

BASES : projet *Hello World* !

Débuter un programme

- 1) Importer les *packages* nécessaires
- 2) Ecrire le corps du programme (déclaration de variables, création des méthodes, création de l'IU...)
- 3) Ecrire la méthode `main()` :

```
• public static void main (String x[] ) {  
• ... // votre code  
• }
```

Note : *x* représente la matrice d'*arguments* c'est pourquoi il sera le plus souvent remplacé par « arguments » ou « args ». Vous pouvez cependant choisir n'importe quel nom.

Note : tous les programmes (pas les applets...) doivent avoir une méthode *main()*, car c'est ce qui les définit en tant que programme !

Débuter une applet

- 1) Importer les *packages* nécessaires
- 2) Déclarer la classe principale :

```
• public class x extends java.applet.Applet {  
• ... // votre code  
• }
```

Note : *x* représente le nom de votre programme ainsi que le nom de son fichier source (*x.java* et *x.class*).

Note : pour programmer des applets en Java 2, on remplace *java.applet.Applet* par *JApplet*.

Il est cependant recommandé de programmer vos applets en java 1.1 ou mieux, en java 1.02 car les dernières version des principaux navigateurs (IE5, Communicator 4.7) ne supportent pas java 2. Java 2 est supporté par le navigateur *HotJava Browser* de SUN (<http://www.sun.java.fr>), ainsi que par l'appletviewer (!), outil qui vous permettra de tester vos applets (inclus dans les JDK).

Insérer des commentaires

Les commentaires sont des lignes dans le programme qui ne sont pas prises en compte par le compilateur. Ils servent seulement à accroître la lisibilité du code source. Ils sont le plus souvent en vert dans les environnements de développement comme Kawa, JBuilder, ou VisualCafé.

Voici comment on crée un commentaire :

- Ce code est pris en compte par le compilateur // ce code n'est plus pris en compte par le compilateur
- /* Les deux slash ne permettent que d'écrire des commentaires sur 1 seule ligne, alors que cette méthode avec une belle étoile et encore un slash permettent d'écrire sur plusieurs lignes des romans ou des fables de centaines de pages commençant par Il était une fois et finissant par le sempiternel ils se marièrent et eurent beaucoup d'enfants..... */

Résumons : // sur 1 seule ligne

/* ... */ sur plusieurs lignes

Ecrire du texte

Le titre de ce paragraphe ne veut rien dire car en programmation, « on n'écrit pas de texte », on affiche sur un périphérique de sortie (par défaut l'écran de l'ordinateur !) du texte ; on peut aussi afficher du texte dans un composant d'une IU (exemple, l'intitulé du menu « fichier » de votre navigateur). Dans notre exemple, on affichera du texte dans la zone de sortie : ici, la fenêtre de ligne de commande (MS-DOS pour les PC sous Windows) dans laquelle a été lancée l'application. (cf. rubrique JDK/exécution d'un programme)

Pour afficher du texte en ligne de commande, on utilise la méthode :

- `System.out.println(" texte à afficher ");`
- `System.out.print(" les deux méthodes sont équivalentes ");`

Il est plus difficile (c'est relatif...) d'afficher du texte dans une applet, car on doit l'afficher dans un composant (une applet est lancée dans une sorte de cadre inséré dans le navigateur) graphique de l'applet, les deux méthodes ci-dessus ne fonctionnent pas. On utilise la méthode *paint()* de l'applet pour définir l'affichage et la méthode *drawString()* pour écrire le texte :

- `public void paint(Graphics screen) {`
- `screen.drawString(" Bonjour ", 5, 40);`
- `}`

Note: 5 et 40 sont les coordonnées du texte dans un repère qui a pour origine de coin haut-gauche de l'écran.

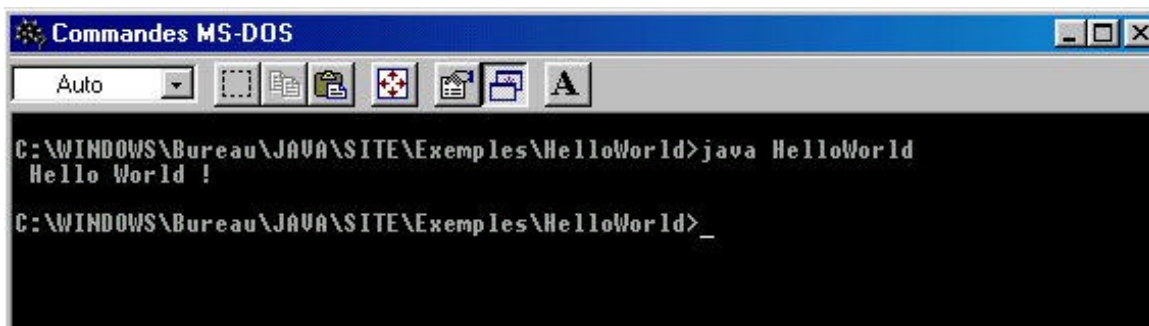
Application : le programme Hello World !

Hello World ! est célèbre : c'est traditionnellement le premier programme que l'on fait faire à une personne qui débute dans un langage de programmation. Il a pour but d'afficher le texte *Hello World* ! Il faut donc pour l'exécuter créer un programme, avec la méthode *main()*, et utiliser la méthode *System.out.println()* pour afficher notre texte. Au travail !

- `class HelloWorld { // début du programme`
- `public static void main (String args[]) { // début le ma méthode main()`
- `System.out.println(" Hello World ! "); // affichage du texte désiré`
- `} // fin de ma méthode main()`
- `} // fin du programme`

Note: pour compiler le programme, vous devez nommer votre fichier *HelloWorld.java*, en respectant la casse.

Voici le programme exécuté :



```
Commandes MS-DOS
Auto
C:\WINDOWS\Bureau\JAVA\SITE\Exemples\HelloWorld>java HelloWorld
Hello World !
C:\WINDOWS\Bureau\JAVA\SITE\Exemples\HelloWorld>_
```