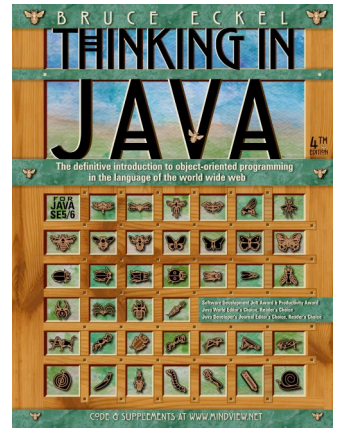
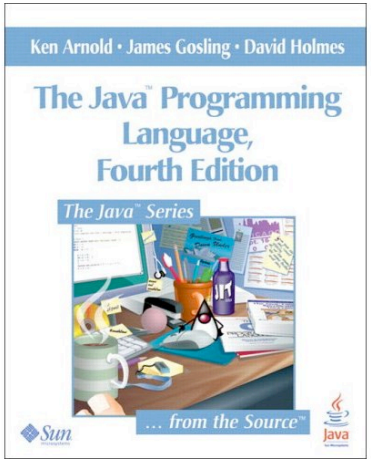
Hugues Fauconnier
hf@liafa.jussieu.fr


BRUCE ECKEL
THINKING IN
JAVA
4TH Edition
The definitive introduction to object-oriented programming in the language of the world wide web
CODE @ SUPPLEMENTS AT WWW.MINDVIEW.NET


Ken Arnold • James Gosling • David Holmes
The Java™ Programming Language, Fourth Edition
The Java™ Series
… from the Source™
Sun
Java

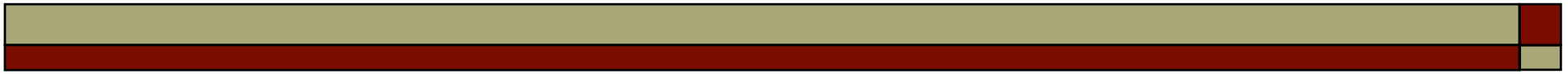
Réutiliser le logiciel

- *polymorphisme*
  -
    - *redéfinir*
    -
    -
      -
      -
      -
        -
      - *transtypage*
    -

Liaison dynamique

d'entiers

- *abstrait*
- *concrète*



- *client vendeur*
- *contrat spécification*
- *interface*
- *pré-condition*
- *post-condition*
- *héritage*

```
package pile;

abstract class Pile <T>{
    abstract public T empiler(T v);
    abstract public T dépiler();
    abstract public Boolean estVide();
}
```

```
package pile;
import java.util.EmptyStackException;
import java.util.Vector;
public class MaPile<T> extends Pile<T>{
    private Vector<T> items;
    // Vector devrait être remplacé par ArrayList
    public MaPile() {
        items =new Vector<T>(10);
    }
    public Boolean estVide(){
        return items.size()==0;
    }
    public T empiler(T item){
        items.addElement(item);
        return item;
    }
    //…
```

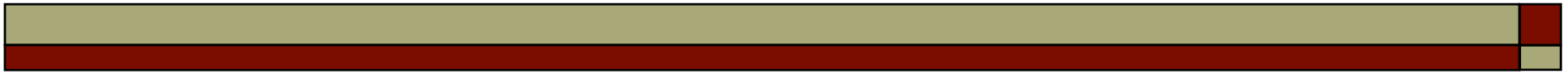
```
//…
public synchronized T dépiler(){
        int len = items.size();
        T item = null;
        if (len == 0)
            throw new EmptyStackException();
        item = items.elementAt(len - 1);
        items.removeElementAt(len - 1);
        return item;
    }
}
```

```
package pile;
import java.util.LinkedList;
public class SaPile<T> extends Pile<T> {
    private LinkedList<T> items;
    public SaPile(){
        items = new LinkedList<T>();
    }
    public Boolean estVide(){
        return items.isEmpty();
    }
    public T empiler(T item){
        items.addFirst(item);
        return item;
    }
    public T dépiler(){
        return items.removeFirst();
    }
}
```

```
public class PileInteger extends Pile<Integer>{
    private Integer[] items;
    private int top=0;
    private int max=100;
    public PileInteger(){
        items = new Integer[max];
    }
    public Integer empiler(Integer item){
        if (this.estPleine())
            throw new EmptyStackException();
        items[top++] = item;
        return item;
    }
    //…
```

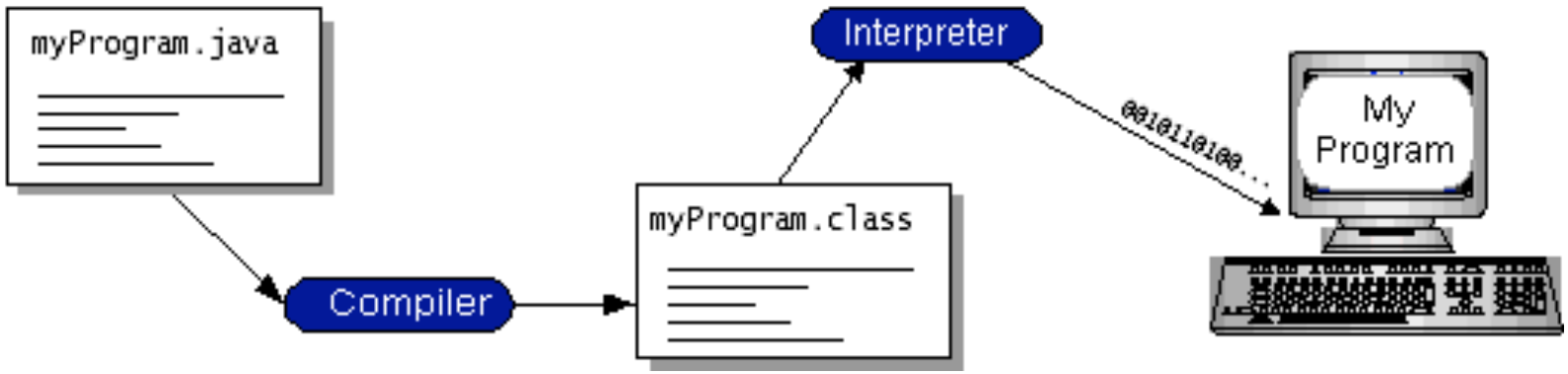
```
public synchronized Integer dépiler(){
        Integer item = null;
        if (this.estVide())
            throw new EmptyStackException();
        item = items[--top];
        return item;
    }
    public Boolean estVide(){
        return (top == 0);
    }
    public boolean estPleine(){
        return (top == max -1);
    }
    protected void finalize() throws Throwable {
        items = null; super.finalize();
    }
}
```

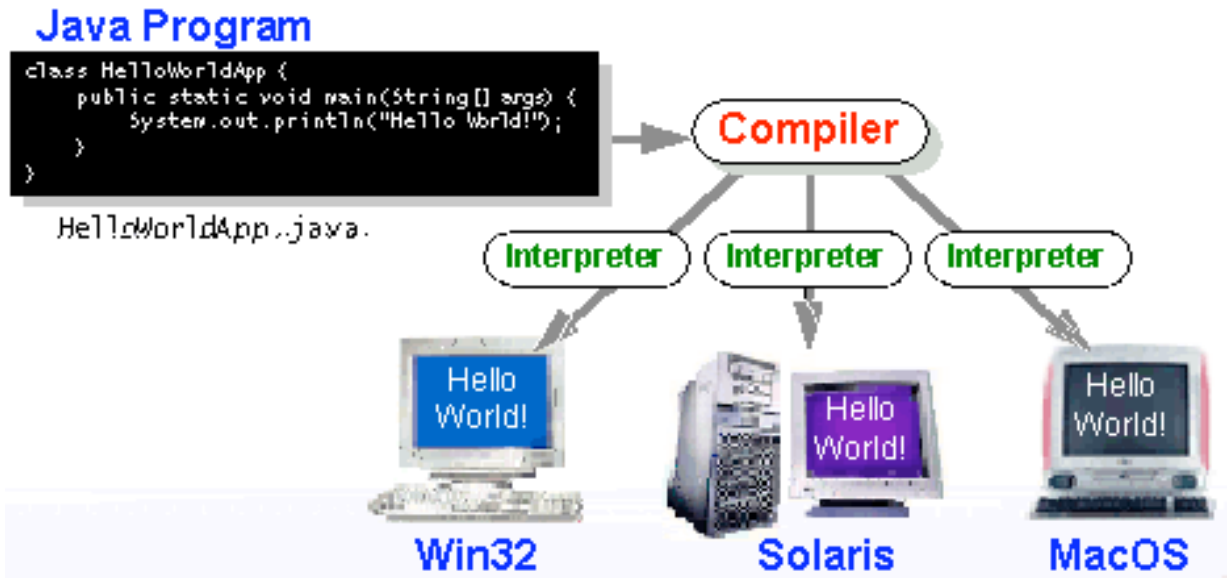
```
package pile;
public class Main {
    public static void vider(Pile p){
        while(!p.estVide()){
            System.out.println(p.dépiler());
        }
    }
public static void main(String[] args) {
        MaPile<Integer> p1= new MaPile<Integer>();
        for(int i=0;i<10;i++)
            p1.empiler(i);
        vider(p1);
        SaPile<String> p2= new SaPile<String>();
        p2.empiler("un");
        p2.empiler("deux");
        p2.empiler("trois");
        vider(p2);
    }
```



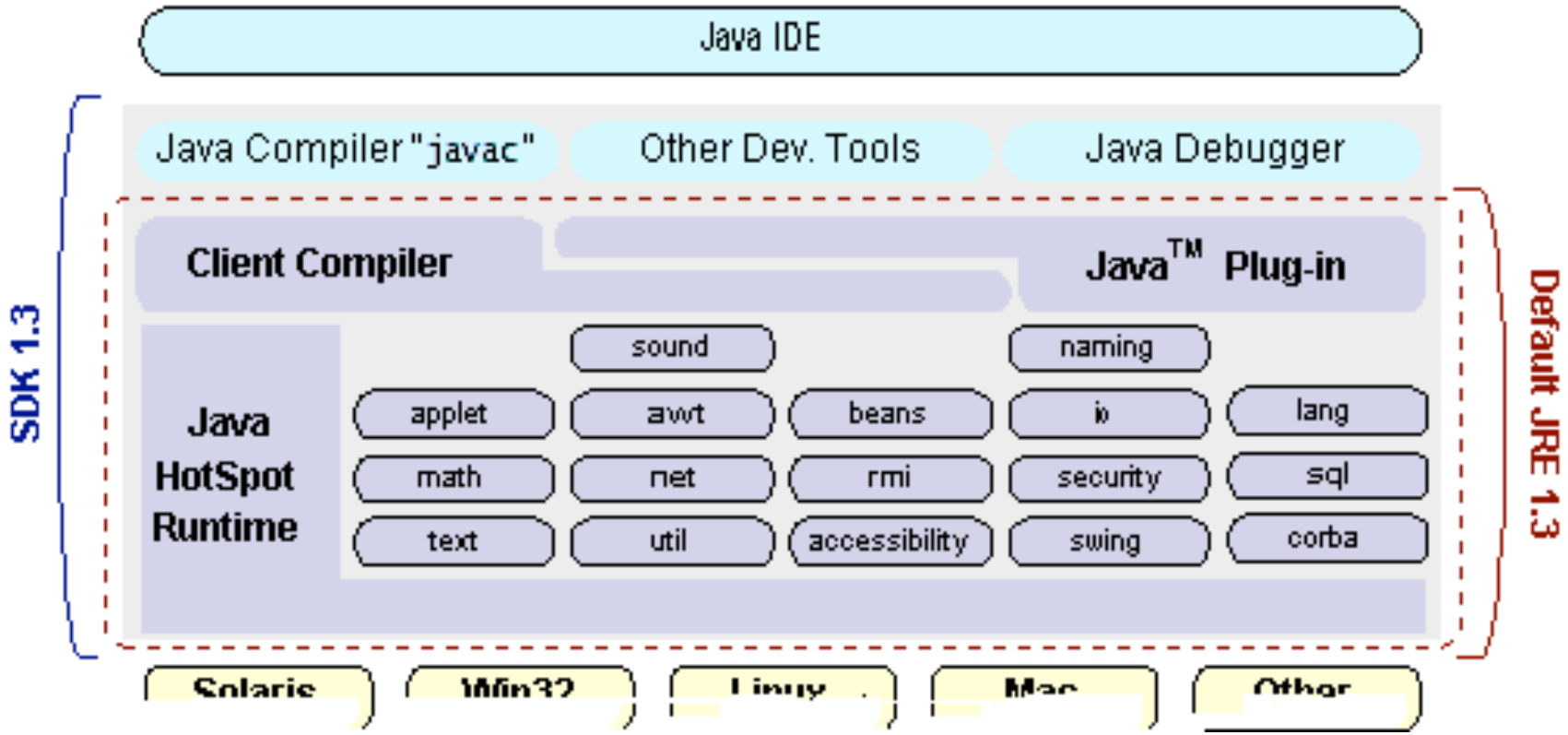
```
public static void main(String[]);
```

- *Langage interprété et byte code*

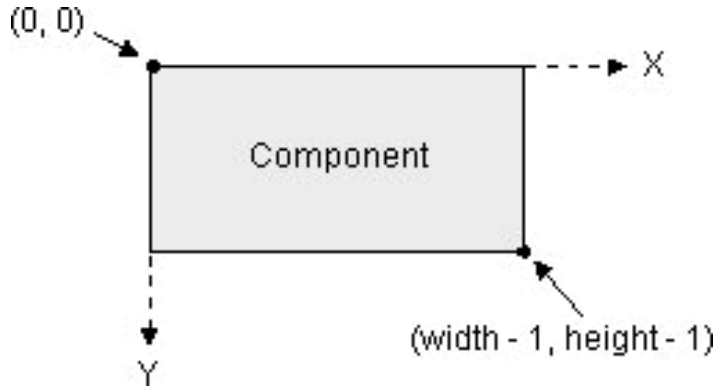

myProgram.java
Compiler
myProgram.class
Interpreter
My Program


Java Program
class HelloWorldApp {
public static void main(String[] args) {
System.out.println("Hello World!");
}
}
HelloWorldApp.java
Compiler
Interpreter
Interpreter
Interpreter
Hello World!
Hello World!
Hello World!
Win32
Solaris
MacOS

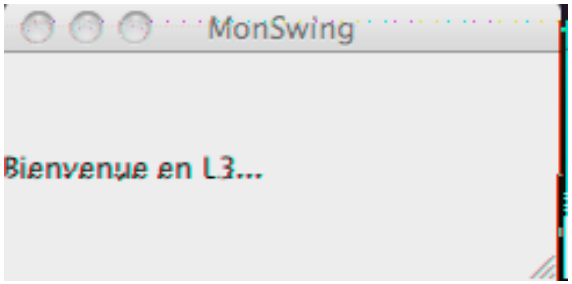

Java API
Java Platform
Java Virtual Machine
Hardware-Based Platform


Java IDE
Java Compiler "javac"
Other Dev. Tools
Java Debugger
Client Compiler
Java™ Plug-in
Java HotSpot Runtime
sound
naming
applet
awt
beans
io
lang
math
net
rmi
security
sql
text
util
accessibility
swing
corba
SDK 1.3
Default JRE 1.3

- paint start
  init paint
- Graphics g paint
  - drawString
  -



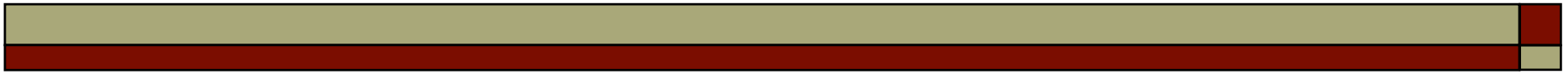
```
/**
 * Une application  basique... avec interface graphique
 */
import javax.swing.*;
public class MonSwing {
    private static void creerFrame() {
        //Une formule magique...
        JFrame.setDefaultLookAndFeelDecorated(true);
        //Creation d'une Frame
        JFrame frame = new JFrame("MonSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         //Afficher un message
        JLabel label = new JLabel("Bienvenue en L3...");
        frame.getContentPane().add(label);
        //Afficher la fenêtre
        frame.pack();
        frame.setVisible(true);
    }
 public static void main(String[] args) {
     creerFrame();
    }
}
```



```
public static void main(String[] args) {
     // sortie avec printf ou
     double a = 5.6d ;
     double b = 2d ;
     String mul = "multiplié par" ;
     String eq="égal";
     System.out.printf(Locale.ENGLISH,
                "%3.2f X %3.2f = %6.4f \n", a ,b , a*b);
     System.out.printf(Locale.FRENCH,
           "%3.2f %s %3.2f %s %6.4f \n", a, mul,b eq,a*b);
      System.out.format(
         "Aujourd'hui %1$tA, %1$te %1$tB,"+
          " il est: %1$tH h %1$tM min %1$tS \n",
                Calendar.getInstance());
       // System.out.flush();
```

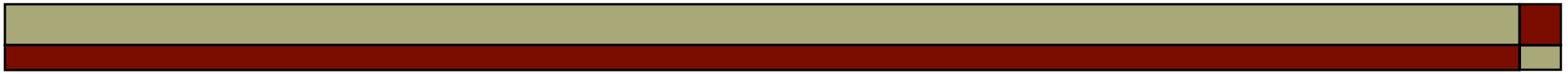
```
Scanner sc = new Scanner(System.in);
for(boolean fait=false; fait==false;){
    try {
       System.out.println("Répondre o ou O:");
       String s1 =sc.next(Pattern.compile("[Oo]"));
              fait=true;
    } catch(InputMismatchException e) {
    sc.next();
    }
}
if (sc.hasNextInt()){
    int i= sc.nextInt();
    System.out.println("entier lu "+i);
}
System.out.println("next token :"+sc.next());
sc.close();
```

```
if (sc.hasNextInt()){
    int i= sc.nextInt();
    System.out.println("entier lu "+i);
}
System.out.println("next token :"+sc.next()); sc.close();
String input = "1 stop 2 stop éléphant gris stop rien";
Scanner s = new(Scanner(input).useDelimiter("\\s*stop\\s*");
        System.out.println(s.nextInt());
        System.out.println(s.nextInt());
        System.out.println(s.next());
        System.out.println(s.next());
        s.close();
    }
```

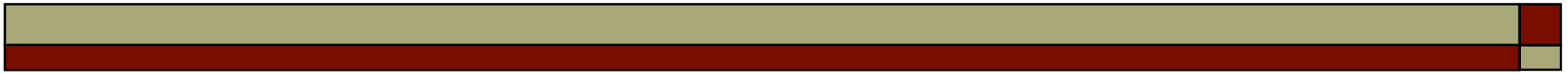


-
  -
    -
    -
  -
-
  -
    -
    -
-

Object



-
  -
  - public
  - abstract
  - final
  - Strictfp



-
  -
  -
    - private
    - protected
    - public
    - package
  - static
  - final
  - transient
  - Volatile
-
- new