

TP CSV et SVG

NFA032

1 Lecture dans un fichier

Pour lire le fichier CSV et pour lire dans un fichier contenant du texte en général, nous vous proposons d'utiliser un scanner exactement comme pour les lectures au clavier. On utilisera donc les méthodes `nextLine`, `nextInt`, etc.

Il y a quand même deux différences avec les lectures au clavier : à la création du scanner, au lieu de donner l'objet `System.in` en paramètre au constructeur, on utilise un autre constructeur qui prend en paramètre un objet représentant en Java le fichier à ouvrir. Il faut commencer par créer cet objet.

L'autre différence est qu'on peut toujours lire du nouveau au clavier alors qu'un fichier contient un texte limité : une fois qu'on a lu tout le fichier, on ne peut plus rien lire dedans.

L'exemple suivant vous montre comment lire dans un fichier ligne par ligne.

```
import java.io.*;
import java.util.Scanner;
import java.util.NoSuchElementException;
public class LectureFichier{
    public static void main(String[] args){
        String line;
        try{
            File file = new File("test.txt");
            Scanner scan = new java.util.Scanner(file);
            int i=1;
            while(true){
                line = scan.nextLine();
                System.out.println("ligne_" + i + " :_" + line);
                i++;
            }
        } catch(FileNotFoundException ioex){
            System.out.println("le_fichier_n'existe_pas");
        } catch(NoSuchElementException nsex){
            System.out.println("fini");
        }
    }
}
```

```
}
```

Dans la ligne : `java.io.File file = new java.io.File("test.txt");` il y a création d'un objet de type `File`. Le paramètre du constructeur est une chaîne de caractère contenant le nom du fichier à ouvrir. Si le fichier n'existe pas, l'exception `FileNotFoundException` est levée.

S'il n'y a plus rien à lire dans le fichier, la méthode `nextLine` lève l'exception `NoSuchElementException`. C'est le cas également de toutes les autres méthodes de lecture (`nextInt`, `nextDouble`, etc).

Question 1

Écrivez un programme qui lit un fichier au format CSV et qui vérifie que toutes les lignes ont bien le même nombre de colonnes. Pour cela, vous pourrez lire le fichier ligne par ligne et utiliser la méthode `split` pour séparer les différents cellules d'une ligne qui sont séparées par une virgule.

Question 2

Modifiez le programme pour que le nom du fichier CSV soit lu au clavier. Pour ce faire, il faut utiliser deux scanner : un qui lit le clavier pour lire le nom du fichier, l'autre qui lit le fichier pour vérifier qu'il y a bien le même nombre de colonnes sur chaque ligne.

2 Fichier SVG

Pour ce qui est d'écrire dans un fichier, nous allons procéder comme pour les affichages à l'écran avec les méthodes `print` et `println`, mais au lieu de les appeler sur `System.out`, on va les appeler sur un objet représentant un fichier en écriture. Pour ce faire, il faut commencer par créer l'objet en question. Puis on écrit dans le fichier avec des `print` et `println`. Enfin, on ferme le fichier. Cette opération de fermeture est celle qui finalise le fichier. Cette opération finale est nécessaire pour assurer que le fichier est correct.

Voici un programme qui écrit un fichier au format SVG.

```
import java.io.*;
import java.util.ArrayList;
public class EcritureSVG{
    public static void main(String[] args){
        try{
            PrintWriter pw = new PrintWriter("exemple.svg");
            pw.println("<?xml_version=\"1.0\"_encoding=\"utf-8\"?>");
            pw.println("<svg_xmlns=\"http://www.w3.org/2000/svg\"");
            pw.println("_version=\"1.1\"_width=\"300\"_height=\"200\">");
            pw.println("<circle_cx=\"90\"_cy=\"80\"_r=\"50\"_fill=\"blue\"_/>");
            pw.println("</svg>");
            pw.close();
        }
    }
}
```

```

        }catch(FileNotFoundException ex){
            System.out.println("Nom_de_fichier_incorrect_" + ex.getMessage());
        }
    }
}

```

À noter dans ce programme : si l'on veut mettre un guillemet à l'intérieur d'une chaîne de caractère il faut la faire précéder d'une barre oblique. En ce qui concerne le nom de fichier, il peut être incorrect s'il spécifie un chemin d'accès qui n'existe pas.

Attention, méfiance : Java ne va pas vérifier si le fichier existe déjà et vous demander de confirmer que vous voulez le réécrire : il va écraser le fichier existant avec les informations écrites par la programme.

Question 1

Modifiez le programme pour que le nom de fichier soit entré au clavier.

Question 2

Modifiez le programme pour que la couleur du cercle soit entrée au clavier.

2.1 Création interactive d'une image

On vous donne le code d'une méthode qui permet de choisir deux opérations : ajouter un cercle ou ajouter un rectangle dans l'image et le squelette d'une méthode main qui permet d'ajouter des cercles et des rectangles à différents endroits et de différentes couleurs. Complétez le code pour que le programme crée une image comportant les formes demandées dans un fichier SVG.

```

import java.io.*;
import java.util.*;
public class InteractiveSVG{
    private Scanner scan;
    public int[] saisirCoordonnees(){
        int[] res = new int[2];
        System.out.print("Entrez_la_coordonnée_X:_");
        res[0] = scan.nextInt();
        System.out.print("Entrez_la_coordonnée_Y:_");
        res[1] = scan.nextInt();
        scan.nextLine();
        return res;
    }
    public String saisirCouleur(){
        String res;
        System.out.print("Entrez_la_couleur_que_vous_voulez:_");

```

```

        res = scan.nextLine();
        return res;
    }
    public int menu(){
        int res;
        System.out.println("1- Ajouter un rectangle");
        System.out.println("2- Ajouter un cercle");
        System.out.println("3- Créer l'image");
        do{
            System.out.print("Votre choix: ");
            res = scan.nextInt();
            scan.nextLine();
            if (res<1 || res>3)
                System.out.println("Choix incorrect: " + res);
        }while(res<1 || res>3);
        return res;
    }
    public void creeImage(){
        scan = new Scanner(System.in);
        ... A compléter
        do{
            choix = menu();
            if (choix == 1){
                ... A compléter

            }else if (choix == 2){
                ... A compléter
            }else if (choix == 3){
                ... A compléter
            }
        }while(choix !=3);
    }
    public static void main(String[] args){
        InteractiveSVG isvg = new InteractiveSVG();
        isvg.creeImage();
    }
}

```

3 Utilisation d'objets

Écrivez maintenant une classe permettant de représenter une forme (cercle ou rectangle). Cette classe comprendra l'information sur la nature de la forme et sur les paramètres de la forme. Elle aura un constructeur et une méthode toString qui renverra le code SVG correspondant à la forme.

Modifiez la classe InteractiveSVG pour qu'elle construise d'abord un ArrayList de formes (ou un tableau de formes), puis utilise cet ArrayList ou ce tableau pour créer le fichier SVG.

4 Héritage

Créez deux ou trois classes utilisant l'héritage de telle sorte qu'il y ait une classe pour les cercles et une autre pour les rectangles.