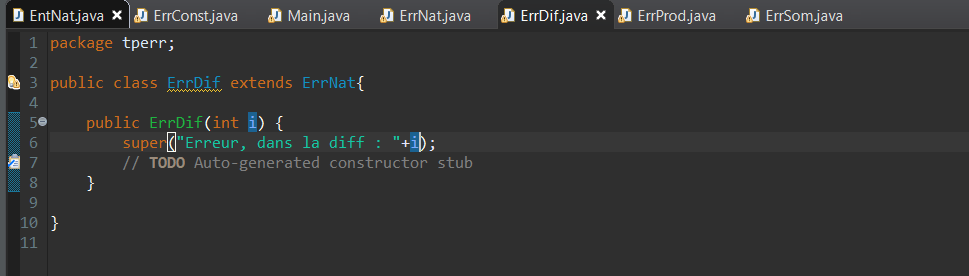
La classe ErrNat avec un message comme paramètre :



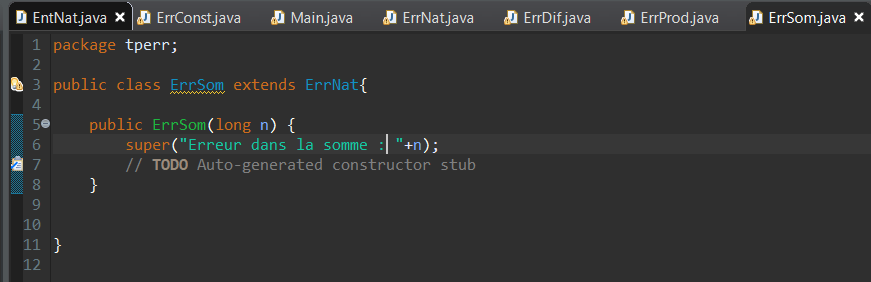
La classe ErrProd qui gère l’exception de la méthode Produit :



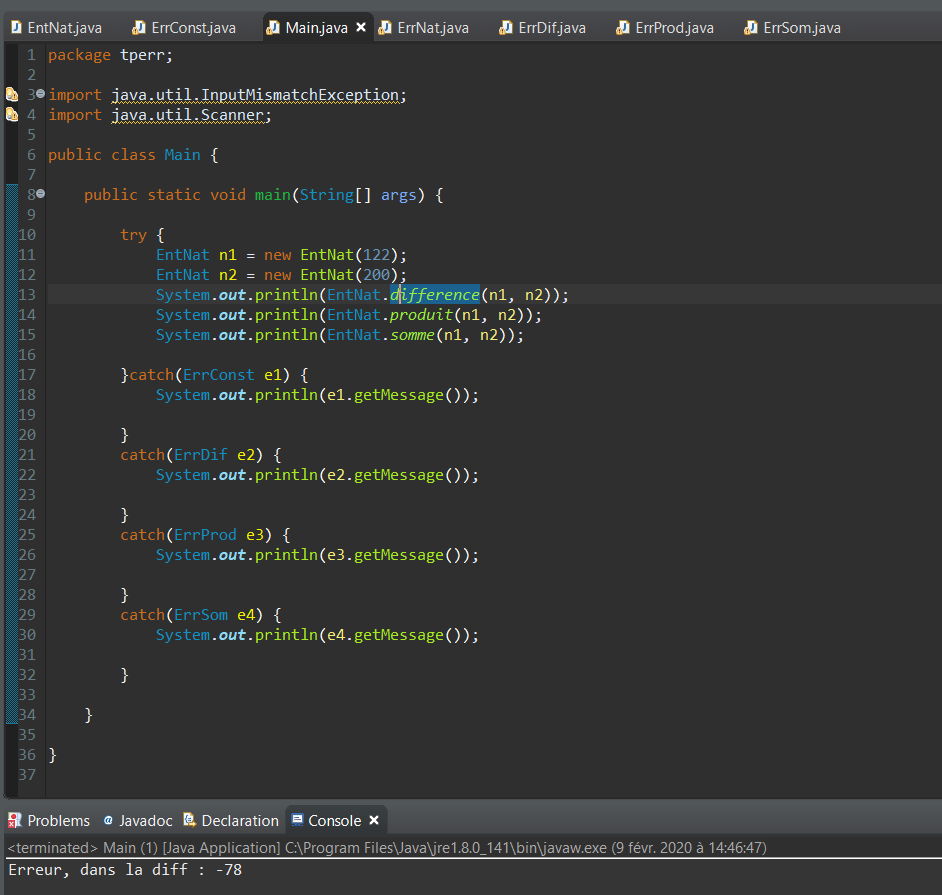
La classe ErrDif qui gère l’exception de la méthode Diff :



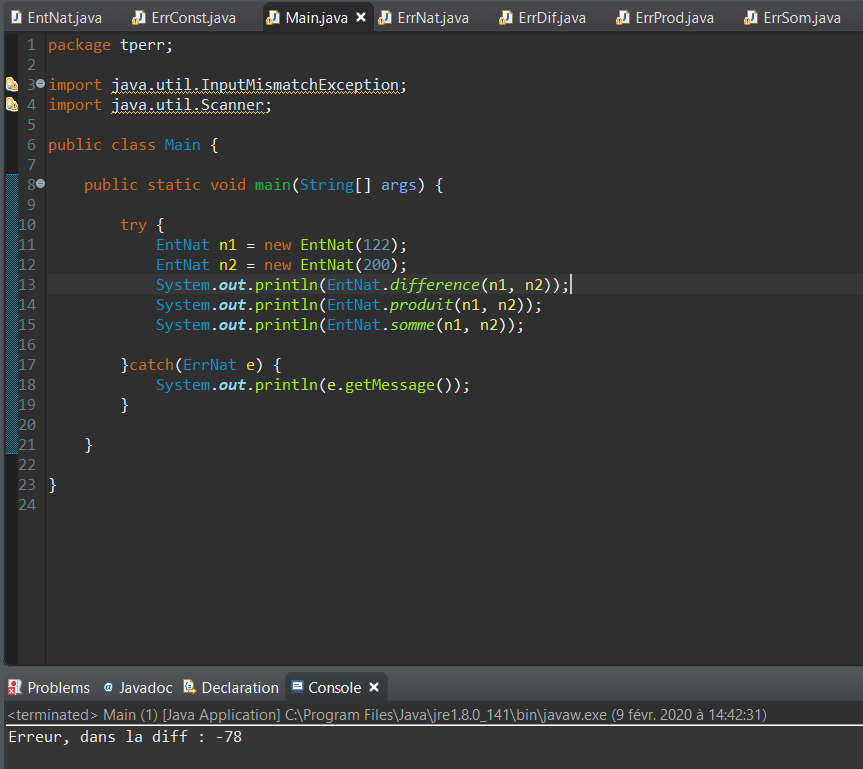
La classe ErrSom qui gère l’exception de la méthode Somme :



Ici la classe Main pour tester les multiples Exceptions :



Ici nous n’avons utilisé que l’exception ErrNat pour attraper les autres erreurs :



CONCLUSION

Finalement c’étais très pratique d’utiliser les plusieurs Exceptions ainsi de tester les différentes cas d’utilisation des exceptions.