Exercices dirigés
séance n°8 - corrigé

Exercice 1 : saisie correcte

On veut écrire la fonction `saisieCorrecte` qui permet de saisir correctement un entier. Si l'utilisateur saisit une donnée dont le format n'est pas celui d'un entier, le programme lève l'exception `InputMismatchException`.

Question 1

La fonction devra traiter cette erreur en fournissant une solution alternative. Un message d'erreur sera affiché avec la proposition d'effectuer une nouvelle saisie.

Note: la classe `InputMismatchException` appartient au package `java.util`.

Question 2

L'entier saisi doit être impérativement supérieur à 10.

On demande donc de créer une classe d'exception adaptée à cette erreur, puis de modifier le programme afin de traiter ce cas d'erreur.

Note:
On aurait pu utiliser l'exception prédéfinie `IllegalArgumentException` pour vérifier que l'entier saisi est supérieur à10.

--- Exercice 1 ------ Solutions ----------

Question 1

```
import java.util.Scanner;
import static java.lang.System.*;
import java.util.InputMismatchException;
public class SaisieEntier{
      static int saisieCorrecte(){
            Scanner input = new Scanner(System.in);
            int x;
            do{
                  try{
                        out.println("entrez un entier -> ");
                        x = input.nextInt();
                        break;
                        }
                  catch(InputMismatchException e){
                        // il faut consommer la valeur du buffer d'entrée
                        s = input.next();
                        out.println("Erreur Recommencez");
                     }
            }while(true);
            return x;
    }

    public static void main(String[]args){
      out.println("entier saisi : "+saisieCorrecte());
    }
}
```

Question 2

```
import java.util.Scanner;
import static java.lang.System.*;
import java.util.InputMismatchException;
public class SaisieEntier{
      static class Inf10Exception extends Exception{
            public Inf10Exception(){
                  super( "division par zéro" );
            }
            public Inf10Exception(String msg){
                  super( msg );
        }
      }

       static int saisieCorrecte () {
            Scanner input = new Scanner(System.in);
             int x = 0;
             String s = null;
             do {
                  try {
                        out.println("entrez un entier -> ");
                        x = input.nextInt();
                        if(x<=10) throw new Inf10Exception();
```

```
                        break;
                    }
                    catch ( InputMismatchException e) {
                        // il faut consommer la valeur du buffer d'entrée
                        s = input.next();
                        out.println(s+"n'est pas un entier");
                        out.println ("Erreur Recommencez");}
                    catch ( Inf10Exception e) {
                        out.println(x+" est inférieur ou égal à 10");
                        out.println ("Erreur Recommencez");}
            } while (true);
             return x;

    }

  public static void main(String[]args){

  out.println("entier saisi : "+saisieCorrecte());

  }
}
```

Exercice 2

On veut créer une classe définissant des cercles.

Question 1

Les cercles sont caractérisés par leur rayon. Définir une classe `Cercle` comportant les deux accesseurs `getRayon` et `setRayon` ainsi qu'une méthode de calcul de la surface. On définira la valeur de π comme une constante de la classe.

Question 2

Ecrire la classe `TestCercle` qui créera 2 instances de cercles et affichera leur surface.

Question 3

A la création d'une instance de cercle, il faut vérifier que la valeur de son rayon est positive.

Dans le cas contraire, on souhaite que le programme de test retourne un message d'erreur. Comment prendre en compte cette contrainte ?

Question 4

Plutôt que d'envoyer un simple message d'erreur, il serait préférable que le programme ne s'interrompe pas mais demande une nouvelle saisie du rayon. Modifier le programme en conséquence.

Question 5

On souhaite encore modifier le programme pour lui permettre d'afficher le nombre d'instances de la classe `Cercle` créées.

--- Exercice 2 ------ Solutions ----------

Question 1

Programme

```
public class Cercle{
  private double rayon;
  private static final double PI=3.14159;
  public Cercle( double rayon ){
    this.rayon = rayon;
  }
  public double getRayon(){
    return rayon;
  }
 public void setRayon( double rayon ){
    this.rayon = rayon;
  }
 public double surface(){
    return rayon*rayon*PI;
  }
}
```

Question 2

```
import static java.lang.System.*;
import java.text.*;
public class TestCercle{
    public static void main(String[] args){
        Cercle c1 = new Cercle(23.7);
        Cercle c2 = new Cercle(8.7);
        NumberFormat formatter = new DecimalFormat("#.00");
        String s = formatter.format(c1.surface());
        out.println("surface du cercle c1 :"+s);
        s = formatter.format(c2.surface());
        out.println("surface du cercle c2 :"+s);
    }
}
```

Question 3

```
public Cercle( double rayon ) throws IllegalArgumentException{
      if(rayon<=0) throw new IllegalArgumentException
                              ("la valeur du rayon est négative");
      this.rayon = rayon;
}
```

```
import static java.lang.System.*;
import java.text.*;
public class TestCercle{
    public static void main(String[] args){
        Cercle c1,c2;
        try{
            c1 = new Cercle(23.7);
            c2 = new Cercle(-8.7);
        }catch( IllegalArgumentException e){
            out.println(e);
        }
        NumberFormat formatter = new DecimalFormat("#.00");
```

```
        String s = formatter.format(c1.surface());
        out.println("surface du cercle c1 :"+s);
        s = formatter.format(c2.surface());
        out.println("surface du cercle c2 :"+s);
    }
}
```

Question4

```
import static java.lang.System.*;
import java.text.*;
import javax.swing.JOptionPane;
public class TestCercle{
    public static void main(String[] args){
        double rayon;
        String data = null;
        String s = null;
        int rep=0;
        NumberFormat formatter = null;
        do{
            try{
                data = JOptionPane.showInputDialog(null,
                            "entrer une valeur derayon",
                            "Boîte de saisie",
                            JOptionPane.INFORMATION_MESSAGE);
                rayon = Double.parseDouble(data);
                formatter = new DecimalFormat("#.00");
                s = formatter.format(new Cercle(rayon).surface());
                out.println("surface du cercle c :"+s);
                rep = JOptionPane.showConfirmDialog(null,
                            "Voulezvouscontinuer ?",
                            "Votre réponse? ",
                            JOptionPane.YES_NO_OPTION);
            }catch( IllegalArgumentException e){
                out.println(e);
            }
        }while(rep==0);
    }
}
```

Question 5

```
public class Cercle{
    private double rayon;
    private static final double PI=3.14159;// ou Math.PI
    private static int nbInstances=0;

    public Cercle( double rayon ){
       if(rayon<=0)
          throw new IllegalArgumentException
                      ("la valeur du rayon est négative");
          this.rayon = rayon;
          nbInstances++;
    }

    public double getRayon(){
        return rayon;
    }

    public void setRayon( double rayon ){
        this.rayon = rayon;
    }

    public double surface(){
        return rayon*rayon*PI;
    }

    public static int getNbInstances(){
        return nbInstances;
    }
}
```

Question 6

On ajoute la ligne suivante à la fin du programme de test :

```
out.println("nombre d'instances : "+Cercle.getNbInstances());
```