

Exercices sur les exceptions

Maria-Virginia Aponte, François Barthélémy

NFA031

Exercice 1 exceptions prédéfinies

Il existe en java des exceptions prédéfinies :

- `ArithmeticException` : elle se manifeste lors d'une division par zéro.
- `NullPointerException` : elle se manifeste lorsqu'on cherche à utiliser un objet ou un tableau non initialisé (sur lequel on n'a pas fait de `new`).
- `ArrayIndexOutOfBoundsException` : il s'agit d'une erreur sur l'indice d'un tableau. On cherche à accéder à une case qui n'existe pas.

Il existe une autre exception qui n'est pas une exception prédéfinie du langage java mais que vous pouvez rencontrer, c'est celle qui est utilisée dans la classe `Terminal` : il s'agit de l'exception `TerminalException`. Elle est déclenchée par exemple quand on demande la lecture d'un entier et qu'une autre chose qu'un entier est lu.

Le point commun à ces exceptions est qu'elles sont toutes d'une catégorie d'exception que l'on n'a pas besoin de déclarer dans la clause `throws` des méthodes parce qu'elles sont fréquentes.

Complétez le programme suivant pour que les erreurs susceptibles de se produire soient gérées jusqu'à ce qu'un calcul soit effectivement mené à bout.

```
class Exceptions1{
    static int division(int[] tab, int indice, int diviseur){
        return tab[indice]/diviseur;
    }
    public static void main(String[] args){
        int[] tableau = {17, 12, 15, 38, 29, 157, 89, -22, 0, 5};
        int x, y;
        Terminal.writeString("Entrez l'indice de l'entier à diviser: ");
        x = Terminal.lireInt();
        Terminal.writeString("Entrez le diviseur: ");
        y = Terminal.lireInt();
        Terminal.writeString("Le résultat de la division est: ");
        Terminal.writeIntln(division(tableau,x,y));
    }
}
```

Exercice 2 exercice de lecture

Qu'affiche le programme suivant si l'entier entré est 3? Même question avec 0, 1 et 2.

Cet exercice est à faire sur papier, avec auto-correction en exécutant le programme.

```
class Exceptions2{
    static void methode1(int p){
        Terminal.writeStringln("Debut d'execution de methode1");
        if (p == 0){
```

```

        throw new Exc1 ();
    }
    Terminal.ecrireStringln("Milieu d'execution de methode1");
    if(p ==1){
        throw new Exc2 ();
    }
    Terminal.ecrireStringln("Fin d'execution de methode1");
}
static void methode2(int p){
    Terminal.ecrireStringln("Debut d'execution de methode2");
    if (p == 2){
        throw new Exc3();
    }
    Terminal.ecrireStringln("Milieu d'execution de methode2");
    methode1(p);
    Terminal.ecrireStringln("Fin d'execution de methode2");
}
static void methode3(int p){
    Terminal.ecrireStringln("Debut d'execution de methode3");
    try{
        methode2(p);
    }catch(Exc2 ex){
        Terminal.ecrireStringln("Debut d'execution de methode3");
    }
    Terminal.ecrireStringln("Fin d'execution de methode3");
}
public static void main(String[] args){
    Terminal.ecrireString("Entrez un nombre: ");
    methode3(Terminal.lireInt());
}
}
class Exc1 extends RuntimeException{}
class Exc2 extends RuntimeException{}
class Exc3 extends RuntimeException{}

```

Exercice 3 menu

Il s'agit de faire une saisie de choix pour un menu sécurisé.

Question 1 *levée d'exception*

Nous allons faire une méthode **saisirChoix** qui prendra en paramètre un entier **n** et renverra une valeur comprise entre 1 et **n**, tapée au clavier par l'utilisateur. Les différentes erreurs qui pourront se produire seront :

- **n** est inférieur ou égal à 1
- l'utilisateur a entré un nombre qui n'est pas compris entre 1 et **n**
- l'utilisateur n'a pas entré un nombre

Chaque erreur devra être détectée par le programme et être signalée par une exception spécifique. Il faut donc créer trois classes différentes d'exception.

Question 2 *affichage du menu*

Ecrire une méthode qui prend en paramètre un tableau de chaînes de caractères. Chaque chaîne de caractère décrit un choix du menu. La méthode doit afficher ces différents choix. La taille du tableau peut être différente d'une exécution à l'autre. La même méthode peut afficher différents menus. Néanmoins, pour être réellement un menu, il faut proposer au moins deux choix différents, ce qui nécessite de déclencher une exception si le tableau a moins de deux cases. Cette exception peut être considérée comme la même que le cas n inférieur ou égal à 1 de la question précédente (n étant le nombre de choix proposés).

Question 3 *question et réponse*

Ecrire une méthode qui utilise les deux méthodes déjà écrites pour afficher un menu et saisir le choix de l'utilisateur. Cette méthode ne cherchera pas à résoudre les problèmes signalés par des exceptions. Elle prendra en entrée un tableau d'options (string) et en sortie le choix effectué (sous la forme d'un entier).

Question 4 *main*

Ecrivez une méthode `main` qui propose deux menus différents avec des nombres de choix différents.

Exercice 4 *classe de notes*

On donne la classe suivante pour représenter les notes d'élèves.

```
public class Note {
    private String nomEleve;
    private String matiere;
    private double note;

    public Note(String nomEleve, String matiere, double note) {
        this.nomEleve = nomEleve;
        this.matiere = matiere;
        this.note = note;
    }

    public String getNomEleve() {
        return nomEleve;
    }

    public String getMatiere() {
        return matiere;
    }

    public double getNote() {
        return note;
    }

    public void corrigerNote(double nouvelleNote) {
        this.note = nouvelleNote;
    }

    @Override
```

```

    public String toString() {
        return "Note[eleve=" + nomEleve + ", matiere=" + matiere +
            ", note=" + note + "];"
    }
}

```

Question 1 *Levées d'exceptions*

Modifiez ce programme pour garantir qu'une note est toujours comprise entre 0 et 20. Il faut intervenir au niveau du constructeur et de la méthode `corrigerNote`.

Question 2 *Capture d'exceptions*

```

package nfa031;

import java.util.ArrayList;

public class GestionNotes {

    public static void main(String[] args) {
        double[] notesJeannot = {12, 23, 13};
        ArrayList<Note> listeNotes = new ArrayList<Note>();

        for (int i = 0; i < notesJeannot.length; i++) {
            Note uneNote = new Note("Jeannot Lapin", "Maths", notesJeannot[i]);
            listeNotes.add(uneNote);
        }

        System.out.println("Liste des notes de Jeannot Lapin en Maths :");
        for (int i = 0; i < listeNotes.size(); i++) {
            System.out.println(listeNotes.get(i).toString());
        }
    }
}

```