



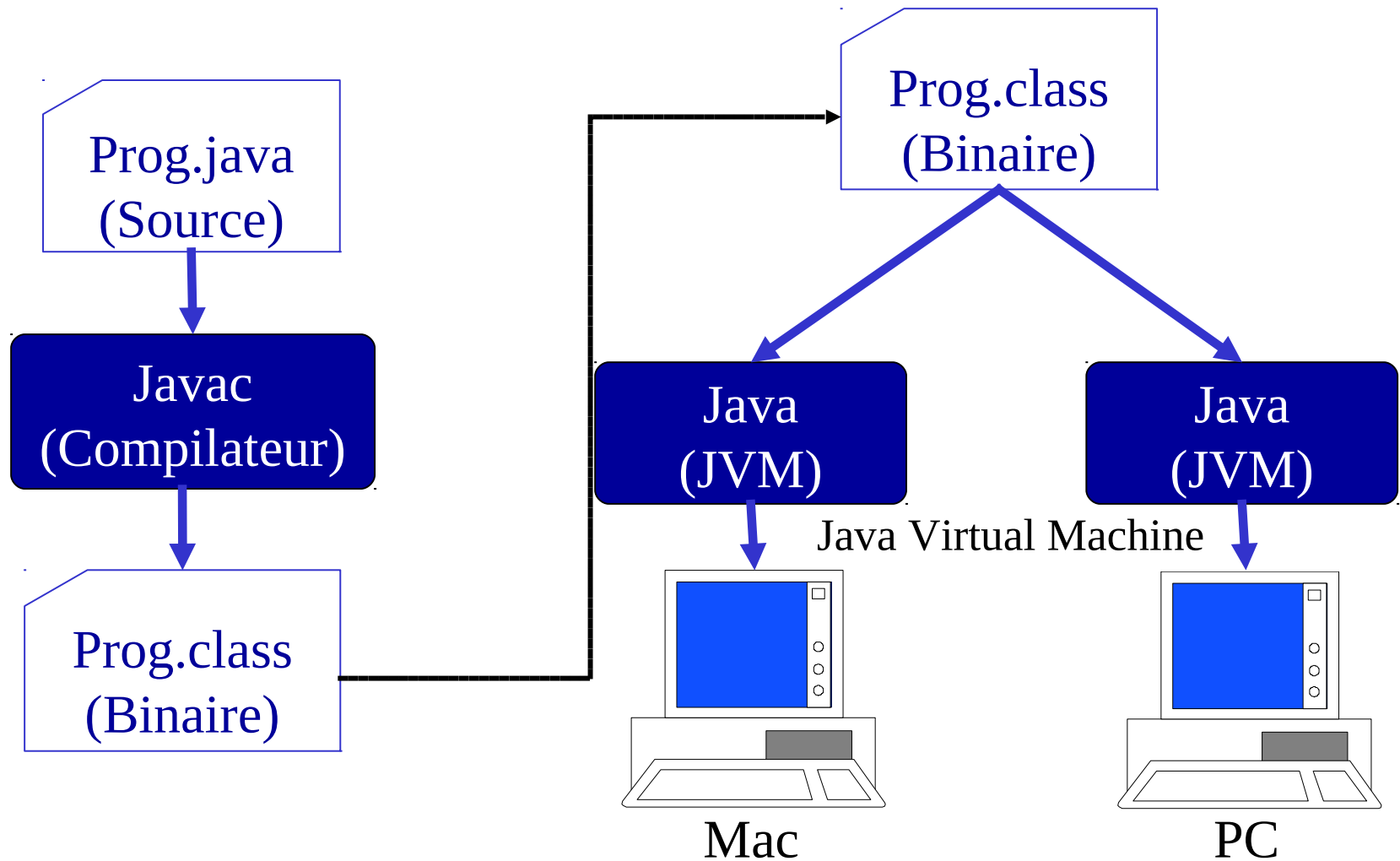
# Le langage Java

# Fiche commerciale

---

- ~~Simple~~ (proche de C)
- Orienté objet
- Distribué
- Interprété
- ~~Robuste~~
- Sécurisé
- Indépendant de la plate-forme
- Portable
- ~~Haute performance~~
- Multithread
- Dynamique

# Principe



# Plate-forme

---

- La plate forme Java est composée de
  - La machine virtuelle (JVM)
  - API Java (Application Programming Interface)
- Elle est disponible (gratuitement) sous :
  - Windows, Solaris, Linux, MacOs
  - <http://java.sun.com/j2se>

# Tradition

---

```
/**
 * Premier programme java
 */
public class HelloWorld {
    public static void main(String[] args){
        //afficher "Hello World!"
        System.out.println("Hello World !");
    }
} //fin classe
```

# Tradition

---

```
/**
 * Premier programme java
 */
public class HelloWorld {
    public static void main(String[] args){
        //afficher "Hello World!"
        System.out.println("Hello World !");
    }
} //fin classe
```

# Tradition

---

```
/**
 * Premier programme java
 */
public class HelloWorld {
    public static void main(String[] args){
        //afficher "Hello World!"
        System.out.println("Hello World !");
    }
} //fin classe
```

# Tradition

---

```
/**
 * Premier programme java
 */
public class HelloWorld {
    public static void main(String[] args){
        //afficher "Hello World!"
        System.out.println("Hello World !");
    }
} //fin classe
```



# Tradition

---

```
/**
 * Premier programme java
 */
public class HelloWorld {
    public static void main(String[] args){
        //afficher "Hello World!"
        System.out.println("Hello World !");
    }
} //fin classe
```

# Tradition

---

```
>javac HelloWorld.java
```

```
>java HelloWorld
```

```
Hello World !
```

```
>
```

# Class

---

```
public class Fibonacci {  
    ...  
}
```

Type de la class




# Class

---

- Modificateurs
  - Public
    - accessible partout
    - une classe publique par fichier .java
    - nom du fichier = nom de la classe.java
  - Private
    - Accessible uniquement dans le fichier où la classe est définie
  - Final
    - Pas d'héritage
  - Abstract
    - Non instanciable

# Donnée d'une classe

```
public class Fibonacci {  
    static final int MAX_INDEX= 10;  
    ...  
}
```



**Attributs de la donnée**

- static
  - Variable de classe
- final
  - Non modifiable (constante)

# Donnée d'une classe

```
public class Fibonacci {  
    static final int MAX_INDEX= 10;  
    ...  
}
```



Type de la donnée

# Type de base

---

- boolean = 'True' ou 'False'
- int = Entier 32bits
- char = Caractère (ex: 'A')
- float, double = Réel sur 32 ou 64 bits
- short, long, byte = Entier 16, 64 ou 8 bits
- void = le type « rien »

# Types élémentaires Objets

---

- boolean
- int
- char

**Non  
Objet**



- Boolean
- Integer
- Character
- String

**Objet**



# Exemple

---


```
public class Variables {  
    public static void main(String params[]) {  
        // Définition de deux variables entières i et j  
        // i sera de plus initialisée avec la valeur 15  
        int i=15 , j;  
        // Définition d'une chaîne de caractères  
        String chaine = "Cours Java";  
        // On donne une valeur à j  
        j = 10;  
        // On utilise ces variables en les affichant  
        System.out.println("i = " + i + " et j = " + j);  
        System.out.println("chaine = " + chaine);  
    }  
}
```

# Tableaux

---

`int a, b[];` // a est un simple entier, b est un tableau NON alloué.

---

`int monTableau[];`  
`nomTableau = new int[10];`  `int[] monTableau2 = new int[10];`

---

`int tableau[][] = new tableau[10][];` // Tableau à 2 dimensions

---

`int[] a, b[];`  `int a[], b[][];`  
// a est un tableau, b est un tableau à deux dimensions

# Accessibilité

```
public class Fibonacci {  
    static final int MAX_INDEX= 10;  
    public static void main(String[] args) {  
        ...  
    }  
}
```

**Accessibilité / Visibilité d'une fonction**

# Accessibilité / Visibilité

---

- public
  - Visible partout
- protected
  - Visible dans un package
- Private
  - Visible dans une classe

# Les opérateurs

---

- Comme en langage C !
  - $a == b$  : a est égal à b
  - $a >= b$  : a est supérieur ou égal à b
  - $a || b$  : a ou b (évaluation partielle)
  - $a | b$  : a ou b (évaluation totale)
  - $a ^ b$  : a xor b
  - $! a$  : non logique de a
  - $\sim a$  : non arithmétique de a

# Structures de contrôle

---

- Comme en langage C !
  - for
  - while
  - do ... while
  - if ...else
  - switch case...
  - etc.

# E/S standard

---

- Objet System
  - Lecture : `System.in`
  - Ecriture : `System.out` ou `System.err`

# Comment écrire

---

- Objet System
  - **System.out.print**("Hello World\n");  
*OU*
  - **System.out.println**("Hello World");
  - **System.err.println**("Error 4012 ! ☹️");



# Comment lire une chaîne

---

- Objet System

```
byte Nom[] = new byte[10];
String NomStr;
try {
    System.in.read(Nom, 0, 10);
} catch (java.io.IOException e) {
    System.out.println( « Pb de saisi !!! » );
    System.exit(1);
};
NomStr = new String(Nom);
```

# Comment lire un caractère

---

- Objet System

```
int Char;
```

```
try {
```

```
    Char = System.in.read();
```

```
} catch (java.io.IOException e) {
```

```
    System.out.println( « Pb de saisi !!! »  
    );
```

```
    System.exit(1);
```

```
};
```

# Comment convertir une chaîne en entier

---

```
String MaChaine;
```

```
int x;
```

```
...
```

```
x = Integer.parseInt(MaChaine);
```