

TP JAVA – Programmation Orientée Objets

Rafael COLARES

Maitre de Conférences ISIMA

Bureau D104 – email:rafael.colares_borges@uca.fr

Quelques annonces

- Le cours de Java fait peau neuve
 - CM + TP
 - Evaluation
- Présence en TP
- Hétérogénéité
 - Premiers TPs assez simples

TP 1

- Contenu:

- ✓ Premier programme en Java
- ✓ Manipulation des variables de type non-objet
- ✓ Manipulation d'objets de classes prédéfinies
- ✓ Manipulation d'objets de classes customisées

(Compilation + exécution)

(Type primitif)

(String, StringBuffer, etc)

(Constructeurs, encapsulation)

- Aller voir <https://perso.isima.fr/loic/java>

Premier programme

```
/* ma première classe */  
public class Exemple {
```

= 1^{ère} classe

Nom de la classe

=

Nom du fichier source

= méthode (de classe) / point d'entrée

```
public static void main(String[] argv) {
```

```
// afficher un message
```

```
System.out.println("Bonjour");
```

paramètres

```
}
```

Fichier source (texte) *Exemple.java*

Bonnes pratiques:

- Une classe par fichier (sauf pour quelques cas très spécifiques)
- Nom de classe avec première lettre MAJ
- Organisation en packages (on verra plus tard)

Compilation + exécution

1. Compiler le programme

```
javac Exemple.java
```

.exe ?

.class ?

2. Lancer le programme

```
java Exemple
```

Pourquoi pas ./Exemple ?

Compilation + exécution

1. Compiler le programme

```
javac Exemple.java
```

.exe ?

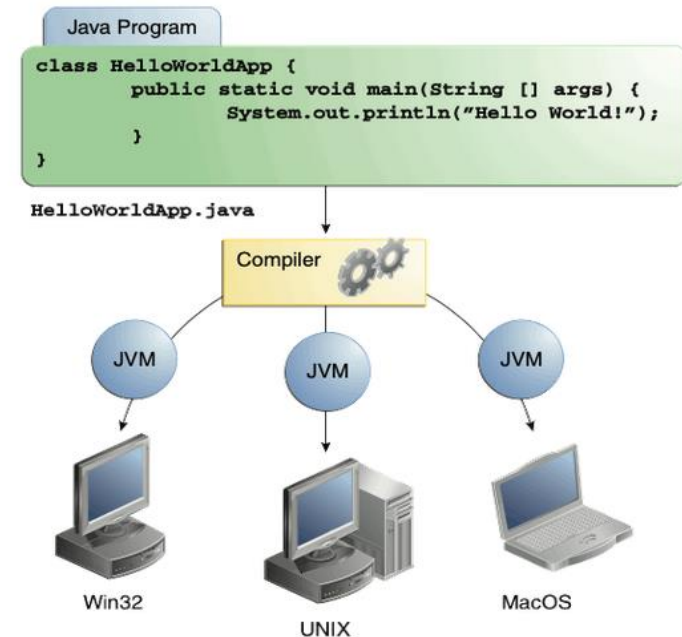
.class ?

2. Lancer le programme

```
java Exemple
```

Pourquoi pas ./Exemple ?

- Fichier compilé : .class
- Pseudo-code (byte-code) **interprété** par la *Java Virtual Machine* (JVM)
- ≠ Code machine



Avant d'aller voir StringBuffer :

```
public class ExempleString {  
  
    public static void main(String[] argv) {  
        String str1 = "Bonjour";  
        String str2 = "Bonjour";  
        System.out.println(str1 == str2);  
  
        str2 = "Hello";  
        System.out.println(str1 == str2)  
  
        String str3 = new String("Bonjour");  
        System.out.println(str1 == str3);  
    }  
}
```



Avant d'aller voir StringBuffer :

```
public class ExempleString {  
  
    public static void main(String[] argv) {  
        String str1 = "Bonjour";  
        String str2 = "Bonjour";  
        System.out.println(str1 == str2);  
  
        str2 = "Hello";  
        System.out.println(str1 == str2)  
  
        String str3 = new String("Bonjour");  
        System.out.println(str1 == str3);  
    }  
}
```



VRAI

FAUX

FAUX

- String is immutable
 - Java String Pool
 - “Interning”
 - Allocation explicite avec **new**
-
- Avantages:
 - Economie en mémoire
 - + de sécurité (thread-safe)

*Java Strings are Immutable - Here's What That Actually Means
Coding With John (Youtube)*

Chaines de caractères modifiables : StringBuffer / StringBuilder