- :
  - ? ?
- ( )
  - , , , ,
- :
  - , , ,
- :
  - , , , ,
- :
- :
- : , ,
-
  - , , ,
-
  - , , ,
- :
  - , , , /
-
  -
-
  - , ,
-

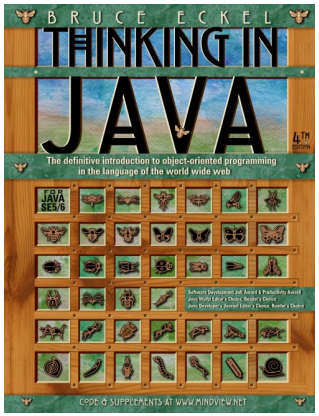
: :// . . - - . / / / / 12-13/ / 3. .

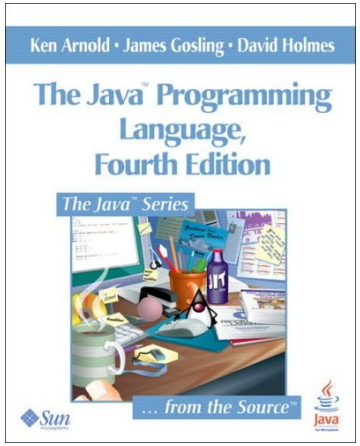
: :// . . - - . / / / . ? =

B

- (
  >= 1.5) 1.7
- :
  - :// . /B / 4
  - , 4 B
  - :// . . / /

- :
  - A A , ,





A)

- :
  -
  - :
  -
- :
  -
  - R ili e le l giciel
  -
    - ?

- :
  - (60)
  - (70) A
  - : +
    (80)
  - : ,

B)

:

( )

- ,
  - ,
    - .
  - ,
  - ,
  - ,
- :
  - ,
  - :
  - ,
  - .

- :
  - A , B:
    - A (())
    - A B
    - :
      - B ;
      - = ; (, B)
      - . ()
        - A!
      - ,
    - (:)

- :
  - ’ ‘ ()’
    (, A B) (=
)

Liai d ami e

  - (,)
    .
    :
    ,
    ,
    ,
    (
)

)

?

( )

(=

)

( )

?

- ,
  - A , (, ,)
  -
  - (' -)

?

( )

-
- ( )
  ,
-
  ,
-
  ,
-
  , ( )
  ,
-
  – ,
  ( )
-
  : ,
  ( )

)

: -> B

: ->

: ->

: ->

(:) <=> (( ))

A

,

(())

((,))

((,))=(,)

-

,

(’)

```
package pile;

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

- package:
- abstract: , ,
- public: ,
-

:

- A ArrayList
- A LinkedList
- A Integer

Integer

```
package pile;
import java.util.EmptyStackException;
import java.util.ArrayList;
public class MaPile<T> extends Pile<T>{
    private ArrayList<T> items;
       public MaPile() {
         items =new ArrayList<T>(10);
    }
    public Boolean estVide(){
         return items.isEmpty();
    }
    public T empiler(T item){
         items.add(item);
         return item;
    }
    //…
```

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

A

```
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();
    }
}
```

?

-
- A , , , ,
  ( )
  ,
-

```
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);
    }
```

) :

- .java
- ( )

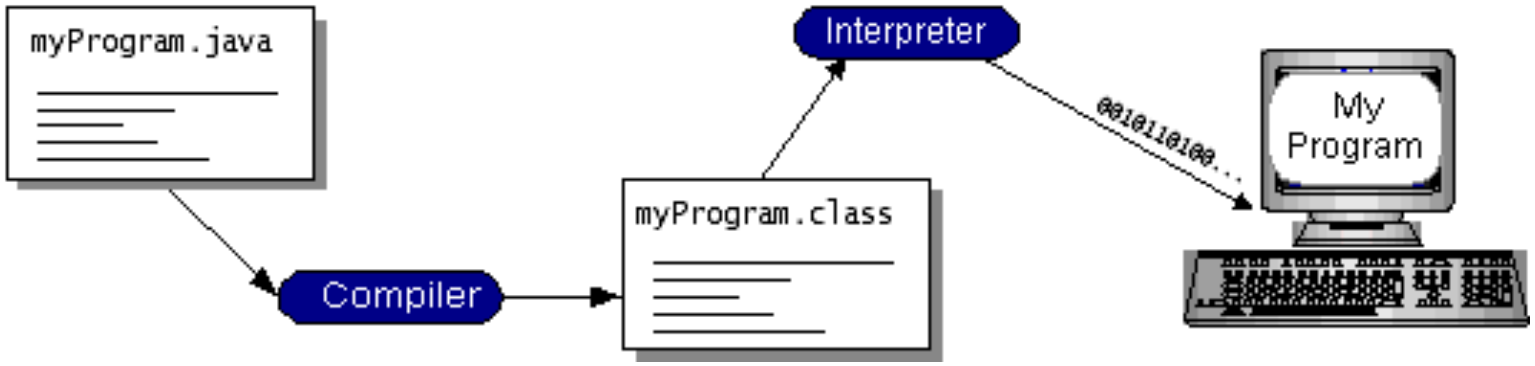
(.java)

-
  ```
  public static void main(String[]);
  ```
  - main ’
- .class
-

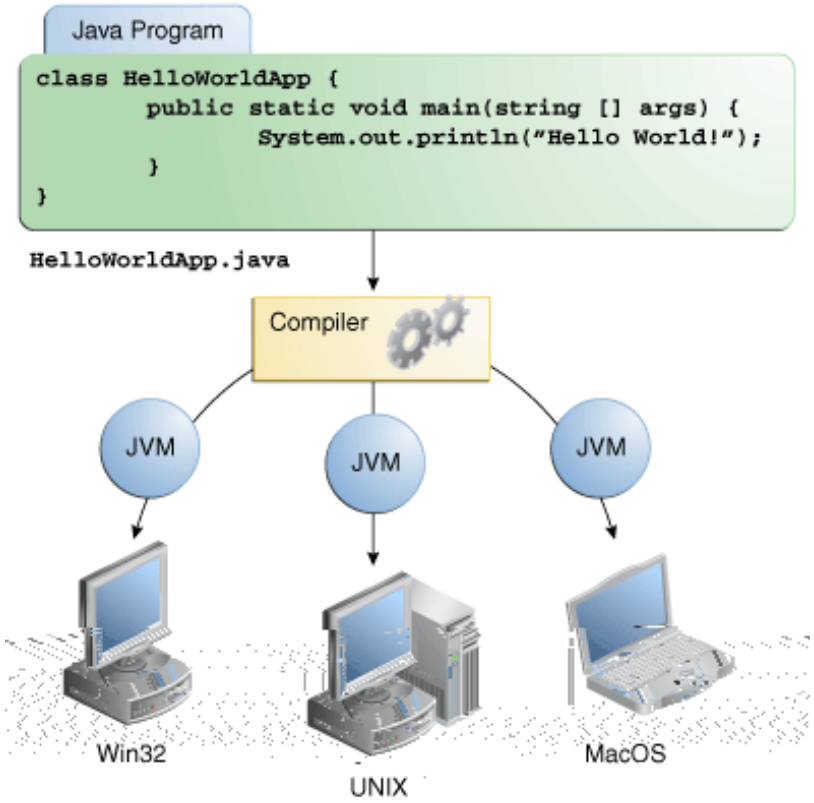
- ’ :
  - “ ”! ( )
  -
  -
  -
  - ( )
    -
  - ! (!)
    - -
  - -
  - (B) , ,
  - 7 ( )
  - : :// . . /

- . (
).
-

`javac`
`java`

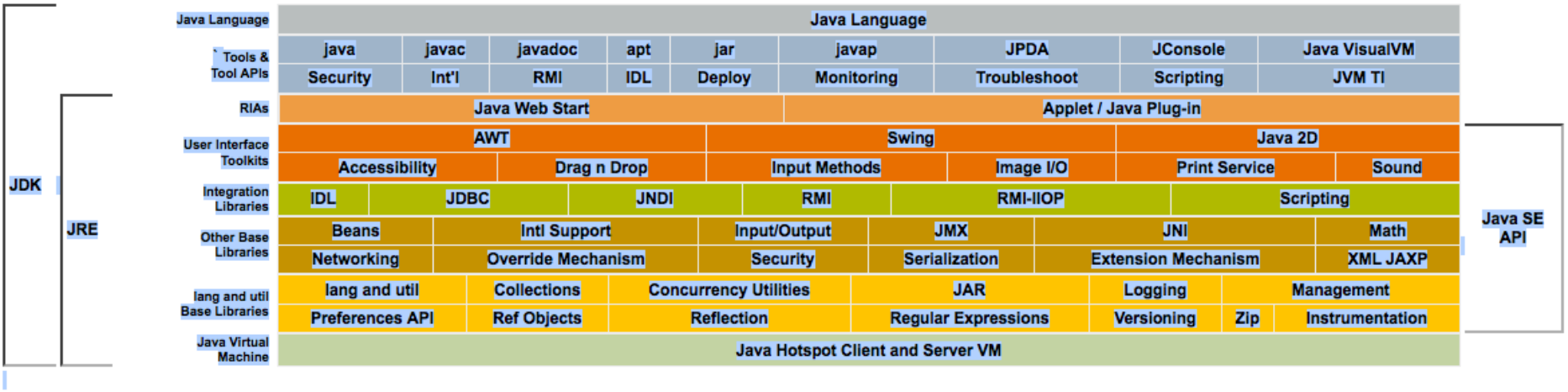

myProgram.java
Compiler
myProgram.class
Interpreter
My Program

- A :
  - -
    ( )
- :


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

- : - ’
  (
  ,)
-
- (A):


Java Platform
Java API
Java Virtual Machine
Hardware-Based Platform



Java SE 6 API Documentation

A :

- Appli.java:

```
/**
 * Une application  basique...
 */
class Appli {
    public static void main(String[] args) {
     System.out.println("Bienvenue en L3...");
                //affichage
    }
}
```

- Appli.java
- :
  - javac Appli.java
- Appli.class( )
- :
  - java Appli
- A !!!
  - ($ A)

```
Exception in thread "main" java.lang.NoClassDefFoundError:
```

  - -> !
  - A A -

- /* */ //
-
  - (=)
  - (
)
- main:
  - public static void main(String[] arg)
    - public
    - static
    - Void
    - String
      ’
  -

- System
  - out System
  - printl System.out
  - out

    PrintStream

    .
    - println

- :
  ( )
  ,
- ( )
  - :
    ( )
    .
  - ,
    :
    .
  - :
    (
)
  - ,
    :
    ( )

A :

- A B
  - ( ) B
  - ,
  - A :
    - A :

- A . :

```
/**
 * Une applet  basique...
 */
import java.applet.Applet;
import java.awt.Graphics;
public class MonApplet extends Applet {
    public void paint(Graphics g){
   g.drawString("Bienvenue en en L3...", 50,25);
    }
}
```

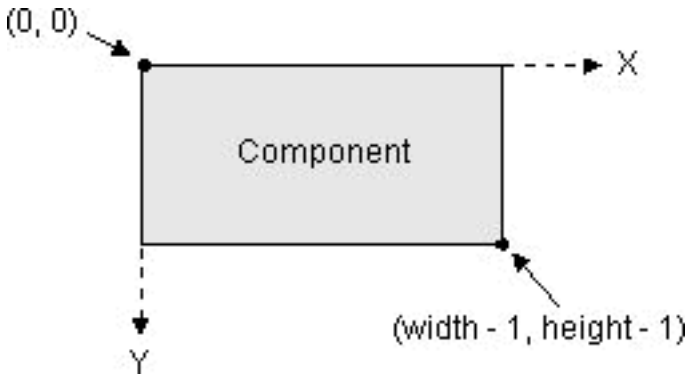
:

- :
  - .
  - .
  - ( )
  - .
- *:
  - . . ;
  - .
    - A ,
    - . .A

:

- A ,
- A :
  - A :
    - A
    - ( ))
  - !!

- A init. paint start paint, ’ .
- Graphics g paint , .
  - drawString (’) ,
  - 50, 25: (0,0) , (,) .



,

,

,

,

- B . :

```
<HTML>
<HEAD>
<TITLE> Une petite applet </TITLE>
<BODY>
<APPLET CODE='MonApplet.class' WIDTH=200
   Height=50>
</APPLET>
</BODY>
</HTML>
```

- :
- :
  - <HTML> </HTML>
  - :
    - page de hf
- :

```
<APPLET CODE='MonApplet.class' WIDTH=200
   Height=50>
</APPLET>
```

-
- , - ,
- A
- ,
- ,
- A
-
-

- 1.5 6 ,
-
-
  ( )
  - , ..

-

```
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();
```

5.60 2.00 = 11.2000

5,60 2,00 11,2000

A ' , 10 , : 15

31 01

```
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();
    }
```

- :
- 1
- 2
-
-