

Nathan
Martins
BTS SIO1

Compte rendu : Java premier pas.

1.Introduction au langage Java :

Dans un premier temps avant d'utiliser java il faut comprendre l'importance d'un compilateur et d'un JVM.

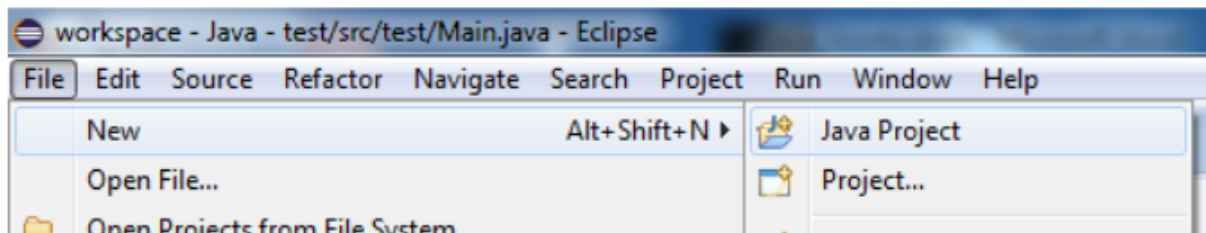
le compilateur de traduire le code source Java en un code binaire optimisé pour n'importe quel système Windows, Linux, Mac et une JVM, ou machine virtuelle Java permet de lire ce code binaire et l'exécuter sur le système de l'ordinateur.

2.Mise en place d'un environnement de développement :

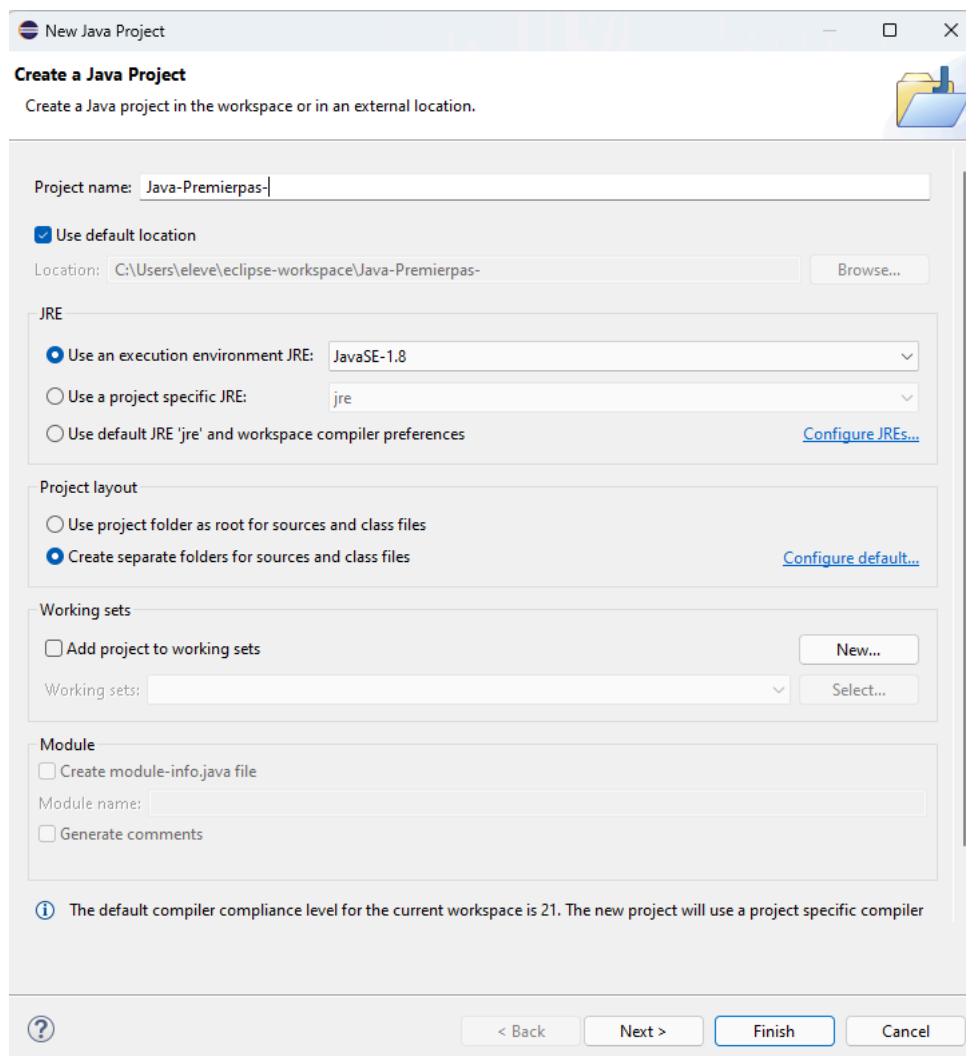
Les compilateurs et JVM utilisé pour ce premier pas sont AdoptOpenJDK et Eclipse.

3.Premier programme Java :

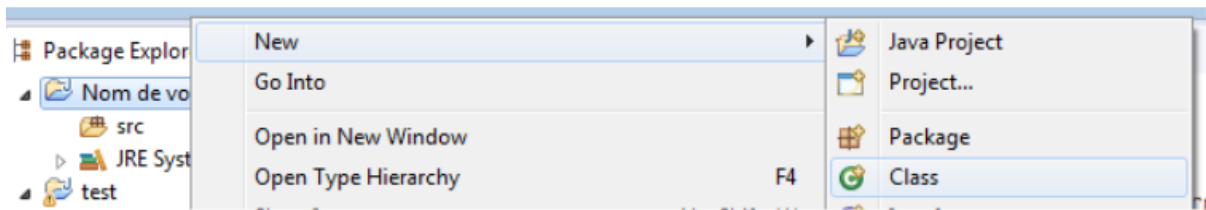
Afin d'avoir un endroit où coder, nous allons créer un nouveau projet.



il faut mettre un nom de projet dans "Project name:" et utiliser le jre "JavaSE-1.8" Ensuite finish permet de créer ce projet.



Créer une classe main (seule une classe Main peut être compilée)



En s'assurant de bien avoir indiqué un nom dans "Name:" et avoir sélectionné "Public static void main (String[]args)" et "Inherited abstract methods"

 A screenshot of the 'New Java Class' dialog box in Eclipse. The dialog has a title bar 'New Java Class' and a warning icon with the text 'The use of the default package is discouraged.' The fields are as follows:

- Source folder: Java-premierpas/src (with a 'Browse...' button)
- Package: (empty) (default) (with a 'Browse...' button)
- Enclosing type: (empty) (with a 'Browse...' button)
- Name: java1pas
- Modifiers: ☒ public, ☐ package, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass: java.lang.Object (with a 'Browse...' button)
- Interfaces: (empty list) (with 'Add...' and 'Remove' buttons)
- Which method stubs would you like to create?
 - ☒ public static void main(String[] args)
 - ☐ Constructors from superclass
 - ☒ Inherited abstract methods
- Do you want to add comments? (Configure templates and default value [here](#))
 - ☐ Generate comments

 At the bottom, there is a question mark icon, a 'Finish' button, and a 'Cancel' button.

Arrivé sur l'endroit où le code va être écrit on peut commencer à coder.

Il est important de toujours mettre un ";" à chaque fin de variables.

```
1
2 public class java1pas {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7     }
8
9 }
10
```

Notre premier programme va être de cette façon :

1. Importer la bibliothèque java.util.Scanner;

2. Créer une variable de type Scanner

3. Appeler la méthode de cette variable :

Pour un String = nextLine();

Pour un entier = nextInt();

Pour un float = nextFloat();

```
1 import java.util.Scanner;
2 public class java1pas {
3
4     public static void main(String[] args) {
5         try (Scanner sc = new Scanner (System.in)) {
6             System.out.println("veuillez saisir un mot : ");
7             String str = sc.nextLine();
8             System.out.println("vous avez saisi :" +str);
9         }
10    }
11 }
```

Ligne 1: permet d'importer la bibliothèque `java.util.Scanner`

Ligne 2: déclare une classe publique nommée `java1pas`. En Java, chaque programme doit être contenu dans une classe, et c'est ici que tu écris ton programme.

Ligne 4: Cette ligne définit la méthode `main`, qui est le point d'entrée de tout programme Java. C'est la première méthode exécutée lorsque le programme démarre. Elle prend en paramètre un tableau de chaînes de caractères `args`, qui peut contenir les arguments passés lors de l'exécution du programme.

Ligne 5 : Cette ligne crée un objet `Scanner` nommé `sc` qui va permettre de lire des entrées depuis le clavier de l'utilisateur (via `System.in`). Le bloc `try` est utilisé ici avec la déclaration de la ressource, ce qui permet de fermer automatiquement l'objet `Scanner` à la fin du bloc `try` (grâce à l'auto-closeable).

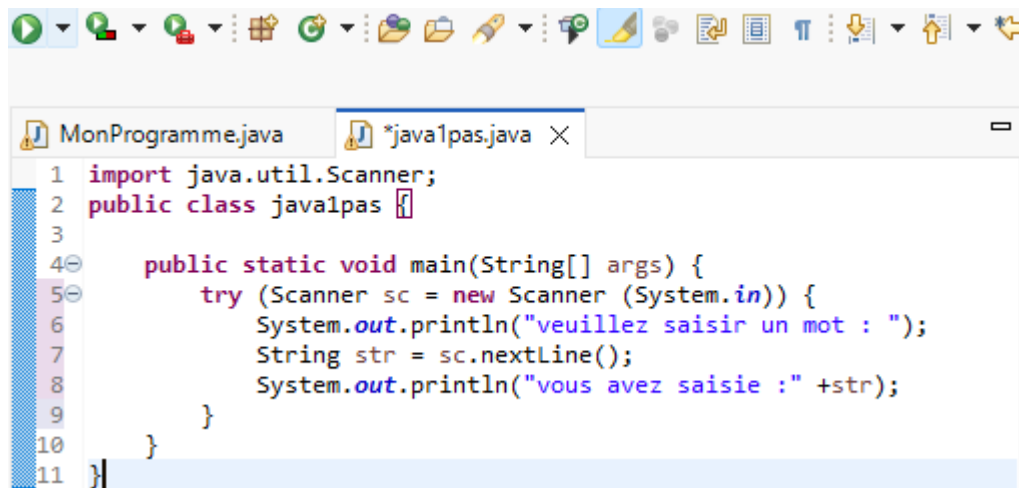
Ligne 6: Cette ligne affiche le message `"Veuillez saisir un mot : "` à l'utilisateur.

Ligne 7 : Cette ligne attend que l'utilisateur saisisse une ligne de texte et appuie sur "Entrée". Le texte saisi est ensuite stocké dans la variable `str` de type `String`.


Ligne 8: Cette ligne affiche le texte que l'utilisateur a saisi, en utilisant la variable `str`. Le `+` permet de concaténer le texte `"vous avez saisi : "` avec la valeur contenue dans la variable `str` de la ligne 7.

Les `}` permettent de fermer les variables `try`, `main`, et la classe `java1pas`.

Exécution du programme :



```
1 import java.util.Scanner;
2 public class java1pas {
3
4     public static void main(String[] args) {
5         try (Scanner sc = new Scanner (System.in)) {
6             System.out.println("veuillez saisir un mot : ");
7             String str = sc.nextLine();
8             System.out.println("vous avez saisie :" +str);
9         }
10    }
11 }
```

En appuyant sur run  le programme se lance dans la console.

Affiche "veuillez saisir un mot :

L'utilisateur saisie un mot

et le programme répond vous avez saisie : (le mots choisi par l'utilisateur)

