

Rappels, compiler mon programme Java

Fabrice.Kordon@lip6.fr



Un petit «Hello World»



Le source

```
// Ma première classe

public class HelloWorld{

    public static void main(String []args){
        System.out.println("Bonjour le monde!");
    }
}
```

Sur le terminal...

```
$ ls
total 8
8 HelloWorld.java
$ javac -Xlint HelloWorld.java
$ ls
total 16
8 HelloWorld.class  8 HelloWorld.java

$ java HelloWorld
Bonjour le monde!
```


Un petit «Hello World»



Le source

```
// Ma première classe

public class HelloWorld{

    public static void main(String [] args){
        System.out.println("Bonjour le monde");
    }
}
```

Version de Javac
Exemples compilés avec
1.8.0_45 (MacOS 10.11.6)

Sur le terminal...

```
$ ls
total 8
8 HelloWorld.java
$ javac -Xlint HelloWorld.java
$ ls
total 16
8 HelloWorld.class 8 HelloWorld.java

$ java HelloWorld
Bonjour le monde!
```


A propos des E/S

Deux variantes

```
// Ma première classe
```

```
public class HelloWorld{  
  
    public static void main(String []args){  
        System.out.println("Bonjour le monde!");  
    }  
}
```

```
// Ma première classe
```

```
public class HelloWorld{  
  
    public static void main(String []args){  
        char c;  
        c = 'e';  
        System.out.printf("Bonjour le mond%c!\n",c);  
    }  
}
```


Éléments importants

Convention de nommage

- Pour la Classe Toto
 - ▶ **Fichier Toto.java**
- Gestion des dépendances automatiques
 - ▶ **Usage potentiel de -classpath (-cp)**

Plusieurs classes dans le même fichier?

- Possible
 - ▶ **Pas terrible d'un point de vue «Génie Logiciel»**
 - ▶ **Pratique juste pour de petits exemples**
- Garde-fou
 - ▶ **Les classes publiques doivent être localisées dans des fichiers distincts**
- La classe «principale» doit donner son nom au fichier
 - ▶ **C'est celle qui contient la méthode «main»**
 - ▶ **Le compilateur sait trouver le reste**

Remarques sur l'environnement de travail

Choix de l'environnement

-  C'est le vôtre
-  Mes exemples en «ligne de commande»

Attention!

-  Les environnements cachent parfois des choses...
-  Vous devez **comprendre!**

Mais...

-  Pas de hot-line en TME garantie sur l'environnement que vous avez choisi!



À propos de Java

Un «ancien» langage

-  1991 \Rightarrow Sun Microsystems (Oak)
-  1995 \Rightarrow Première implémentation publique (1.0)

▶ 1.8 = «Java SE 8» en 2014

-  2006/2007 \Rightarrow Compilateur + machine virtuelle en GPL
-  2010 \Rightarrow Rachat de Sun par Oracle



Objet

-  Impératif

Indépendant de la plateforme

-  Si JVM installée (courant), byte-code interprété

Multi-threadé

-  Plusieurs fils d'exécution...
-  Possiblement réparti

▶ Discuté plus tard (et surtout en M)

À propos de Java

Un «ancien» langage

1991

1995

► 1.8 = «Java 8»

2006/2009

2010

Objet

Impératif

Indépendant

Si JVM installé

Multi-threads

Plusieurs threads

Possiblement

Langage bien connu (TIOBE, juillet 2016)...



Language	2016	2011	2006	2001	1996	1991	1986
Java	1	1	1	3	21	-	-
C	2	2	2	1	1	1	1
C++	3	3	3	2	2	2	7
C#	4	5	6	11	-	-	-
Python	5	6	7	25	20	-	-
PHP	6	4	4	9	-	-	-
JavaScript	7	9	8	7	23	-	-
Visual Basic .NET	8	103	-	-	-	-	-
Perl	9	8	5	4	3	-	-
Ruby	10	10	22	32	-	-	-

► Discuté plus tard (et surtout en M)