I-Thread

1-java.lang.Thread EX BANIO

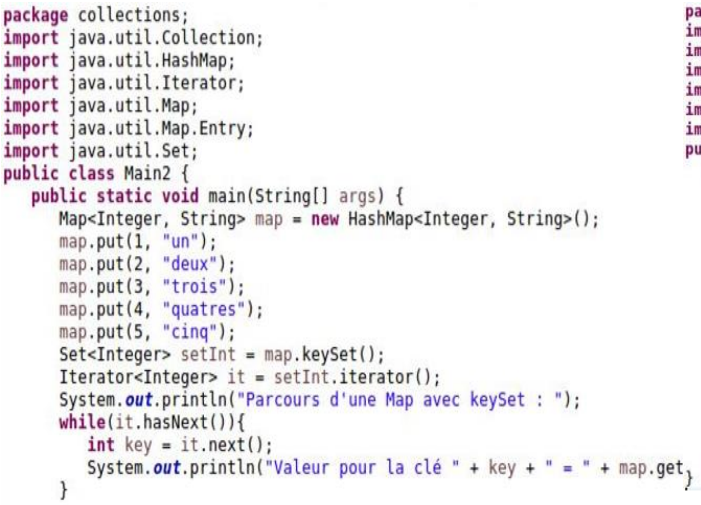
```
public class Baignoire  implements Runnable{
public static int QteMax=5000;
int fuite;
public static Eau Contune;
public Baignoire(int e,int f) {
            Contune=new Eau(e);
            fuite=f;
}
public  void run(){
for(;;) {
if(Baignoire.Contune.volumeEauA<=0) {
System.out.println("(Baignoire)Baignoire Vide");
Baignoire.Contune.volumeEauA=0;
break;
}else if(Baignoire.Contune.volumeEauA>=this.QteMax) {
System.out.println("(Baignoire)Baignoire Pleine");
Baignoire.Contune.volumeEauA=this.QteMax;
break;
}else {
Baignoire.Contune.volumeEauA-=fuite;
if(Baignoire.Contune.volumeEauA<0)
{Baignoire.Contune.volumeEauA=0;}
System.out.println("(Baignoire)Baignoire
"+Baignoire.Contune.volumeEauA);
}
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
e.printStackTrace();
```

```
public class Robinet  extends Thread {
            int Qte;
            public Robinet(int a) {
                        Qte=a;
            }
            public  void run() {
                        for(;;) {
                                    if(Baignoire.Contune.volumeEauA<=0) {
                        System.out.println("(Robinet)Baignoire Vide");

            Baignoire.Contune.volumeEauA=0;
            break;}
                        else
if(Baignoire.Contune.volumeEauA>=Baignoire.QteMax) {
                                    System.out.println("(Robinet)Baignoire
Pleine");
            Baignoire.Contune.volumeEauA=Baignoire.QteMax;
                        break;
            }else {
            Baignoire.Contune.volumeEauA+=Qte;

            System.out.println("(Robinet)Baignoire
"+Baignoire.Contune.volumeEauA);
            }
            try {
            Thread.sleep(1000);
            } catch (InterruptedException e) {
            e.printStackTrace();
}}}
```

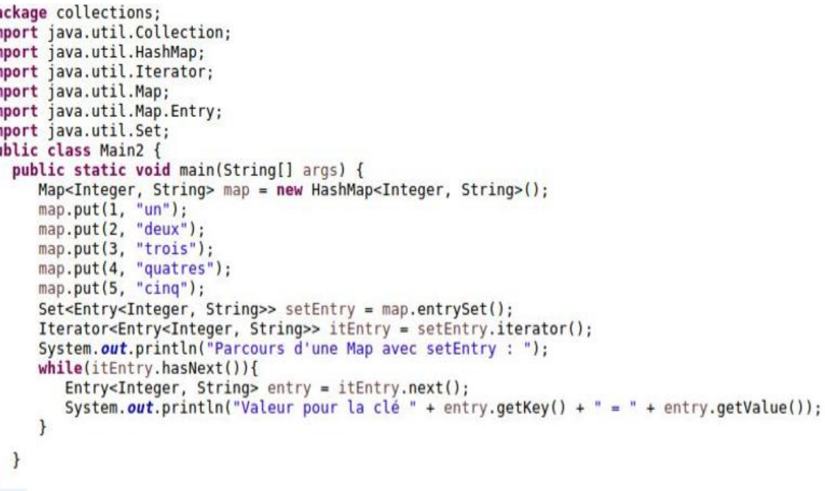
```
public class Eau {
public static int volumeEauA;
public Eau(int e) {this.volumeEauA=e;
}
}
```

```
public class Remplir {
      public static void main(String[] args) {
            Baignoire b=new Baignoire(10,1);
            Robinet r=new Robinet(1);
            new Thread(r).start();
            new Thread(b).start() }}
```

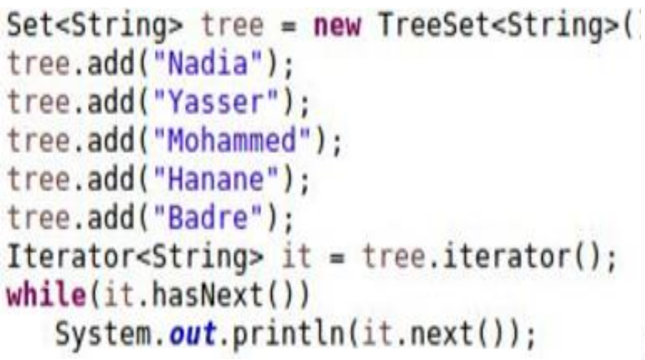
II-collection



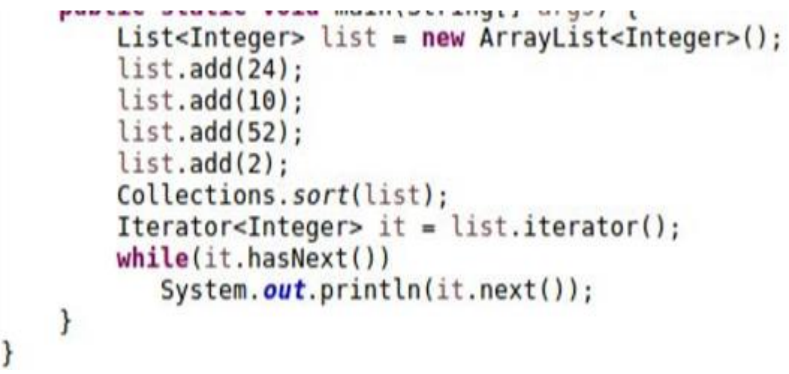
```
package collections;
import java.util.Collection;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
public class Main2 {
   public static void main(String[] args) {
      Map<Integer, String> map = new HashMap<Integer, String>();
      map.put(1, "un");
      map.put(2, "deux");
      map.put(3, "trois");
      map.put(4, "quatres");
      map.put(5, "cinq");
      Set<Integer> setInt = map.keySet();
      Iterator<Integer> it = setInt.iterator();
      System.out.println("Parcours d'une Map avec keySet : ");
      while(it.hasNext()){
         int key = it.next();
         System.out.println("Valeur pour la clé " + key + " = " + map.get
      }
```



```
package collections;
import java.util.Collection;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
public class Main2 {
   public static void main(String[] args) {
      Map<Integer, String> map = new HashMap<Integer, String>();
      map.put(1, "un");
      map.put(2, "deux");
      map.put(3, "trois");
      map.put(4, "quatres");
      map.put(5, "cinq");
      Set<Entry<Integer, String>> setEntry = map.entrySet();
      Iterator<Entry<Integer, String>> itEntry = setEntry.iterator();
      System.out.println("Parcours d'une Map avec setEntry : ");
      while(itEntry.hasNext()){
         Entry<Integer, String> entry = itEntry.next();
         System.out.println("Valeur pour la clé " + entry.getKey() + " = " + entry.getValue());
      }

   }
}
```



```
Set<String> tree = new TreeSet<String>(
tree.add("Nadia");
tree.add("Yasser");
tree.add("Mohammed");
tree.add("Hanane");
tree.add("Badre");
Iterator<String> it = tree.iterator();
while(it.hasNext())
   System.out.println(it.next());
```



```
public static void main(String[] args) {
   List<Integer> list = new ArrayList<Integer>();
   list.add(24);
   list.add(10);
   list.add(52);
   list.add(2);
   Collections.sort(list);
   Iterator<Integer> it = list.iterator();
   while(it.hasNext())
      System.out.println(it.next());
   }
}
```

III-IO

DataInputStream dis;DataOutputStream dos;   dos.writeByte(100); dos.writeChar('C'); dis.readByte()

```
import java.io.File;
public class Fichier {
  public static void main(String[] args) {
    File f = new File("fichier.txt");
    System.out.println("Chemin absolu du fichier : " +
f.getAbsolutePath());
    System.out.println("Est-ce qu'il existe ? " + f.exists());
    System.out.println("Est-ce un répertoire ? " + f.isDirectory());
    System.out.println("Affichage des lecteurs à la racine du PC : ");
    for(File file : File.listRoots()){
      System.out.println(file.getAbsolutePath());
      try {
        int i = 1;
        for(File nom : file.listFiles()){
                 System.out.print("\t\t" + ((nom.isDirectory()) ?
nom.getName()+"/" : nom.getName()));
                 if((i%5) == 0){
                          System.out.print("\n");
                 }
                 i++;
        }
        System.out.println("\n");
      } catch (NullPointerException e) {}
    }
  }
}
```

Transient- Serializable

```
ObjectInputStream ois;
ObjectOutputStream oos;
                  try {
oos = new ObjectOutputStream(new
BufferedOutputStream(newFileOutputStream(new File("game.txt"))));
oos.writeObject(new Game("Assassin Creed", "Aventure", 45.69));
oos.writeObject(new Game("Tomb Raider", "Plateforme", 23.45));
oos.writeObject(new Game("Tetris", "Stratégie", 2.50));
oos.close();
FileInputStream fis =new FileInputStream(new File("game.txt"));
ois = new ObjectInputStream(fis);
try {
                                   Game g;
                                   while(fis.available()>0) {
                                            g=(Game)ois.readObject();
                                            System.out.println(g);
                                   }
                          } catch (ClassNotFoundException e)
{e.printStackTrace();}
                          ois.close();
                  } catch (FileNotFoundException e) {e.printStackTrace();}
                          catch (IOException e) {e.printStackTrace();}
```

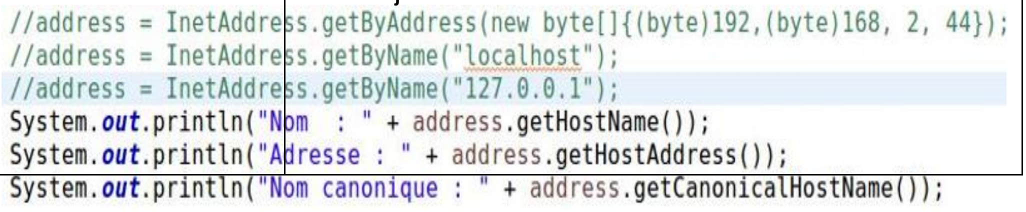
```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException
public class FileCPbis {
          public static void main(String[] args) {
            FileInputStream fis = null;
            FileOutputStream fos = null;
            BufferedInputStream bis=null;
            BufferedOutputStream bos=null;
            File f = new File("fichier.txt");
            try {
              fis = new FileInputStream(f);
              fos = new FileOutputStream(new File("fichierCp.txt"));
              bis = new BufferedInputStream(new
FileInputStream(new File("fichier.txt")));
              bos = new BufferedOutputStream(new
FileOutputStream(new File("fichierCp1.txt")));
              byte[] buf = new byte[8];
              int n = 0;
              long start = System.currentTimeMillis();
              while (fis.read(buf) >= 0) {
                fos.write(buf);
              }
              System.out.println("Copie terminée dans:
"+(System.currentTimeMillis()-start)+"ms");

              start = System.currentTimeMillis();
              while (bis.read(buf) >= 0) {
                bos.write(buf);
              }
              System.out.println("Copie terminée dans:
"+(System.currentTimeMillis()-start)+"ms");
            }
            catch (FileNotFoundException e) {e.printStackTrace();}
            catch (IOException e) {e.printStackTrace();}
            finally {
                   try {
                            if (fis != null) fis.close();
                   } catch (IOException e) {e.printStackTrace();}
                   try {
                            if (fos != null)fos.close();
                   } catch (IOException e) {e.printStackTrace();}
            }
          }
}
```

```
------------------------------file----------------------------------------------
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class FichierTxt {
         public static void main(String[] args) {
                  File file = new File("fichie.txt");
           FileWriter fw;
           FileReader fr;
           try {
                  fw = new FileWriter(file);
```

```
         String str = "Bonjour à tous\n";
                  str += "\tComment allez-vous ? \n";
                  fw.write(str);
                  fw.close();
                  fr = new FileReader(file);
                  str = "";
                  int i = 0;
                  while((i = fr.read()) != -1) str += (char)i;
                  System.out.println(str);
           } catch (FileNotFoundException e) {e.printStackTrace();}
                  catch (IOException e) {e.printStackTrace();}
         }
```

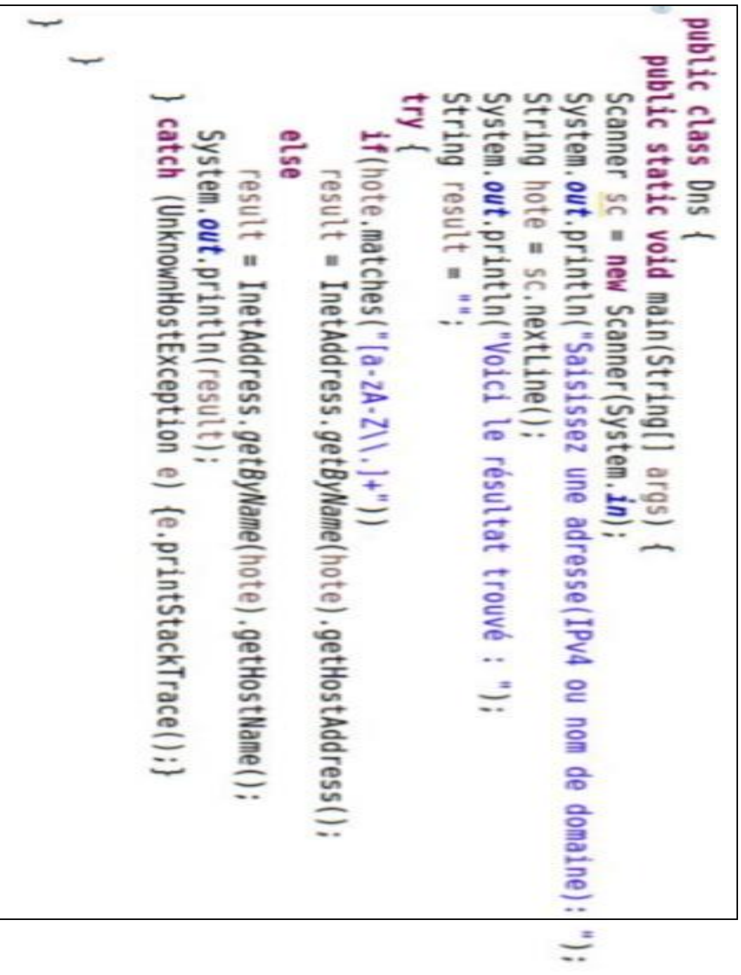
java-nio

```
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
```



```
//address = InetAddress.getByAddress(new byte[]{(byte)192,(byte)168, 2, 44});
//address = InetAddress.getByName("localhost");
//address = InetAddress.getByName("127.0.0.1");
System.out.println("Nom  : " + address.getHostName());
System.out.println("Adresse : " + address.getHostAddress());
System.out.println("Nom canonique : " + address.getCanonicalHostName());
```

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class FileChan {
            public static void main(String[] args) {
             FileInputStream fis;
             FileOutputStream fos;
             BufferedInputStream bis;
             FileChannel fc,fco;
             try {
 fis = new FileInputStream(new File("fichier.txt"));
 fos = new FileOutputStream(new File("fichier2.txt"));
 fc = fis.getChannel();
 fco= fos.getChannel();
int size = (int)fc.size();
 ByteBuffer buf = ByteBuffer.allocate(size);
 fc.read(buf);
buf.flip();
 fco.write(buf);
 byte[] tabByte = buf.array();
 for(byte b : tabByte)System.out.print((char)b);
 } catch (FileNotFoundException e) {e.printStackTrace();}
 catch (IOException e) {e.printStackTrace();}
            }
}
```

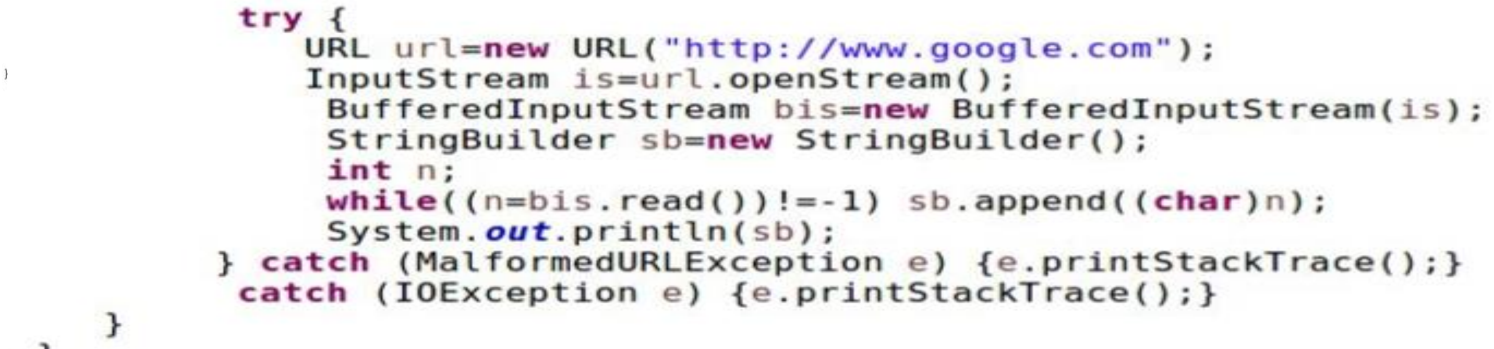


```
public class Dns {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Saisissez une adresse(IPv4 ou nom de domaine): ");
        String hote = sc.nextLine();
        System.out.println("Voici le résultat trouvé : ");
        String result = "";
        try {
            if(hote.matches("[a-zA-Z\\.]+"))
                result = InetAddress.getByName(hote).getHostAddress();
            else
                result = InetAddress.getByName(hote).getHostName();
            System.out.println(result);
        } catch (UnknownHostException e) {e.printStackTrace();}
    }
}
```

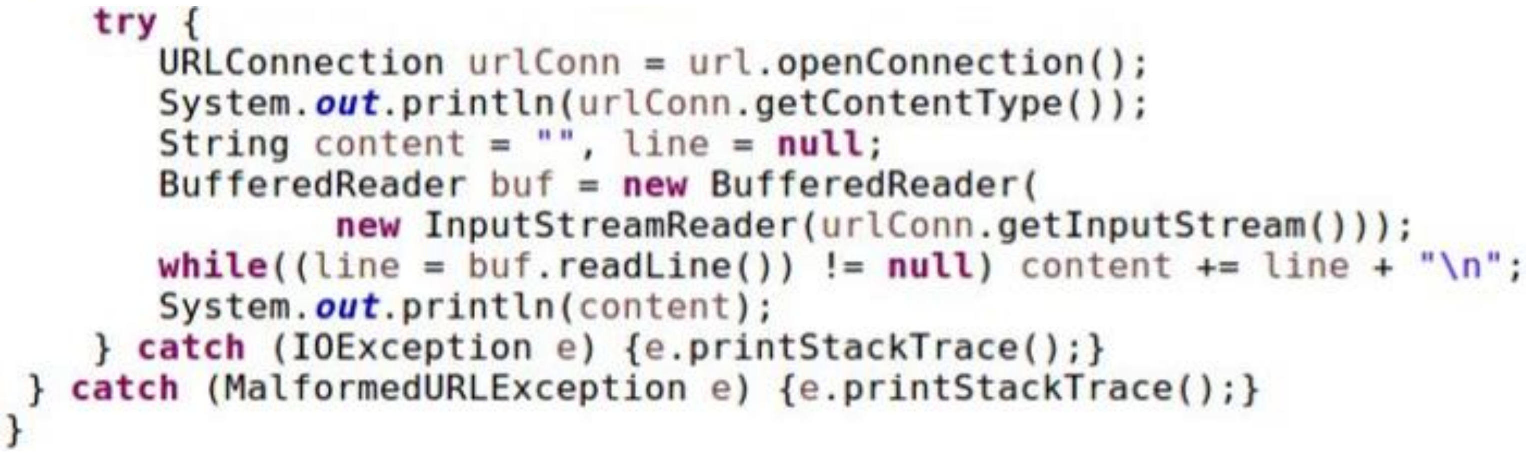
Download html :

a

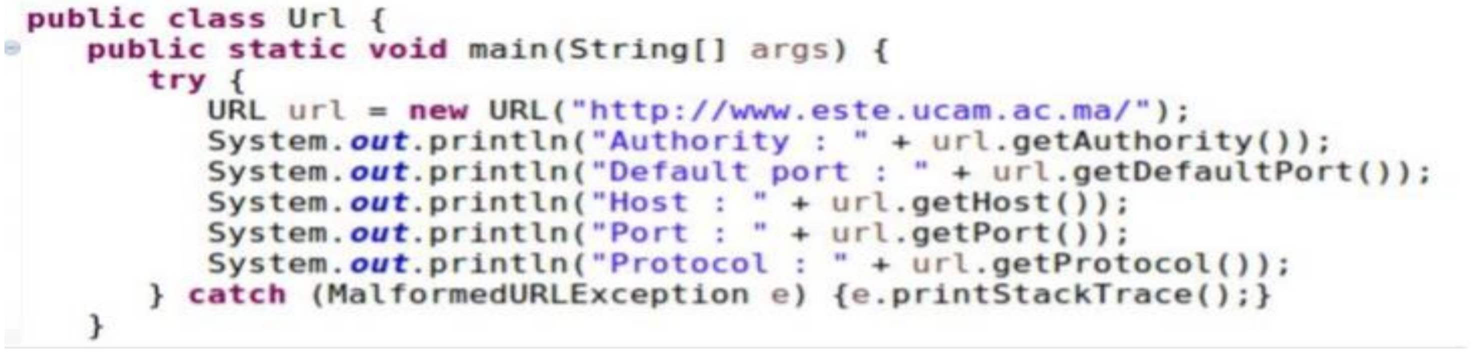
```
try {
URL url=new URL("http://www.google.com");
```



```
                try {
                    URL url=new URL("http://www.google.com");
                    InputStream is=url.openStream();
                     BufferedInputStream bis=new BufferedInputStream(is);
                     StringBuilder sb=new StringBuilder();
                     int n;
                     while((n=bis.read())!=-1) sb.append((char)n);
                     System.out.println(sb);
            } catch (MalformedURLException e) {e.printStackTrace();}
                catch (IOException e) {e.printStackTrace();}
        }
}
```



```
    try {
        URLConnection urlConn = url.openConnection();
        System.out.println(urlConn.getContentType());
        String content = "", line = null;
        BufferedReader buf = new BufferedReader(
                new InputStreamReader(urlConn.getInputStream()));
        while((line = buf.readLine()) != null) content += line + "\n";
        System.out.println(content);
    } catch (IOException e) {e.printStackTrace();}
} catch (MalformedURLException e) {e.printStackTrace();}
}
```



```
public class Url {
    public static void main(String[] args) {
        try {
            URL url = new URL("http://www.este.ucam.ac.ma/");
            System.out.println("Authority : " + url.getAuthority());
            System.out.println("Default port : " + url.getDefaultPort());
            System.out.println("Host : " + url.getHost());
            System.out.println("Port : " + url.getPort());
            System.out.println("Protocol : " + url.getProtocol());
        } catch (MalformedURLException e) {e.printStackTrace();}
    }
```

READ FROM NETWORK

```
public class ReadNetworkFileSock {
            public static void main(String[] args){
                          Socket soc=null;
                          try{
                                     soc = new Socket("www.este.ucam.ac.ma", 80);
                                     String req = "GET / HTTP/1.1\r\n";
                  req += "Host: www.este.ucam.ac.ma\r\n";
                  req += "\r\n";
                  BufferedOutputStream bos = new BufferedOutputStream(soc.getOutputStream());
                  bos.write(req.getBytes());
                  bos.flush();
                  BufferedInputStream bis = new BufferedInputStream(soc.getInputStream());
                  StringBuilder sb=new StringBuilder();
                  int n;
                  while((n = bis.read()) != -1){
                    sb.append((char) n);
                  }
                  System.out.println(sb);
                } catch (UnknownHostException e) {
                  e.printStackTrace();
                } catch (IOException e) {
                  e.printStackTrace();
                }}}
```

SERVER-CLIENT

****

```
************************SERVER**********************
public class Server {
public static void main(String[] args) throws
InterruptedException {
Socket s =null;
try {
ServerSocket Ss =new ServerSocket(71);
s=Ss.accept();
System.out.println("1 server");
boolean done = false;
while (!done)
{  try
{
Thread.sleep(2000);
BufferedOutputStream bos = new
BufferedOutputStream(s.getOutputStream());
bos.write(new String("Server : hello from server").getBytes());
bos.flush();//vider bos;
bos.close();
BufferedInputStream b =new
BufferedInputStream(s.getInputStream());
Thread.sleep(2000);
int n=0;
while ((n=b.read())!=-1) {
System.err.print((char)n);
}s.close();
}catch(IOException ioe)
{  done = true;
}}
} catch (IOException e) {
e.printStackTrace();
}
```

```
*****************CLIENT***********************
public class thread_Client implements Runnable{
public void run() {
Socket soc =null;
try {
soc=new Socket(InetAddress.getLocalHost(),71);
boolean done = false;
while (!done)
{  try
{
BufferedInputStream b =new
BufferedInputStream(soc.getInputStream());
int n=0;
while ((n=b.read())!=-1) {
System.err.print((char)n);
}
Thread.sleep(2000);
b.close();
BufferedOutputStream bos = new
BufferedOutputStream(soc.getOutputStream());
bos.write(new String("CLIENT : hello ESTE").getBytes());
bos.flush();//vider bos;
}
catch(IOException | InterruptedException ioe)
{  done = true;
}
}
} catch (IOException e) {
e.printStackTrace();
}
try {
soc.close();
```

```
 import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.io.BufferedInputStream;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.net.ServerSocket;
import java.net.Socket;
```

```
} catch (IOException e) {
e.printStackTrace();
                                    }
*******************THREAD*************
thread_Client t = new thread_Client();
      Thread t2 = new Thread(t);
            t2.start();
```

```
 **********************************exercice :************************************

class persone implements Serializable{
      String nom;
      String Email;
      String password;
      public  static void update  (ArrayList<persone> l) {
            try {
                  ObjectOutputStream oos  = new ObjectOutputStream(new
BufferedOutputStream(new FileOutputStream(new File("C:\\Users\\toshiba\\Desktop\\othmane2.txt"))));
     for(persone s : l) {
      oos.writeObject(s);
     }
     oos.close();
            } catch (IOException e1) {
                  // TODO Auto-generated catch block
                  e1.printStackTrace();
            }
      }
      public String toString() {
            return "nom : "+this.nom+" Email : "+this.Email+" passwod"+this.password;
      }
}
*************************************classe*************************************
public class Main extends Application {
      @Override
      public void start(Stage primaryStage) throws IOException, ClassNotFoundException {

            List<persone> l = new ArrayList<persone>();
            BufferedInputStream fis = new BufferedInputStream(new FileInputStream(new
File("C:\\Users\\toshiba\\Desktop\\othmane2.txt")));
            ObjectInputStream ois =new ObjectInputStream(fis);
            persone g;
            while(fis.available()>0) {
            g=(persone)ois.readObject();
   System.out.println(g.nom);
   l.add(g);
            }

            try {


                  BorderPane root = new BorderPane();
                  ///////////////////////////FORM1//////////////////////////////
                  GridPane gp= new GridPane();
                  gp.setHgap(10);
                  gp.setVgap(10);
                  Text t1 = new Text("Login");
                  t1.setFont(new Font(24));
                  Text t2 = new Text("name");
```

```
                        Text t3 = new Text("passworde");
                        TextField T1 = new TextField();
                        TextField T2 = new TextField();
                        Button b1 = new Button("login");
                        Button b2 = new Button("Singup");
                        HBox hb = new HBox();
                        hb.getChildren().add(b1);
                        hb.setAlignment(Pos.BASELINE_RIGHT);
                        gp.add(b2, 0, 3);
                        gp.add(hb, 1, 3);
                        gp.add(t1, 1, 0,1,1);
                        gp.add(t2,0,1);
                        gp.add(t3,0,2);
                        gp.add(T1,1,1);
                        gp.add(T2,1,2);
        //     gp.setGridLinesVisible(true);
                        root.setCenter(gp);
                        gp.setAlignment(Pos.CENTER);
                ****************************FORM2 affichage*******************
                        BorderPane gp2= new BorderPane();
                        MenuBar menubar = new MenuBar();
                    SeparatorMenuItem s1 = new SeparatorMenuItem();

                  Menu m = new Menu("File");
                        MenuItem m2 = new MenuItem("Modifier");
                        MenuItem m3 = new MenuItem("Supprimer");
                        m.getItems().addAll(m2,m3);
                         Menu checked = new Menu("check");
                         CheckMenuItem mc1 = new  CheckMenuItem("Modifier");
                         CheckMenuItem mc2= new  CheckMenuItem("Supprimer");
                         checked.getItems().addAll(mc1,mc2);
                         menubar.getMenus().addAll(m,checked);
                         ///**************************************///
                         ObservableList<persone> jours=FXCollections.observableArrayList(l);
                         ListView<persone> listev =new ListView<>(jours);
                                MultipleSelectionModel<persone> lvselect =listev.getSelectionModel();
                                lvselect.selectedItemProperty().addListener(new ChangeListener<persone>() {
                                        @Override
                                        public void changed(ObservableValue<? extends persone> observable,
persone oldValue, persone newValue) {

        System.out.println(listev.getSelectionModel().getSelectedIndex());
                                        }
                                });
                   gp2.setTop(menubar);
                    gp2.setCenter(listev);
                    Button  romove = new Button("romove");
                    Button update = new Button("update");
                    TextField textupdate=new TextField();
                    HBox buttom = new HBox();
                    buttom.getChildren().addAll(romove,update,textupdate);
                    gp2.setBottom(buttom);
                    /////**********************************romove/update---------------------------------------
                    update.setOnAction((e)->{
                         persone p=new persone();
                         p.nom=update.getText().toString();
                         l.remove(listev.getSelectionModel().getSelectedIndex());
                         l.add(p);
                         listev.getItems().set(listev.getSelectionModel().getSelectedIndex(), p);
                         persone.update((ArrayList<persone>)l);
```

```
            })
            romove.setOnAction((e)->{

                 listev.getItems().remove(listev.getSelectionModel().getSelectedIndex());
                 l.remove(listev.getSelectionModel().getSelectedIndex());
                persone.update((ArrayList<persone>)l);
            });
                *********************FORM3-sinup******************************
           GridPane gp1= new GridPane();
                gp1.setHgap(10);
                gp1.setVgap(10);
                Text t11 = new Text("INSCRIR");
                t11.setFont(new Font(24));
                Text t22 = new Text("name");
                Text t21 = new Text("Email");
                Text t33 = new Text("passworde");
                TextField T11 = new TextField();
                TextField T21 = new TextField();
                TextField T22 = new TextField();
                Button login1 = new Button("login");
                Button valde = new Button("VALIDE");
                HBox hb1 = new HBox();
                hb1.getChildren().add(login1);
                hb1.setAlignment(Pos.BASELINE_RIGHT);
                gp1.add(t11, 1, 0,1,1);
                gp1.add(t22,0,1);
                gp1.add(t33,0,2);
                gp1.add(T11,1,1);
                gp1.add(T22,1,2);
                gp1.add(T21,1,3);
                gp1.add(t21,0,3);
                gp1.add(hb1, 1, 4);
                gp1.add(valde, 0, 4);
    gp1.setAlignment(Pos.CENTER);
******************************************evenment************************************
                b1.setOnAction((e)->{
                        for(persone s : l) {
         if (s.nom.equals(T1.getText())) {
                 root.setCenter(gp2);}
         System.out.println("boocle"+s.nom+"t1 "+T1.getText().toString());
        }
                });
                b2.setOnAction((e)->{

                                root.setCenter(gp1);
                });
                valde.addEventHandler(MouseEvent.MOUSE_CLICKED, (e)->{
                        persone p=new persone();
                        p.nom=T11.getText().toString();
                        p.Email=T22.getText().toString();
                        p.password=T21.getText().toString();
                        l.add(p);
                        try {

                                ObjectOutputStream oos  = new ObjectOutputStream(new
BufferedOutputStream(new FileOutputStream(new File("C:\\Users\\toshiba\\Desktop\\othmane2.txt"))));
               l.add(p);
               for(persone s : l) {
                oos.writeObject(s);
               }
            oos.close();
                } catch (IOException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
                }
            });
            login1.addEventHandler(MouseEvent.MOUSE_CLICKED,(e)->{

                root.setCenter(gp);
            } );

    //  gp.setGridLinesVisible(true);
            root.setCenter(gp);
            gp.setAlignment(Pos.CENTER);
            Scene scene = new Scene(root,400,400);

    scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
********************************************main*******************************
    public static void main(String[] args) {

        launch(args);
    }
}

*******************************************with table view ***************************

    TableView<Contact> tvContacts;
    tvContacts = new TableView<Contact>(contactList);
    TableColumn<Contact, String> prnm = new TableColumn<>("LastName");
    prnm.setCellValueFactory(new PropertyValueFactory<>("prenom"));
    tvContacts.getColumns().add(prnm);
    TableColumn<Contact, String> nom = new TableColumn<>("FirstName");
    nom.setCellValueFactory(new PropertyValueFactory<>("nom"));
    tvContacts.getColumns().add(nom);
    TableColumn<Contact, String> num = new TableColumn<>("Phone");
    num.setCellValueFactory(new PropertyValueFactory<>("Tele"));
    tvContacts.getColumns().add(num);
    tvContacts.setPrefWidth(300);
    tvContacts.setPrefHeight(160);
        TableView.TableViewSelectionModel<Contact> tvSelContact =  tvContacts.getSelectionModel();
    tvSelContact.selectedIndexProperty().addListener(new ChangeListener<Number>()
    {
      public void changed(ObservableValue<? extends Number> selected,
          Number oldVal, Number newVal) {
        int index = (int)newVal;
        rep.setText("Full Name : "+contactList.get(index).getNom()+" "+contactList.get(index).getPrenom()+"
- Phone : "
            +contactList.get(index).getTele());
      }
    });
    rep.setFont(Font.font("Arial", 14));
    root.getChildren().addAll(title,tvContacts, rep);
```

**********************server*************************************

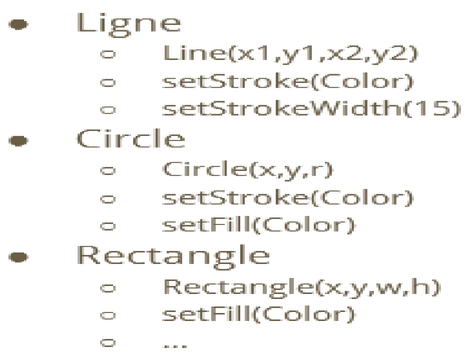
```
package application;
import javafx.application.Application;
import java.io.*;
import java.net.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;
import javafx.application.Platform;
import javafx.event.EventHandler;
import javafx.geometry.* ;
import javafx.stage.Stage;
import javafx.scene.control.*;
import javafx.scene.input.*;
import javafx.scene.layout.*;
import javafx.scene.text.* ;
import javafx.scene.effect.*

public class Serveur extends Application {
public String DateNow()
{
//Methode pour Prendre le temps du message
DateFormat df=new SimpleDateFormat("HH:mm:ss");
Date d=new Date();
return df.format(d);
}
void  SendTo(VBox A,VBox B,String t) {
//Methode pour affichez le message qui arrive
Label lvide=new Label();
Label lrempli=new Label(t);
//style au message
DropShadow d=new DropShadow();
d.setOffsetX(3.0);
d.setOffsetY(3.0);
d.setColor(Color.LIGHTSLATEGREY);
//position et font du message
lrempli.setPadding(new Insets(5,5,5,5));
lrempli.setFont(Font.font ("Verdana", 20));
lrempli.setEffect(d);
//l'insertion des elements aux roots
A.getChildren().add(lvide);
B.getChildren().add(lrempli);
}
void ReceiveFrom(VBox A,VBox B,String t) {
//Methode pour affichez notre message
Label lvide=new Label();
Label lrempli=new Label(t);
//style au message
DropShadow d=new DropShadow();
d.setOffsetX(3.0);
d.setOffsetY(3.0);
d.setColor(Color.LIGHTGREEN);

//position et font du message
lrempli.setPadding(new Insets(5,5,5,5));
lrempli.setFont(Font.font ("Cursive", 20));
```

```

public class Client extends Application {

public String DateNow()
{
//Methode pour Prendre le temps du message
DateFormat df=new SimpleDateFormat("HH:mm:ss");
Date d=new Date();
return df.format(d);

}

void  SendTo(VBox A,VBox B,String t) {
//Methode pour affichez le message qui arrive
Label lvide=new Label();
Label lrempli=new Label(t);
//style au message
DropShadow d=new DropShadow();
d.setOffsetX(3.0);
d.setOffsetY(3.0);
d.setColor(Color.LIGHTSLATEGREY);
//position et font du message
lrempli.setPadding(new Insets(5,5,5,5));
lrempli.setFont(Font.font ("Verdana", 20));
lrempli.setEffect(d);

//l'insertion des elements aux roots
A.getChildren().add(lvide);
B.getChildren().add(lrempli);
}
void ReceiveFrom(VBox A,VBox B,String t) {

//Methode pour affichez notre message
Label lvide=new Label();
Label lrempli=new Label(t);
//style au message
DropShadow d=new DropShadow();
d.setOffsetX(3.0);
d.setOffsetY(3.0);
d.setColor(Color.LIGHTGREEN);
//position et font du message
lrempli.setPadding(new Insets(5,5,5,5));
lrempli.setFont(Font.font ("Cursive", 20));
lrempli.setEffect(d);

//l'insertion des elements aux roots

A.getChildren().add(lvide);
B.getChildren().add(lrempli);
}
@Override
public void start(Stage primaryStage) {
try {
/*   CODE JAVA FX
* */
//Creation des root Element
```

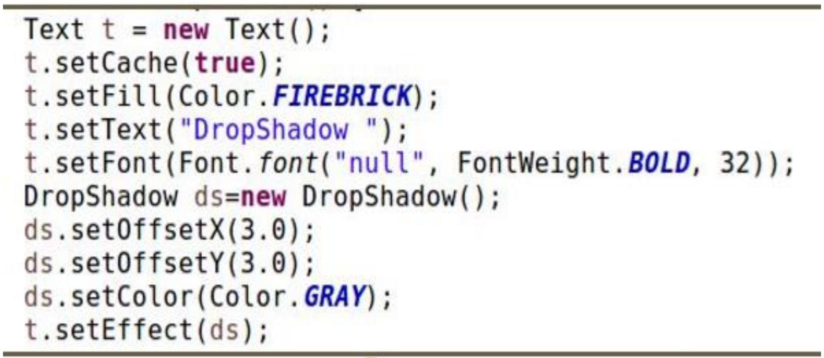
```
lrempli.setEffect(d);
//l'insertion des elements aux roots
A.getChildren().add(lvide);
B.getChildren().add(lrempli);
}
void  ReceiveFromServer(VBox A,VBox B,String t) {

A.getChildren().add(new Label(" "));
B.getChildren().add(new Label(t));
}
@Override
public void start(Stage primaryStage) {
try {
/*   CODE JAVA FX
* */
//Creation des root Element
AnchorPane Ap=new AnchorPane();
VBox hb=new VBox();
AnchorPane vb=new AnchorPane();
            BorderPane Bd=new BorderPane();
VBox hbText=new VBox();
hbText.setSpacing(10);
VBox hbText2=new VBox();
hbText2.setSpacing(10);
Bd.setRight(hbText);
Bd.setLeft(hbText2);

//creation des elements
TextField txEcrit=new TextField();
Button btnEnvoyer=new Button("Envoyer");

vb.getChildren().addAll(txEcrit,btnEnvoyer);
//positionnement des elemnt dans anchore
vb.setLeftAnchor(txEcrit,1.0);
vb.setRightAnchor(txEcrit,80.0);
vb.setRightAnchor(btnEnvoyer, 1.0);
Ap.getChildren().addAll(Bd,vb);
Ap.setTopAnchor(Bd,1.0);
Ap.setLeftAnchor(Bd, 1.0);
Ap.setLeftAnchor(vb,1.0);
Ap.setRightAnchor(Bd, 1.0);
Ap.setRightAnchor(vb,1.0);
Ap.setBottomAnchor(vb,1.0);
Ap.setBottomAnchor(Bd,35.0);
//thread pour ne pas selectioné partie d'affichage
Platform.runLater(new Runnable() {
public void run() {
txEcrit.requestFocus();
}
});
Scene scene=new Scene(Ap,800,500);
primaryStage.setScene(scene);
primaryStage.show();
primaryStage.setTitle("Chat Room");
/*   CODE Socket
//creation du socketserver
ServerSocket socketServer;
Socket socketduserver;
PrintWriter out;
```

```
AnchorPane Ap=new AnchorPane();
VBox hb=new VBox();
AnchorPane vb=new AnchorPane();
BorderPane Bd=new BorderPane();
VBox hbText=new VBox();
hbText.setSpacing(10);
VBox hbText2=new VBox();
hbText2.setSpacing(10);
Bd.setRight(hbText);
Bd.setLeft(hbText2);
TextField txEcrit=new TextField();
Button btnEnvoyer=new Button("Envoyer");

/*txAffichage.setEditable(false);
txAffichage.setAlignment(Pos.TOP_LEFT); // pour
aligner l'affichage de text
txAffichage.setFocusTraversable(false);
txAffichage.setMouseTransparent(true); //pour ne pas
selectionner un controle
*/
vb.getChildren().addAll(txEcrit,btnEnvoyer);
//positionnement des elemnt dans anchore
vb.setLeftAnchor(txEcrit,1.0);
vb.setRightAnchor(txEcrit,80.0);
vb.setRightAnchor(btnEnvoyer, 1.0);
Ap.getChildren().addAll(Bd,vb);
Ap.setTopAnchor(Bd,1.0);
Ap.setLeftAnchor(Bd, 1.0);
Ap.setLeftAnchor(vb,1.0);
Ap.setRightAnchor(Bd, 1.0);
Ap.setRightAnchor(vb,1.0);
Ap.setBottomAnchor(vb,1.0);
Ap.setBottomAnchor(Bd,35.0);
//thread pour ne pas selectioné partie d'affichage
Platform.runLater(new Runnable() {

@Override
public void run() {
txEcrit.requestFocus();
}
});
Scene scene=new Scene(Ap,800,500);
primaryStage.setScene(scene);
primaryStage.show();
primaryStage.setTitle("Chat Room");
/*   CODE Socket
try {
//creation socket client
Socket socket;
PrintWriter out;
BufferedReader in;
socket=new Socket("127.0.0.1",5000);//listen port
meme du server
out=new PrintWriter(socket.getOutputStream());
//Lanceur des message
in=new BufferedReader(new
InputStreamReader(socket.getInputStream()));
//reader des message
//a chque click en fait thread d'envoie
BufferedReader in;
try {
socketServer=new ServerSocket(500);//Port libre
socketduserver=socketServer.accept();//l'attent connexion d'un client
out=new PrintWriter(socketduserver.getOutputStream());//pour envoyer
in=new BufferedReader(new InputStreamReader(socketduserver.getInputStream()));//pour recevoir

//a chaque click on lance thread d'envoi des message
btnEnvoyer.addEventHandler(MouseEvent.MOUSE_CLICKED,evnt->{
ReceiveFrom(hbText,hbText2,DateNow()+"->"+txEcrit.getText());
Platform.runLater(new Runnable() {
String msg;
public void run() {
msg=txEcrit.getText();
out.println(DateNow()+"->"+msg);//l'envoi avec la date actuel
out.flush();
}
});
});
//Timer pour ecouter chaque message Arrive
Timer m=new Timer();
TimerTask mt=new TimerTask() {
String msg="";
@Override
public void run() {
while(true){
try {
msg=in.readLine() ;
Platform.runLater(new Runnable() {
@Override
public void run() {
SendTo(hbText2,hbText, msg); //Recevoir des messages
}
});
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
}
};
m.scheduleAtFixedRate(mt, 1000,1000);

}catch(Exception e) {e.printStackTrace();}

} catch(Exception e) {
e.printStackTrace();
}
}
public static void main(String[] args) {
launch(args);
}
}
```

```
btnEnvoyer.addEventHandler(MouseEvent.MOUSE_CLICKED,evnt->{
ReceiveFrom(hbText,hbText2,DateNow()+"->"+txEcrit.getText());
Platform.runLater(new Runnable() {
String msg;
@Override
public void run() {

msg=txEcrit.getText();
out.println(DateNow()+"->"+msg);
out.flush();

}
});

});
//chaque quelque second on verifie s il y a des message a lire
Timer m=new Timer();
TimerTask mt=new TimerTask() {
String msg="";
@Override
public void run() {
while(true){
try {
msg=in.readLine() ;
Platform.runLater(new Runnable() {

@Override
public void run() {
SendTo(hbText2,hbText, msg); //l'affichage de message
}
});
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
}
};
m.scheduleAtFixedRate(mt, 1000,1000);
}catch(Exception e) {e.printStackTrace
} catch(Exception e) {
e.printStackTrace();
}
}
public static void main(String[] args) {
launch(args);
}
}
```

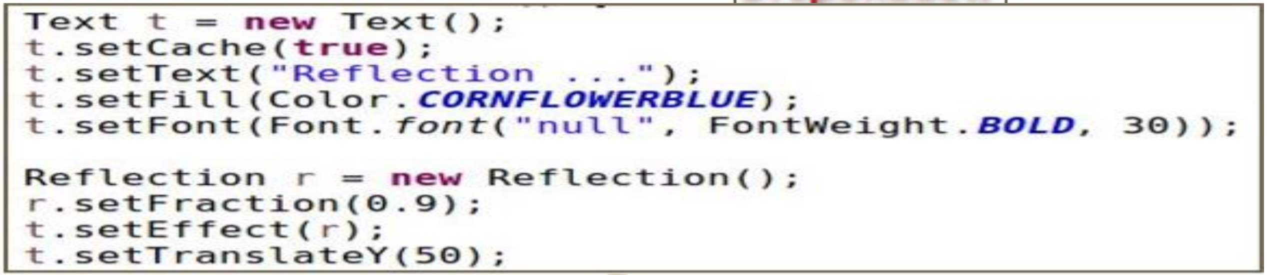


- Ligne
  - Line(x1,y1,x2,y2)
  - setStroke(Color)
  - setStrokeWidth(15)
- Circle
  - Circle(x,y,r)
  - setStroke(Color)
  - setFill(Color)
- Rectangle
  - Rectangle(x,y,w,h)
  - setFill(Color)
  - ...



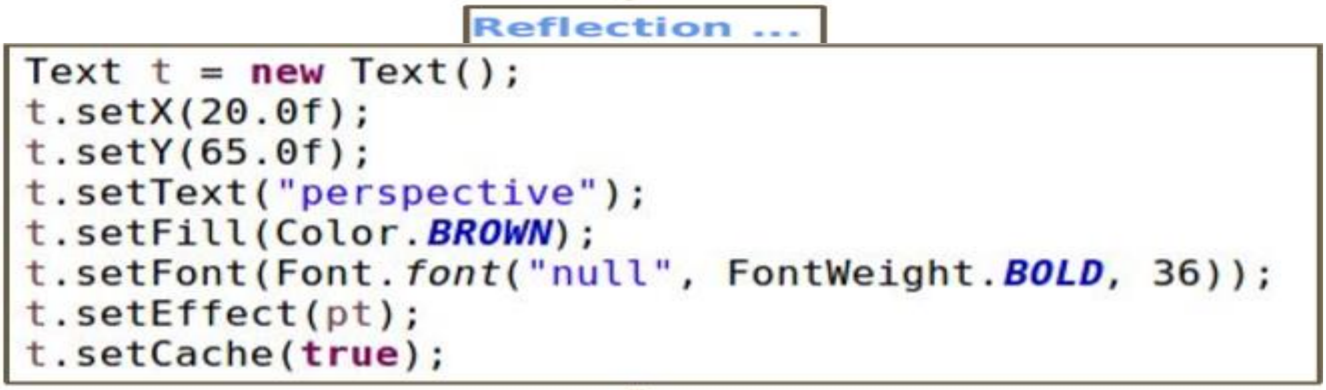
```
Text t = new Text();
t.setCache(true);
t.setFill(Color.FIREBRICK);
t.setText("DropShadow ");
t.setFont(Font.font("null", FontWeight.BOLD, 32));
DropShadow ds=new DropShadow();
ds.setOffsetX(3.0);
ds.setOffsetY(3.0);
ds.setColor(Color.GRAY);
t.setEffect(ds);
```

DropShadow



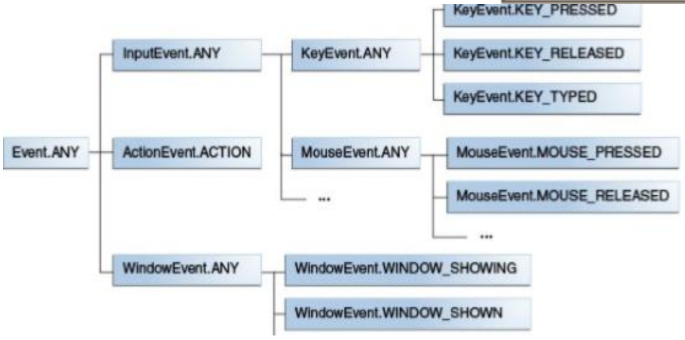
```
Text t = new Text();
t.setCache(true);
t.setText("Reflection ...");
t.setFill(Color.CORNFLOWERBLUE);
t.setFont(Font.font("null", FontWeight.BOLD, 30));

Reflection r = new Reflection();
r.setFraction(0.9);
t.setEffect(r);
t.setTranslateY(50);
```

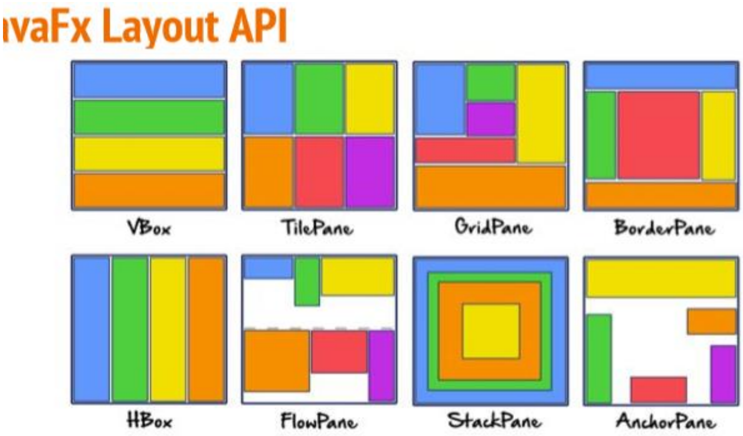
Reflection ...



```
Text t = new Text();
t.setX(20.0f);
t.setY(65.0f);
t.setText("perspective");
t.setFill(Color.BROWN);
t.setFont(Font.font("null", FontWeight.BOLD, 36));
t.setEffect(pt);
t.setCache(true);
```

perspective





```
public class Main extends Application {
@Override
public void start(Stage primaryStage) {
try {

BorderPane root = new BorderPane();
Scene scene = new Scene(root,400,400);
scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
primaryStage.setScene(scene);
primaryStage.show();
} catch(Exception e) {
e.printStackTrace();

public static void main(String[] args) {
launch(args);
```

```
    }
}
```