جـامعـة الحسن الثـاني بالدار البيـضـاء
UNIVERSITÉ HASSAN II DE CASABLANCA

ECOLE NORMALE SUPERIEURE
DE L'ENSEIGNEMENT TECHNIQUE
DE MOHAMMEDIA

ENSET

DÉPARTEMENT MATHÉMATIQUE INFORMATIQUE

Programmation orientée objet Java

Rapport

Gestion des exceptions

Réaliser par :
ETOULLALI Ayoub

Professeur :
Mr. BOUSSELHAM Abdelmajid

2ème année II-BDCC
Filière d'ingénieur : Ingénieur informatique, Big Data et Cloud Computing

2022/2023

SOMMAIRE

Introduction

Le langage Java a été conçu pour permettre l'exécution du même code sur diverses plate-formes. En particulier, mais pas uniquement, sur le web. Il y a plusieurs types de programmes Java, dont en particulier les applets Java, qui sont intégrées à des pages web et doivent respecter des règles très strictes pour ne pas risquer de causer des dégâts sur les machines d'innocents surfers, et les applications Java, qui fonctionnent comme d'autres programmes, en local sur une machine, et qui ne sont pas limités comme les applets.

Dans les deux cas, le code Java est "compilé", mais les fichiers résultant de la compilation nécessitent encore une interprétation différente suivant chaque plate-forme: cette opération est réalisée par la JVM (Java Virtual Machine).

EXERCICE 1

On souhaite réaliser une application Java contenant une classe **EntierNaturel** permettant de gérer des entiers naturels (positifs ou nuls) et un nouveau type d'exception personnalisé en écrivant une classe **NombreNegatifException** qui spécialise la classe Exception. La classe **EntierNaturel** dispose :

- d'un constructeur avec un argument de type int pour initialiser l'attribut **val**; il générera une exception de type **NombreNegatifException** si la valeur de son argument est négative ;
- un accesseur en lecture getVal() qui fournira sous forme d'un int la valeur encapsulée dans un objet de type **EntierNaturel**;
- -un accesseur en écriture setVal() qui modifiera la valeur de l'entier naturel grâce à un int passé en paramètre; cette méthode générera une exception de type **NombreNegatifException** si la valeur passée en paramètre est négative ;
- une méthode decrementer() qui décrémente de 1 un objet **EntierNaturel**; cette méthode devra pouvoir lever une exception de type **NombreNegatifException**;

Écrire une méthode main qui utilise les méthodes de la classe **EntierNaturel**, en capturant les exceptions susceptibles d'être générées.

On souhaite également mémoriser la valeur erronée qui a entrainé sa génération. Modifiez la classe d'exception **NombreNegatifException** de façon à ce qu'elle permet le stockage de cette valeur, et fournissent une méthode permettant de consulter cette valeur. Testez à nouveau.

Exercice 1

Projet

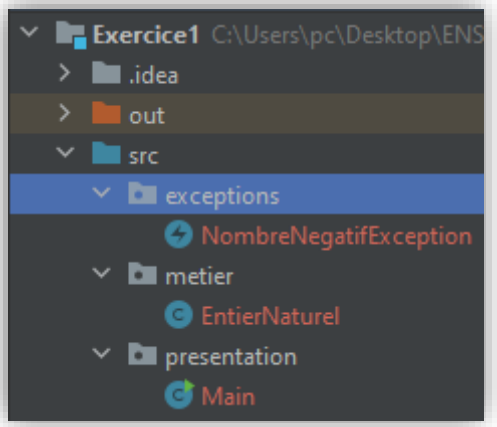
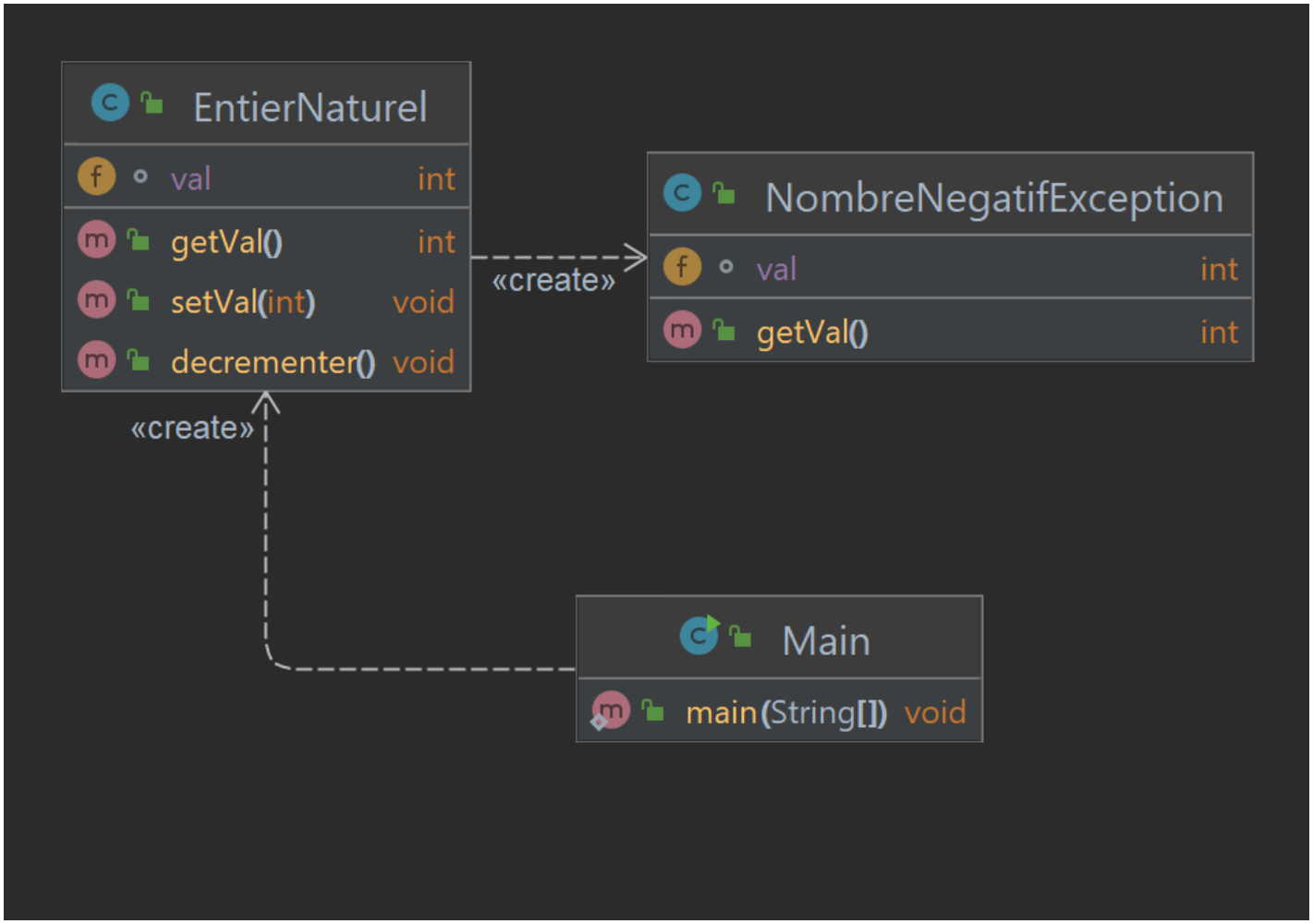


Diagramme de classes



Exécution

```
System.out.println("\n**********cas 1***********");
EntierNaturel en1=new EntierNaturel();
try {
    en1.setVal(2);
    System.out.println(en1.getVal());
    en1.decrementer();
    System.out.println(en1.getVal());
}catch (NombreNegatifException e){
    System.out.println(e.getMessage());
    System.out.println(e.getVal());
}
```

```
**********cas 1***********
2
1
```

```
System.out.println("\n**********cas 2***********");
EntierNaturel en2=new EntierNaturel();
try {
    en2.setVal(-4);
    System.out.println(en2.getVal());
    en2.decrementer();
    System.out.println(en2.getVal());
}catch (NombreNegatifException e){
    System.out.println(e.getMessage());
    System.out.println(e.getVal());
}
```

```
**********cas 2***********
la valeur est négative !!
-4
```

```
System.out.println("\n**********cas 3***********");
EntierNaturel en3=new EntierNaturel();
try {
    en3.setVal(0);
    System.out.println(en3.getVal());
    en3.decrementer();
    System.out.println(en3.getVal());
}catch (NombreNegatifException e){
    System.out.println(e.getMessage());
    System.out.println(e.getVal());
}
```

```
**********cas 3***********
0
On peut pas décrementer la valeur !!
0
```

CONCLUSION :

J'ai appris, au travers de la réalisation des exercices, les fondements du langage Java parce que la maîtrise de ces notions est indispensable pour produire des applications ou des bibliothèques convenables. Néanmoins, pour pleinement profiter des nombreuses autres possibilités offertes par Java, j'ai fait dès maintenant se pencher sur les nombreuses facettes de java.