

Corrigés des exercices sur les exceptions

Exercice 4.1.1 *exceptions prédéfinies*

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_a_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));
    }
}
```

Corrigé :

```
class Exo15_1_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;
        boolean ok = true;
        do{
            try{
                Terminal.ecrireString("Entrez_l'indice_de_l'entier_a_diviser:_");
                x = Terminal.lireInt();
                Terminal.ecrireString("Entrez_le_diviseur:_");
                y = Terminal.lireInt();
                Terminal.ecrireStringln("Le_resultat_de_la_division_est:_");
                Terminal.ecrireIntln(division(x,y));
            }catch(ArithmeticException e){
                Terminal.ecrireStringln("Le_diviseur_ne_doit_pas_etre_nul.");
                ok = false;
            }
        }while(!ok);
    }
}
```

```

    }catch(ArrayIndexOutOfBoundsException e){
        Terminal.ecrireString("L'indice_doit_etre_compris_entre_0_");
        Terminal.ecrireStringln("et_" + (tableau.length -1) + ".");
        ok = false;
    }catch(TerminalException e){
        Terminal.ecrireStringln("Vous_devez_entrer_un_nombre_entier.");
        ok = false;
    }
}while(!ok);
}
}

```

Exercice 4.1.2 *exercice de lecture*

L'exécution du programme donne l'affichage suivant :

```

> java Exo15_2
Entrez un nombre: 3
Debut d'execution de methode3
Debut d'execution de methode2
Milieu d'execution de methode2
Debut d'execution de methode1
Milieu d'execution de methode1
Fin d'execution de methode1
Fin d'execution de methode2
Fin d'execution de methode3
> java Exo15_2
Entrez un nombre: 0
Debut d'execution de methode3
Debut d'execution de methode2
Milieu d'execution de methode2
Debut d'execution de methode1
Exception in thread "main" Excl
    at Exo15_2.methode1(Exo15_2.java:5)
    at Exo15_2.methode2(Exo15_2.java:19)
    at Exo15_2.methode3(Exo15_2.java:25)
    at Exo15_2.main(Exo15_2.java:33)
> java Exo15_2
Entrez un nombre: 1
Debut d'execution de methode3
Debut d'execution de methode2
Milieu d'execution de methode2
Debut d'execution de methode1
Milieu d'execution de methode1
Debut d'execution de methode3
Fin d'execution de methode3
> java Exo15_2

```

```
Entrez un nombre: 2
Debut d'execution de methode3
Debut d'execution de methode2
Exception in thread "main" Exc3
    at Exo15_2.methode2(Exo15_2.java:16)
    at Exo15_2.methode3(Exo15_2.java:25)
    at Exo15_2.main(Exo15_2.java:33)
```

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.

```

class Exo_15_3{
    public static void main(String[] args){
        String[] st1 = {"Partir_loin", "Rester_au_lit", "Se_lever"};
        String[] st2 = {"Rouge", "Vert", "Bleu", "Jaune", "Violet"};
        String[] st3 = {"Canard", "Poule", "Pintade", "Oie", "Dinde",
                        "Pleiad"};
        Menu[] mt = new Menu[3];
        boolean saisieOk;
        try{
            mt[0] = new Menu("Que_voulez-vous_faire?_", st1);
            mt[1] = new Menu("Quelle_est_votre_couleur_préférée?", st2);
            mt[2] = new Menu("Trouvez_l'intrus.", st3);
        } catch(NPasBon ex){
        }
        for (int i=0; i<5; i++){
            do{
                try{
                    saisieOk = true;
                    mt[i % 3].executerMenu();
                } catch(HorsIntervalle ex){
                    Terminal.ecrireString("Vous_devez_entrer_un");
                    Terminal.ecrireStringln("_des_choix_proposés.");
                    saisieOk = false;
                } catch(PasNombre ex){
                    Terminal.ecrireStringln("Vous_devez_entrer_un_nombre.");
                    saisieOk = false;
                }
            } while(!saisieOk);
        }
    }
}

class NPasBon extends Exception{ }
class PasNombre extends Exception{ }
class HorsIntervalle extends Exception{ }
class Menu{
    String[] tabChoix;
    String question;
    Menu(String q, String[] t) throws NPasBon{
        if (t.length > 9){
            throw new NPasBon();
        }
        tabChoix = t;
        question = q;
    }
    void affiche(){
        Terminal.ecrireStringln(question);
        for(int i=0; i<tabChoix.length; i++){
            Terminal.ecrireStringln(tabChoix[i] + ":_tapez_" + (i+1));
        }
        Terminal.ecrireString("Entrez_votre_choix:_");
    }
    int charToInt(char c) throws PasNombre{

```

```
        if (c == '1'){
            return 1;
        } else if (c == '2'){
            return 2;
        } else if (c == '3'){
            return 3;
        } else if (c == '4'){
            return 4;
        } else if (c == '5'){
            return 5;
        } else if (c == '6'){
            return 6;
        } else if (c == '7'){
            return 7;
        } else if (c == '8'){
            return 8;
        } else if (c == '9'){
            return 9;
        } else if (c == '0'){
            return 0;
        }
        throw new PasNombre();
    }
    int saisir() throws PasNombre, HorsIntervalle{
        char rep = Terminal.lireChar();
        int repInt;
        repInt = charToInt(rep);
        if ((repInt == 0) || (repInt > tabChoix.length)){
            throw new HorsIntervalle();
        }
        return repInt;
    }
    int executerMenu() throws PasNombre, HorsIntervalle{
        affiche();
        return saisir();
    }
}
```
