

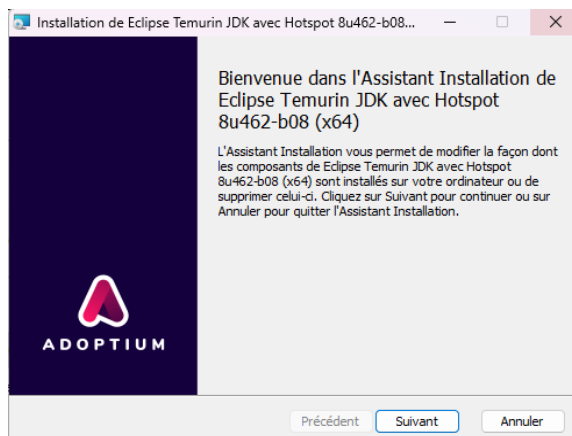
# B1 TP2: Mes premiers pas en Java

1 - Installation	1
2 - Création du Projet	1
3 - Création du premier programme	3

## 1 - Installation

Pour commencer, nous avons dû installer un JDK, qui nous permet de compiler et d'exécuter du code.

Nous avons choisi le JDK AdoptOpenJDK car il est gratuit, libre de droit et vise à fournir les logiciels d'exécutions Java.



Après avoir installé le JDK, nous devons installer Eclipse sur l'ordinateur. C'était déjà le cas sur mon ordinateur, donc je n'avais plus qu'à lancer Eclipse.

## 2 - Création du Projet

Après avoir choisi un directory pour mes projets, j'ai créé un nouveau projet Java en faisant File > New > Java Project, projet que j'ai nommé projet\_tp\_2.

**New Java Project**

**Create a Java Project**  
Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location  
Location:  [Browse...](#)

**JRE**  
☒ Use an execution environment JRE:   
☐ Use a project specific JRE:   
☐ Use default JRE 'jre' and workspace compiler preferences [Configure JREs...](#)

**Project layout**  
☐ Use project folder as root for sources and class files  
☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**  
☐ Add project to working sets [New...](#)  
 Working sets:  [Select...](#)

**Module**  
☒ Create module-info.java file  
 Module name:   
☒ Generate comments  
 module name will be "projet\_tp\_2" (if no module is specified, then project name will be used as module name)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

Après ça, j'ai créé une classe Main en faisant File>New>Class.

**New Java Class**

**Java Class**  
Create a new Java class.

Source folder:  [Browse...](#)

Package:  [Browse...](#)

☐ Enclosing type:  [Browse...](#)

Name:

Modifiers:  
☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static  
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:  [Browse...](#)

Interfaces:  [Add...](#) [Remove](#)

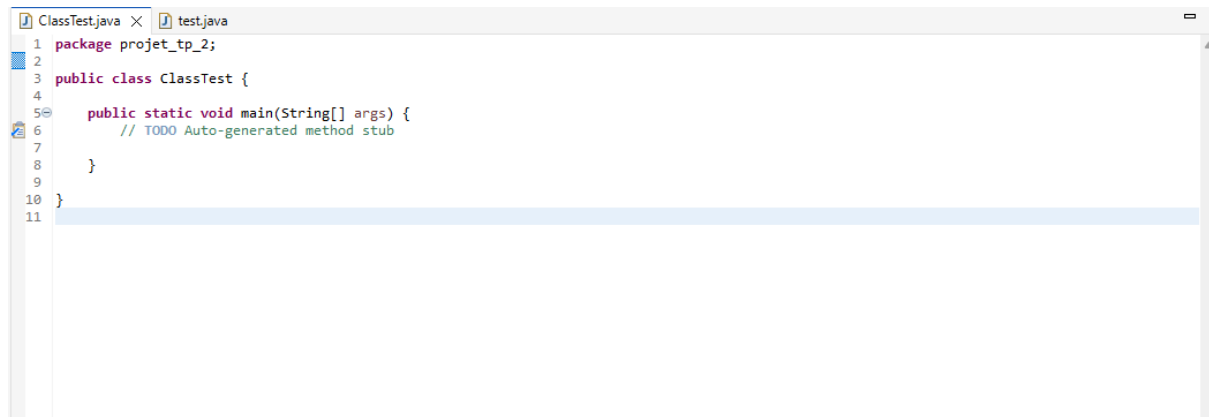
Which method stubs would you like to create?  
☐ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

[?](#) [Finish](#) [Cancel](#)

J'ai aussi coché "public static void main" avant de créer la class.

Et voilà:

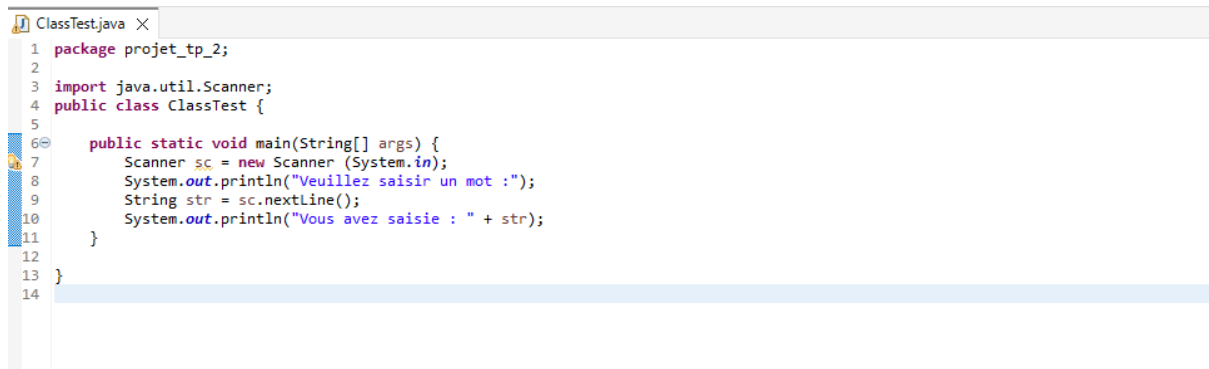


```
1 package projet_tp_2;
2
3 public class ClassTest {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7     }
8 }
9
10
11
```

### 3 - Création du premier programme

Ensuite, je veux faire un programme dans lequel je donne un mot, et le programme me dit quel mot j'ai saisi.

En m'aidant du modèle du PDF, j'écris le programme suivant:



```
1 package projet_tp_2;
2
3 import java.util.Scanner;
4 public class ClassTest {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner (System.in);
8         System.out.println("Veuillez saisir un mot :");
9         String str = sc.nextLine();
10        System.out.println("Vous avez saisi : " + str);
11    }
12
13 }
14
```

Je teste ensuite en lançant le programme, et on peut constater qu'il marche comme voulu.

```
/euillez saisir un mot :
Samir
/ous avez saisi : Samir
```

J'ai apprécié la partie installation du TP mais j'ai trouvé dommage que la partie code ne soit que du recopiage sans trop d'explications, par exemple je ne sais pas ce que certains termes comme "sc" ou "[System.in](#)" veulent dire que j'ai juste copié du PDF d'explications.