

Travaux dirigés n° 4

Exception - Enumération

Exercice 1 (Capture d'exceptions)

1°) Déroulez la classe de test `TestExcept1` suivante, et expliquez la raison de son comportement.

```
import java.io.*;
public class Except1{
    public Except1(){
        System.out.println("Except1:_construction_de_l'instance");
    }
    public void methodeA(int[] tab){
        System.out.println("methodeA:_debut");
        try {
            System.out.println("methodeA:_appel_de_methodeB");
            this.methodeB(tab);
            System.out.println("methodeA:_retour_de_methodeB");
            if (tab.length > 99)
                throw new IOException();
        } catch (IOException e) {
            System.out.println("methodeA:_capture:_"+ e);
        } finally {
            System.out.println("methodeA:_execute_finally");
        }
        System.out.println("methodeA:_fin");
    }
    public void methodeB(int[] t) {
        System.out.println("methodeB:_debut");
        try {
            System.out.println("methodeB:_tente_d'accéder_a_t[99]");
            int a = t[99];
            System.out.println("methodeB:_a_reussi_a_accéder_a_t[99]");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("methodeB:_capture:_"+ e);
        } finally {
            System.out.println("methodeB:_execute_finally");
        }
        System.out.println("methodeB:_fin");
    }
}
```

```
import java.io.*;
public class TestExcept1{
    public static void main(String args[]) {
        System.out.println("main:_debut");
        Except1 ex = new Except1();
        int[] X = {0,1,2,3,4,5,6,7,8,9};
        try {
            System.out.println("main:_appel_de_methodeA");
            ex.methodeA(X);
            System.out.println("main:_retour_de_methodeA");
        } catch (Exception e) {
            System.out.println("main:_capture:_"+e);
        } finally {
            System.out.println("main:_execute_finally");
        }
        System.out.println("main:_fin");
    }
}
```

Exercice 2 (Les cercles le retour)

Reprenons la classe `Cercle` réalisé à l'exercice 4 du TD1 : pour rappel voici le diagramme de classe qui a été réalisé. Nous souhaitons gérer le problème du rayon négatif avec les exceptions.

Cercle
- abscisse : double - ordonnee : double - rayon : double
+ getAbscisse() : double + getOrdonnee() : double + getRayon() : double + setAbscisse(double) : - + setOrdonnee(double) : - + setRayon(double) : - + toString() : String

1°) A quelles méthodes devons-nous apporter des modifications ? et quels types de modifications ?

2°) Sachant que nous souhaitons utiliser une `ArithmeticException` avec une message personnalisé, réécrivez le code de ces méthodes.

3°) Et qu'est-ce que cela change dans une classe de test ? Réécrivez une classe de test qui permettra la gestion correcte des exceptions possiblement engendrées.

Exercice 3 (Do ré mi fa sol la si do...)

On souhaite créer une énumération nommée `Note` qui contient les valeurs suivantes :

`ut`, `re`, `mi`, `fa`, `sol`, `la`, `si` (notez l'emploi du libellé `ut` car `do` n'est pas utilisable puisqu'il s'agit d'un mot-clé)

1°) Ecrivez le fichier définissant cette énumération.

2°) Ecrivez une classe de test qui réalise les affichages suivants :

- le nombre de valeurs du type `Note`
- les valeurs de rang impair du type `Note`
- la dernière valeur du type `Note`
- lit une chaîne au clavier et indique si cette chaîne correspond ou non à un libellé du type `Note`

Exercice 4 (Les cafetières le retour)

Reprenons la classe `Cafetiere` réalisée à l'exercice 2 du TD3 : pour rappel voici le diagramme de classe correspondant. Nous souhaitons ajouter une couleur à la cafetière et réaliser la gestion des erreurs avec les exceptions.

Cafetiere
- capacite : int - quantite : int - <u>compteur : int</u>
+ getCapacite() : int + getQuantite() : int + <u>getCompteur() : int</u> + faireCouler(int) : - + transferer(Cafetiere) : - + egalA(Cafetiere) : boolean + toString() : String + afficher() : -

1°) Comment peut-on représenter cette couleur sachant que nos cafetières sont noires, blanches ou rouges ? Quelles modifications / ajouts devons-nous apporter aux codes pour gérer la couleur ?

2°) Dans quelles méthodes devons-nous gérer des exceptions ?

Réalisez les modifications/ajouts pour l'énumération et les exceptions dans le code.

3°) Modifiez la classe de test en fonction.