

Corrigés des exercices sur les exceptions

Exercice 8.1 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.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(tableau,x,y));
    }
}
```

Corrigé :

```
class Exceptions1C{
    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;
        boolean ok;
        do{
            try{
                ok = true;
                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(tableau,x,y));
            }catch(ArithmeticException e){
                Terminal.ecrireStringln("Le_diviseur_ne_doit_pas_etre_nul.");
                ok = false;
            }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);
    }
}

```

À chaque tour de boucle, on commence très optimistes à supposer que tout va bien se passer en positionnant la variable booléenne à `true`. Cette variable, si elle reste à la valeur `true` jusqu'à la fin du tour de boucle, permet d'arrêter celle-ci. Cela n'arrive que si toutes les instructions entre `try` et `catch` sont exécutées sans erreur. Si en revanche, une des trois exceptions est levée à un moment donnée, le bloc `catch` correspondant est exécuté et dans ce bloc, la variable `ok` prend la valeur `false`. Cela force le programme à faire un tour de boucle en plus.

Exercice 8.2 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.ecrireStringln("Debut_d'execution_de_methode1");
        if (p == 0){
            throw new Exc1 ();
        }
    }
}

```

```

        Terminal.afficheStringln("Milieu_d'execution_de_methode1");
        if(p ==1){
            throw new Exc2 ();
        }
        Terminal.afficheStringln("Fin_d'execution_de_methode1");
    }
    static void methode2(int p){
        Terminal.afficheStringln("Debut_d'execution_de_methode2");
        if (p == 2){
            throw new Exc3();
        }
        Terminal.afficheStringln("Milieu_d'execution_de_methode2");
        methode1(p);
        Terminal.afficheStringln("Fin_d'execution_de_methode2");
    }
    static void methode3(int p){
        Terminal.afficheStringln("Debut_d'execution_de_methode3");
        try{
            methode2(p);
        }catch(Exc2 ex){
            Terminal.afficheStringln("Debut_d'execution_de_methode3");
        }
        Terminal.afficheStringln("Fin_d'execution_de_methode3");
    }
    public static void main(String[] args){
        Terminal.afficheString("Entrez_un_nombre:");
        methode3(Terminal.lireInt());
    }
}
class Exc1 extends RuntimeException{}
class Exc2 extends RuntimeException{}
class Exc3 extends RuntimeException{}

```

Les exécutions du programme produisent les affichages suivants :

```

> java Exceptions2
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 Exceptions2
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 Exceptions2.methode1(Exceptions2.java:5)
    at Exceptions2.methode2(Exceptions2.java:19)
    at Exceptions2.methode3(Exceptions2.java:25)
    at Exceptions2.main(Exceptions2.java:33)
> java Exceptions2
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 Exceptions2
Entrez un nombre: 2
Debut d'execution de methode3
Debut d'execution de methode2
Exception in thread "main" Exc3
    at Exceptions2.methode2(Exceptions2.java:16)
    at Exceptions2.methode3(Exceptions2.java:25)
    at Exceptions2.main(Exceptions2.java:33)
```

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.

```
public static int saisir(int n){
    int rep;
    if (n<2){
        throw new PasAssezDeChoix();
    }
    try{
        rep = Terminal.lireInt();
    }catch(TerminalException ex){
        throw new PasNombre();
    }
}
```

```

    }
    if (rep<1 || rep>n){
        throw new HorsIntervalle();
    }
    return rep;
}
...
class PasAssezDeChoix extends RuntimeException{}
class PasNombre extends RuntimeException{}
class HorsIntervalle extends RuntimeException{}

```

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).

```

public static void affiche(String question, String[] tabChoix){
    if (tabChoix.length <2){
        throw new PasAssezDeChoix();
    }
    Terminal.ecrireStringln(question);
    for(int i=0; i<tabChoix.length; i++){
        Terminal.ecrireStringln(tabChoix[i] + " : tapez " + (i+1));
    }
    Terminal.ecrireString("Entrez votre 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).

```

public static int executerMenu(String question, String[] tabChoix){
    affiche(question, tabChoix);
    return saisir(tabChoix.length);
}

```

Question 4 *main*

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

```

class Exceptions3{
    public static void affiche(String question, String[] tabChoix){
        if (tabChoix.length <2){
            throw new PasAssezDeChoix();
        }
        Terminal.ecrireStringln(question);
        for(int i=0; i<tabChoix.length; i++){
            Terminal.ecrireStringln(tabChoix[i] + ":_tapez_" + (i+1));
        }
        Terminal.ecrireString("Entrez_votre_choix:_");
    }
    public static int saisir(int n){
        int rep;
        if (n<2){
            throw new PasAssezDeChoix();
        }
        try{
            rep = Terminal.lireInt();
        }catch(TerminalException ex){
            throw new PasNombre();
        }
        if (rep<1  rep>n){
            throw new HorsIntervalle();
        }
        return rep;
    }
    public static int executerMenu(String question, String[] tabChoix){
        affiche(question,tabChoix);
        return saisir(tabChoix.length);
    }
    public static void main(String[] args){
        String[] st1 = {"Partir_loin", "Rester_au_lit", "Se_lever"};
        String[] st2;
        int n, rep;
        for (int i=0; i<3; i=i+1){
            Terminal.ecrireString("Taille_de_st2?_");
            n = Terminal.lireInt();
            st2=new String[n];
            for (int j=0; j<n; j=j+1){
                st2[j] = ""+(j*100);
            }
            try{
                rep = executerMenu("Que_voulez-vous_faire?",st1);
                Terminal.ecrireStringln("Vous_voulez_" + st1[rep-1]);
                rep = executerMenu("Combien_sont-ils?",st2);
                Terminal.ecrireStringln("Ils_sont_" + st2[rep-1]);
            }
        }
    }
}

```

```
    }catch(HorsIntervalle ex){
        Terminal.ecrireString("Vous_devez_entrer_un");
        Terminal.ecrireStringln("_des_choix_proposés.");
    }catch(PasNombre ex){
        Terminal.ecrireStringln("Vous_devez_entrer_un_nombre.");
    }catch(PasAssezDeChoix ex){
        Terminal.ecrireStringln("Il_faut_au_moins_deux_possibilités.");
    }
}
}
}
class PasAssezDeChoix extends RuntimeException{}
class PasNombre extends RuntimeException{}
class HorsIntervalle extends RuntimeException{}

```
