



M2
Année 2021-2022
Java Multimedia

Travaux Pratiques

Jalel Ben-Othman

Sommaire

TD 1 - Rappels Java
TD 2 - Rappels Java suite
TD 3 - Applets Java
TD 4 - Animations
TD 5 - Multimedia

La rédaction de ces sujets est l'œuvre collective d'un ensemble d'enseignants qui se sont succédés dans ce module. Les auteurs indiqués ici ne sont que les auteurs de la dernière version. Nous souhaitons cependant remercier tous ceux qui ont participé à ce travail.

Equipe enseignante (responsable de cours : J. BEN OTHMAN):

J. Ben-Othman (jbo@univ-paris13.fr)

TP 1**RAPPELS JAVA****1. START UP****Exercice 1**

Compiler et exécuter une application

Voici le source de l'**application HelloWorld** :

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Compilez cette classe à l'aide de la commande : **javac HelloWorld.java**

Exécuter la classe HelloWorld à l'aide de la commande : **java HelloWorld**

Exécuter la classe HelloWorld à l'aide d'eclipse

Exercice 2**Compiler et exécuter une applet**

Voici le source de l'**applet** HelloWorldApplet :

```
import java.awt.Graphics;  
import java.applet.Applet;  
  
public class HelloWorldApplet extends Applet {  
    public void paint(Graphics g) {  
        g.drawString("Hello world!", 50, 25);  
    }  
}
```

Compiliez cette classe à l'aide de la commande : **javac HelloWorldApplet.java**

L'applet doit s'exécuter à l'intérieur d'une page HTML contenant un tag <APPLET CODE=nom de la classe HEIGHT=... WIDTH=...> </APPLET>

Source de la page HelloWorldApplet.html :

```
<HTML>  
<BODY>  
    Ici se trouve notre applet :  
    <P>  
    <APPLET CODE="HelloWorldApplet.class" WIDTH=150 HEIGHT=25>  
    Votre Browser ne permet pas de visualiser les applets Java  
    </APPLET>  
    <P>  
    Et voici du texte placé après la zone réservée à l'applet.  
</BODY>  
</HTML>
```

Testez avec l'appletviewer : **appletviewer HelloWorldApplet.html**

Afficher l'applet en utilisant Eclipse

2. PREMIER PROGRAMME

Exercice 3

Une première "vraie" classe

Voici le source de la classe Livre :

```
public class Livre {  
    // Variables  
    private String titre, auteur;  
    private int nbPages  
  
    // Constructeur  
    public Livre(String unAuteur, String unTitre) {  
        auteur = unAuteur;  
        titre = unTitre;  
    }  
  
    // Accesseur  
    public String getAuteur() {  
        return auteur;  
    }  
  
    // Modificateur  
    public void setNbPages(int n) {  
        nbPages = nb;  
    }  
}
```

Corrigez quelques petites erreurs et ajoutez une méthode main pour
Créer 2 livres,
Faire afficher les auteurs de ces 2 livres.

Lancez l'exécution de la classe Livre**Exercice 4**

Modifiez la classe Livre :

- Ajoutez un accesseur pour la variable titre et la variable nbPages.
- Ajouter un modificateur pour les variables auteur et titre.
- Changez le modificateur de nbPages : il ne devra changer le nombre de pages que si on lui passe en paramètre un nombre positif, et ne rien faire sinon, en affichant un message d'erreur. On remarquera l'intérêt des variables private : on peut contrôler leur modification dans les éventuelles méthodes qui permettent de les modifier.
- Dans la méthode main,
 - donnez le nombre de pages de chacun des 2 livres,
 - faites afficher ces nombres de pages,
 - calculez le nombre de pages total de ces 2 livres et affichez-le.

Lancez l'exécution de la classe Livre.

Exercice 5

Ecrivez une classe TestLivre dans un fichier TestLivre.java. Cette classe a une seule méthode main() qui fait ce que fait la méthode main() de la classe Livre.

Faites exécuter la méthode main de la classe TestLivre.

Exercice 6

Créer 2 classes Livre et TestLivre dans 2 fichiers séparés.

méthode toString()

- Dans la classe Livre, ajoutez une méthode afficheToi() qui affiche une description du livre (auteur, titre et nombre de pages). Utilisez afficheToi() dans la méthode main de TestLivre.
- Ajoutez l'instruction System.out.println(livre) où livre désigne un des livres que vous avez créés. Lancez l'exécution de la classe TestLivre. Vous essaierez de comprendre ce qui est affiché après avoir fait les 2 questions suivantes.
- Ajoutez une méthode toString() qui renvoie une chaîne de caractères qui décrit le livre.

Exercice 7

1. Ajoutez un prix aux livres (nombre qui peut avoir **2 décimales** de type Java double) avec 2 méthodes getPrix et setPrix pour obtenir le prix et le modifier. Ajoutez au moins un constructeur qui prend le prix en paramètre. Testez. Si le prix d'un livre n'a pas été donné, la description du livre (toString()) devra indiquer "Prix pas encore donné". Attention, un livre peut être gratuit (0 est une valeur possible pour un prix) ; la valeur -1 indiquera que le prix n'a pas encore été donné.

2. On bloque complètement les prix : un prix ne peut être donné qu'une seule fois et ne peut être modifié ensuite (une tentative pour changer le prix ne fait qu'afficher un message d'erreur).

Récrivez la méthode setPrix (et autre chose si besoin est). Vous ajouterez une variable booléenne prixFixe (pour "prix fixé") qui indiquera que le prix ne peut plus être modifié. Faut-il écrire une méthode "setPrixFixe" pour modifier la variable booléenne ? Ajoutez une méthode "isPrixFixe" qui renvoie vrai si le prix a déjà été fixé.

TP 2

RAPPELS JAVA (2)

1. ANNUAIRE

Ex 1 :

Les fiches de personnes doivent permettre de :

- Ecrire le nom de la personne : `public void setName(String name);`
- Lire le nom de la personne : `public String getName();`
- Ecrire l'adresse de la personne : `public void setAdd(String add);`
- Lire l'adresse de la personne : `public String getAdd();`
- Ecrire le num de tel : `public void setTel(int tel);`
- Lire le num de tel : `public int getTel();`

Ecrire la classe des fiches de personnes.

Ex 2 :

L'annuaire doit permettre de :

- Ajouter une nouvelle fiche
- Supprimer une fiche
- Recherche d'une fiche en fonction de son nom (complet ou pas)
- Recherche d'une fiche en fonction de son tel
- Modifier une fiche

Ecrire la classe de l'annuaire.

2. FICHER CLIENT

Ex 3 :

Créer une classe client à partir de la classe personne permettant de rajouter des informations commerciales sur le client à savoir :

- Nom produit
- Code produit
- Quantité
- Prix Unitaire

Rajouter les méthodes suivantes :

- Affichage commande qui affiche la commande d'un client
- Ajout d'une commande
- Affichage des clients d'un produit

TP 3

APPLETS JAVA

1. AFFICHAGE DE CARACTERES

Ci dessous un petit programme qui permet d'afficher une chaîne de caractères sur une fenêtre appletviewer

```
// lors de la creation de la class la faire executer comme une
applet
import java.applet.*;
// requis pour appeler le package applet
import java.awt.*;
// la classe HelloWorld est une classe executable
// correspondant a une extension de l'applet
public class HelloWorld extends Applet
{

// la method init () doit être invoqué au démarrage de l'applet
meme comme dans ce cas elle est vide
    public void init()
    {
// elle ne nécessite rien
    }

// This method gets called when the applet is terminated
// That's when the user goes to another page or exits the browser.
    public void stop()
    {
// no actions needed here now.
    }

// il faut utiliser la method paint pour afficher une chaîne de
caractère sur l'appletviewer

    public void paint(Graphics g)
    {
```



```
// la method draw.String prend une chaine de caractère et des
// coordonnées où vont s'afficher les chaines de caractères
g.drawString("Salut c'est ton premier",140,20);
    g.drawString("Hello World",20,140);
    g.drawString("avec l'appletviewer",20,160);

}

}
```

Ex 1 :

Modifier le code afin d'afficher de manière justifier les trois lignes de caractères

2. AFFICHAGE DE POINT LIGNES

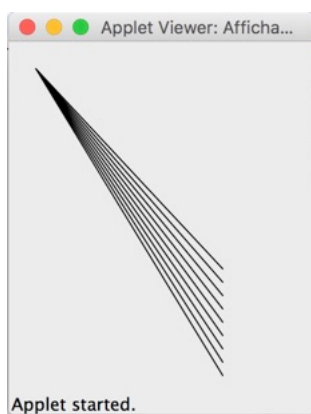
// Draw a line (int x1, int y1, int x2, int y2)

```
g.drawLine(10,10,160,160);
```

Permet d'afficher une ligne allant de x1, y1 à x2, y2

Ex 2 : Faite un programme affichant un X dans l'appletviewer

Ex 3 : Faite un programme affichant :

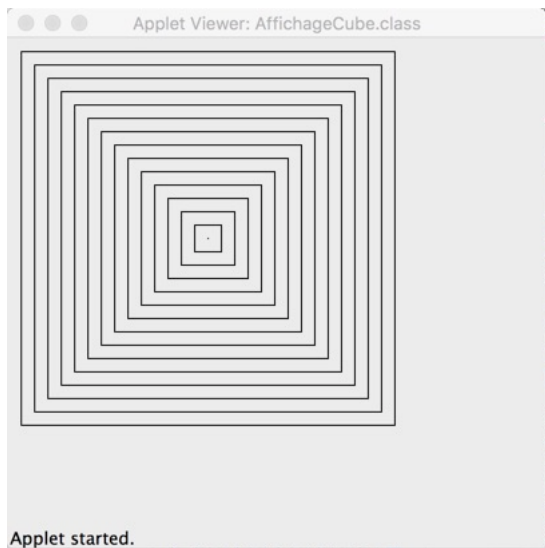


Ex 3 : A l'aide des lignes réaliser

// This will draw a rectangle (xco,yco,xwidth,height); g.drawRect(100,100,100,100);

Permet de représenter un rectangle dont le haut gauche est à x1,y1 et xwidth donne la taille en abscisse et height donne la taille en ordonnée

Ex 4 : Réaliser un programme permettant de réaliser la forme suivante :



Les autres commandes sont :

```
g.setColor(Color.blue);
g.fill3DRect(arg0, arg1, arg2, arg3, arg4)
g.draw3DRect(arg0, arg1, arg2, arg3, arg4)
g.setColor(arg0)
g.drawRoundRect(arg0, arg1, arg2, arg3, arg4, arg5)
g.drawPolygon(arg0)
g.drawOval(arg0, arg1, arg2, arg3)
g.drawArc(arg0, arg1, arg2, arg3, arg4, arg5)
g.drawImage(arg0, arg1, arg2, arg3, arg4, arg5, arg6, arg7, arg8, arg9)
g.fillRect(arg0, arg1, arg2, arg3)
g.fillRoundRect(arg0, arg1, arg2, arg3, arg4, arg5)
g.fill3DRect(arg0, arg1, arg2, arg3, arg4)
g.fillPolygon(arg0, arg1, arg2)
g.fillOval(arg0, arg1, arg2, arg3)
g.fillArc(arg0, arg1, arg2, arg3, arg4, arg5)
g.drawChars(arg0, arg1, arg2, arg3, arg4)
```

Expliquez l'utilité de chaque méthode et testez les.

TP 4

ANIMATIONS

1. SOURIS

Ex 1 : Que fait le programme suivant

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
public class Prog67
    extends Applet
    implements MouseListener {
    Rectangle6 rectRouge, rectBleu;
    boolean affiche;
    public void init() {
        addMouseListener(this);
        affiche = true;
        rectRouge = new Rectangle6(30,30,50,80,Color.blue);
        rectBleu = new Rectangle6(30,30,50,80,Color.blue);
    }
    public void paint(Graphics g) {

        if (affiche)
            rectRouge.dessiner(g);
        else
            rectBleu.dessiner(g);
    }
    public void mouseClicked(MouseEvent e) {
        affiche = ! affiche;
        repaint();
    }
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
}
```

Ex 2 : Etendre le programme pour changer la position du rectangle et la couleur

Faire disparaître le rectangle

2. THREADS

```
import java.applet.*;
import java.awt.*;
public class animation extends Applet implements Runnable
{ Thread runner = null;          //variables de classe
  double t;
  public void init()
  { setBackground (Color.lightGray);}
  public void paint(Graphics g)
  { int X=140+(int)(120*Math.sin(t)); //calcul de la position
    t+=0.05;                          //déplacement lors de l'appel suivant
    g.fillOval(X,20+X,50,100);}
  public void start()              //surcharge de la méthode
  { if (runner == null)           //test d'existence
    { runner = new Thread(this); //création , this désigne l'applet
      runner.start();}           // lancement de la méthode run( )
  public void stop()
  { if (runner != null)
    { runner.stop();
      runner = null;}} //destruction

  public void run()
  { while (true)                  // boucle sans fin
    { try                          //action à réaliser
      { repaint();                // redessiner
        Thread.sleep(20);}        //pause de 20 ms
      catch (InterruptedException e)
      { stop();}}                 //traitement (facultatif) de l'exception
  }
```

Ex 3 : que fait le programme

Comment faire pour diminuer la vitesse d'exécution

Afficher l'éclipse de telle sorte à ce qu'elle se déplace en diagonale

Modifier le programme pour afficher un rectangle

Afficher un rectangle avec une trainée (dans les deux sens puis dans un sens)

Réaliser une spirale rectangulaire.

3. IMAGES ANIMES

En rajoutant les commandes suivantes dans la méthode paint :

Image img ;

```
img = getImage(getCodeBase(),"/Users/jalelben-othman/Documents/workspace/Java/new/clipart.jpg");  
g.drawImage(img,0,0,200,200,this);
```

Vous pouvez afficher une image.

Ex 3 :

Adapté le programme pour faire bouger l'image sur l'axe des abscisse

Ecrire un programme qui affiche une image animée (avec les images image1.jpg, image2.jpg, image3.jpg que je vous remettrai)

Ajouter sur l'image animée la possibilité d'arrêter une image et de relancer l'animation

TP 5

MULTIMEDIA

1. CLAVIER

De même que la souris pour utiliser le clavier sur une applet il faut utiliser `KeyListener`. Avec la commande :

```
public class clavier extends Applet implements KeyListener
```

Dans la méthode `init()` il faut utiliser l'instruction :

```
addKeyListener(this);
```

Les événements sur le clavier sont :

```
public void keyPressed(KeyEvent evt)
```

intercepte une touche au clavier

```
public void keyTyped(KeyEvent evt) { }
```

intercepte une touche au clavier

```
public void keyReleased(KeyEvent evt) { }
```

relâchement du clavier

Ces méthodes ont comme paramètre un objet de type `KeyEvent`; celui-ci possède des méthodes permettant de savoir quelle touche a été utilisée. On trouve en particulier :

- **`getKeyChar()`** qui renvoie le caractère associé à la touche.
- **`getKeyCode()`** qui renvoie un code sous forme d'entier de type `int` pour les touches qui ne sont pas associées à un caractère : flèches, F1, F2, ..., Suppr, etc...

pour le clavier les codes sont les suivants :

- o `evt.VK_LEFT` pour un déplacement à gauche
- o `evt.VK_RIGHT` pour un déplacement à droite
- o `evt.VK_UP` pour un déplacement en haut
- o `evt.VK_DOWN` pour un déplacement à droite

Ex 1 :

En utilisant le modèle du programme sur la souris de la partie 1 du Td précédent, écrire un programme qui prend en argument une chaîne de 5 caractères et l'affiche sur l'appletviewer.

Ex 2 :

Ecrire un programme qui déplace un rectangle sur une fenêtre appletviewer à partir du clavier. Etendre ce programme pour qu'il puisse déplacer ce rectangle à partir du clavier et de la souris.

Ex 3 :

Ecrire un programme qui déplace la chaîne bonjour dans l'appletviewer et à chaque clique la chaîne reste sur l'appletviewer.

2. SON

Soit le programme suivant :

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class son extends Applet
implements ActionListener {

    AudioClip ding;
    AudioClip fond;
    //Button bding=new Button("DING");
    Button bfond=new Button("DEPART");
    boolean bruitFond=false;

    public void init() {
        fond=getAudioClip(getCodeBase(),"/Users/jalelben-othman/portable/son.au");
        add(bfond);
        bfond.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==bfond) {
            if (bruitFond) {
                fond.stop();
            }
        }
    }
}
```



```
        bfond.setLabel("DEPART");
        bruitFond=false;
    }
    else {
        fond.loop();
        bfond.setLabel("STOP");
        bruitFond=true;
    }
}
}

public void stop() {
    if (bruitFond) fond.stop();
}
}
```

Ex 4 :

Que fait ce programme ?

Faire une extension pour rajouter l'option pause.

3. MULTIMEDIA AVEC JMF

Soit le programme :

```
import javax.media.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class JMFTTest extends JFrame
{
    private static final long serialVersionUID = 7342473856313282669L;

    Player _player;

    public JMFTTest()
    {
        addWindowListener( new WindowAdapter()
```

```
        {
            public void windowClosing( WindowEvent e )
            {
                _player.stop();
                _player.deallocate();
                _player.close();
                System.exit( 0 );
            }
        }
    );

    setBounds( 0, 0, 320, 260 );
    JPanel panel = (JPanel) getContentPane();
    panel.setLayout( new BorderLayout() );
    String mediaFile =
        "file://Users/jalelbenothman/portable/sample.mov";
    try {
        MediaLocator mlr = new MediaLocator( mediaFile );
        _player = Manager.createRealizedPlayer( mlr );
        if ( _player.getVisualComponent() != null )
            panel.add("Center", _player.getVisualComponent());
        if ( _player.getControlPanelComponent() != null )
            panel.add("South", _player.getControlPanelComponent());
    }
    catch (Exception e)
    {
        System.err.println( "Got exception " + e );
        System.exit(0) ;
    }
}

public static void main(String[] args)
{
    JMFTest jmfTest = new JMFTest();
    jmfTest.show();
}

}
```

Ex 5 :

Récupérer le fichier sample.mov

Que fait ce programme ?

PROJET

EXAMEN

1. MODALITES

Le projet est à réaliser individuellement. Toute copie (cad code de livre, internet ...) sera sanctionnée par une note à 0/20. Une présentation du projet sera faite le en salle de TP suivant l'ordre alphabétique. Vu le nombre d'étudiants, chaque étudiant aura ¼ d'heure pour exposer ses travaux et répondre aux questions. A l'issue de l'exposé une copie du code ainsi qu'un rapport détaillé sera à remettre.

L'étudiant doit choisir un sujet parmi les trois proposés. La note finale qui comptera comme note d'examen sera composé de la qualité du code, de la présentation de la réponse aux questions ainsi que le rapport (1/4 chacun).

2. SUJET 1 : HOROLGE ANALOIGIQUE ET NUMERIQUE

Il est demandé d'implémenter une horloge qui suivant le clic sur la souris permet un affichage analogique ou numérique (l'affichage doit inclure l'heure les minutes les secondes). Il faut de plus au lancement de l'application demander à l'utilisateur de saisir l'heure (cad heure :minutes :secondes)

3. SUJET 2 : TENNIS

Il est demandé d'implémenter le jeux du tennis des premiers systèmes informatiques. Le programme doit offrir la possibilité de jouer contre un adversaire ou bien contre la machine. Pour le déplacement du joueur (représenté par une barre) l'utilisateur peut utiliser le clavier ou la souris (ou les deux si on est en mode adversaire). Sur la fenêtre doit s'afficher le score de chaque joueur à savoir le score du set et le jeu en cour (cf wikipedia pour connaître les règles).

Une partie se termine lorsque l'un des deux joueur a gagné le nombre de set requis.

4. SUJET 3 : VIDEO ANIMEE

Il est demander de réaliser une animation où l'on voit une image se déroulée comme un dessin animé, sur cette image il est possible de mettre différentes bandes sonores choisies par l'utilisateur ainsi qu'un texte. Ainsi au démarrage il est demandé à l'utilisateur de choisir l'image désirée, le son et le texte. Les images, sons et textes doivent être fournies.