

Exercices sur les exceptions

Exercice 4.1.1 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 Exo15_1{
    static int[] tableau = {17, 12, 15, 38, 29, 157, 89, -22, 0, 5};
    static int division(int indice, int diviseur){
        return tableau[indice]/diviseur;
    }
    public static void main(String[] args){
        int x, y;
        Terminal.ecrireString("Entrez_l'indice_de_l'entier_à_diviser:_");
        x = Terminal.lireInt();
        Terminal.ecrireString("Entrez_le_diviseur:_");
        y = Terminal.lireInt();
        Terminal.ecrireString("Le_résultat_de_la_division_est:_");
        Terminal.ecrireIntln(division(x,y));
    }
}
```

Exercice 4.1.2 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 Exo15_2{
    static void methode1(int p) throws Exc1, Exc2{
        Terminal.ecrireStringln("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) throws Exc1, Exc2, Exc3{
        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) throws Exc1, Exc3{
        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) throws Exc1, Exc3{
        Terminal.ecrireString("Entrez_un_nombre:_");
        methode3(Terminal.lireInt());
    }
}
class Exc1 extends Exception{ }
class Exc2 extends Exception{ }
class Exc3 extends Exception{ }

```

Exercice 4.1.3 *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.

Pour simplifier, vous pouvez vous limiter à $n < 10$, ce qui permet de ne saisir qu'un caractère au moyen de `Terminal.lireChar`.

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.

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 devra gérer les trois exceptions définies à la question 1. Elle prendra en entrée un tableau d'options (string) et en sortie le choix effectué (sous la forme d'un entier).

Question 4 *classe*

Ecrivez une classe `Menu` qui possède un constructeur pour initialiser un tableau de chaînes représentant différents choix. Cette classe permettra de réaliser une saisie d'un des choix au moyen d'un entier, en utilisant les exception nécessaires.

Cette classe reprendra les fonctionnalités développées aux question précédentes, mais adaptées à la structure de classe.

Ecrivez un programme qui utilise cette classe `Menu` avec trois menus différents qui seront trois objets différents. Ce programme capturera les exceptions susceptibles d'être levées.