TP sur les exceptions
Algorithmique – Programmation FIP (ING39)

V. Aponte

Dans ce Tp, une mise en train et 2 petits exercices pour tester vos connaissances sur les exceptions. Les sources se trouvent dans un projet gitlab **déjà existant** ! Ne créez pas de nouveau projet !

Mise en train : Récupération d'erreus de lecture

On vous propose d'étudier différentes versions de la lecture d'un entier au clavier. Toutes utilisent la méthode `Terminal.lireInt()` de la classe `Terminal`. Cette méthode tente de lire un entier au clavier. Si le format de ce qui est lu n'est pas celui d'un entier, `Terminal.lireInt()` échoue en lançant l'exception `TerminalException`. S'il n'y a pas de block try-catch qui entoure les appels à cette méthode, l'exception va se propager vers la méthode appelante. Le but de l'exercice est donc de se tromper de format lors de la saisie, et d'expliquer le comportement de l'exécution de chaque version de lecture.

Comprehénsion du code

Exécutez ce programme et prennez le temps de comprendre le comportément de chaque appel. Les commentaires dans le code vous aideront.

Généraliser la méthode de lecture Robuste

Modifiez la méthode `lectureRobusteValidee()` de sorte qu'elle prenne en paramètre le message affiché pour inviter à taper un entier. Ça pourrait être par exemple : « entrez une température », « entrez une taille de tableau supérieure à zéro et inférieure ou égale à 50 ». Vous devez maintenant appeller cette méthode améloirée pour :

1.
2. Lire une taille de tableau comprise entre 1 et 20 en utilisant votre méthode, puis déclaration d'un tableau de cette taille.
3. lecture et initialisation des composantes de cet tableau. La lecture de chaque composante se fera également par un appel à votre méthode modifiée.
4. Votre programme affichera ensuite la moyenne de tous entiers dans le tableau.

```
public class RecuperationLectureTerminal {

        // Si la lecture échoue on retourne systématiquement -1
        // c'est un peu du bricolage
        public static int lectureRobusteEntierV1() {
                System.out.print("Entrez un entier: ");
                try {
                        // tentative de lecture d'un entier
                        int x = Terminal.lireInt();
                        return x;
                }catch(TerminalException e) {
```

```
                return -1;
        }
}
// Si la lecture échoue on retourne systématiquement -1
// c'est un peu du bricolage
public static int lectureRobusteEntierV2() {
        while(true) {
                System.out.print("Entrez un entier: ");
                try {
                        // tentative de lecture d'un entier
                        int x = Terminal.lireInt();
                        return x;
                }catch(TerminalException e) {
                        System.out.print("Ce n'est pas un entier, recommencez");
                }
        }
}
/**
 * Retourne un entier lu au clavier, compris dans les bornes inf et sup
 * @param inf borne inférieure
 * @param sup borne supérieure
 * @return un entier compris entre inf et sup, autrement redemande la lecture
 */
public static int lectureRobusteValidee(int inf, int sup) {
        while(true) {
                System.out.print("Entrez un entier: ");
                try {
                        // tentative de lecture d'un entier
                        int x = Terminal.lireInt(); // peut échouer
                        if (inf<=x && x <=sup) { // arrivé ici, on a bien entier!
                                return x;  // s'il est dans les bornes on le retourne
                        } else { // sinon message, et reste dans la boucle
                                System.out.println("doit être compris entre "+inf+" et "+
                        }
                }catch(TerminalException e) {
                        // On n'a pas pu lire un entier
                        // Mais on a attrappé l'exception lancé par Terminal
                        // Ici, c'est le code de récupération
                        System.out.print("Ce n'est pas un entier, recommencez");
                }
        }
}

// Le problème: si on se trompe de format d'entier, le programme plante
public static int lecturePasRobuste() {
        System.out.print("Entrez un entier: ");
        return Terminal.lireInt();
}
public static void main(String[] args) {

        int x = lecturePasRobuste();
        System.out.println("Entier lu: "+x);


        // On  tente la solution V1:
        x = lectureRobusteEntierV1();
```

```
                System.out.println("Entier␣lu:␣"+x);

                // On  tente la solution V2:
                x = lectureRobusteEntierV2();
                System.out.println("Entier␣lu:␣"+x);

                x = lectureRobusteValidee(1,10);
                System.out.println("Entier␣lu:␣"+x);

        }
}
```

Exercice 1 : Récupération d'erreurs déclenchées dans une boucle

Expliquez les affichages produits lors d l'exécution de cette classe ;

```
package cours2_exceptions;

public class RecuperationDansBoucle {
        public static void changeCase(int index, int x, int [] t) {
                System.out.print("Tentative␣de␣changement␣case␣"+index+":␣");
                if (t==null | index <0 | index > t.length) {
                        throw new IllegalArgumentException();
                }
                t[index]=x;
                System.out.println("Case␣modifiée!");
        }

        public static void afficheTabInt(int [] t) {
                for (int i=0; i<t.length; i++) {
                        System.out.print(t[i]+"␣");
                }
                System.out.println();
        }

        public static void main(String[] args) {
                int [] tIdx = {0, 6, 1, -1, 2};
                int [] tab = {10,20,30,40};
                // Question 1:
                // En considérant les tableaux tIdx et tab,
                // montrez  tous les affichages de ce main
                try {
                        for (int i=0; i<tIdx.length; i++) {
                                changeCase(tIdx[i], i*10, tab);
                        }
                } catch(IllegalArgumentException e) {
                        System.out.println("Changement␣impossible!");
                }
                System.out.println("Le␣tableau␣tab␣après␣modifications");
                for (int i=0; i<tab.length; i++) {
                        System.out.print(tab[i]+"␣");
                }
                System.out.println();
                // Question 2:
                // En considérant les tableaux tIdx et tab2
```

```
                // donnez tous les affichages de ce main
                int [] tab2 = {10,20,30,40};
                for (int i=0; i<tIdx.length; i++) {
                        try {
                                changeCase(tIdx[i], i*20, tab2);
                        }catch(IllegalArgumentException e) {
                                System.out.println("␣Changement␣impossible!");
                        }
                }
                System.out.println("Le␣tableau␣tab2␣après␣modifications");
                for (int i=0; i<tab2.length; i++) {
                        System.out.print(tab2[i]+"␣");
                }
        }

}
```

Exercice 2 : Récupération d'erreurs qui traversent des appels imbriqués

Jouez avec ce code en réalisant plusieurs appels et en commentant/dé- commentant les blocs try-catch.

```
package cours2_exceptions;

public class RecuperationDansBoucle {
        public static void changeCase(int index, int x, int [] t) {
                System.out.print("Tentative␣de␣changement␣case␣"+index+":␣");
                if (t==null | index <0 | index > t.length) {
                        throw new IllegalArgumentException();
                }
                t[index]=x;
                System.out.println("Case␣modifiée!");
        }

        public static void afficheTabInt(int [] t) {
                for (int i=0; i<t.length; i++) {
                        System.out.print(t[i]+"␣");
                }
                System.out.println();
        }

        public static void main(String[] args) {
                int [] tIdx = {0, 6, 1, -1, 2};
                int [] tab = {10,20,30,40};
                // Question 1:
                // En considérant les tableaux tIdx et tab donnés ci-dessus, donnez tous les affi
                try {
                        for (int i=0; i<tIdx.length; i++) {
                                changeCase(tIdx[i], i*10, tab);
                        }
                } catch(IllegalArgumentException e) {
                        System.out.println("Changement␣impossible!");
                }
                System.out.println("Le␣tableau␣tab␣après␣modifications");
                for (int i=0; i<tab.length; i++) {
                        System.out.print(tab[i]+"␣");
```

```
                }
                System.out.println();
                // Question 2:
                // En considérant les tableaux tIdx et tab2 donnés ci-dessous, donnez tous les af
                int [] tab2 = {10,20,30,40};
                for (int i=0; i<tIdx.length; i++) {
                        try {
                                changeCase(tIdx[i], i*20, tab2);
                        }catch(IllegalArgumentException e) {
                                System.out.println("␣Changement␣impossible!");
                        }
                }
                System.out.println("Le␣tableau␣tab2␣après␣modifications");
                for (int i=0; i<tab2.length; i++) {
                        System.out.print(tab2[i]+"␣");
                }
        }

}
```