

**DEV1 – JAVL – Laboratoires Java****TD 1 – IntelliJ**

Dans ce TD vous ferez connaissance avec l'environnement intégré IntelliJ et vous réaliserez vos premiers programmes en Java.

Les codes sources et les solutions de ce TD se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/supports/dev1/tree/master/labo-java/labo-java-01-introduction/>

**Table des matières**

<b>1</b>	<b>IntelliJ : l'Environnement de Développement Intégré (IDE)</b>	<b>2</b>
<b>2</b>	<b>Affichage</b>	<b>5</b>
<b>3</b>	<b>Astuce pour coder rapidement</b>	<b>6</b>
<b>4</b>	<b>Expressions</b>	<b>7</b>

# 1 IntelliJ : l'Environnement de Développement Intégré (IDE)

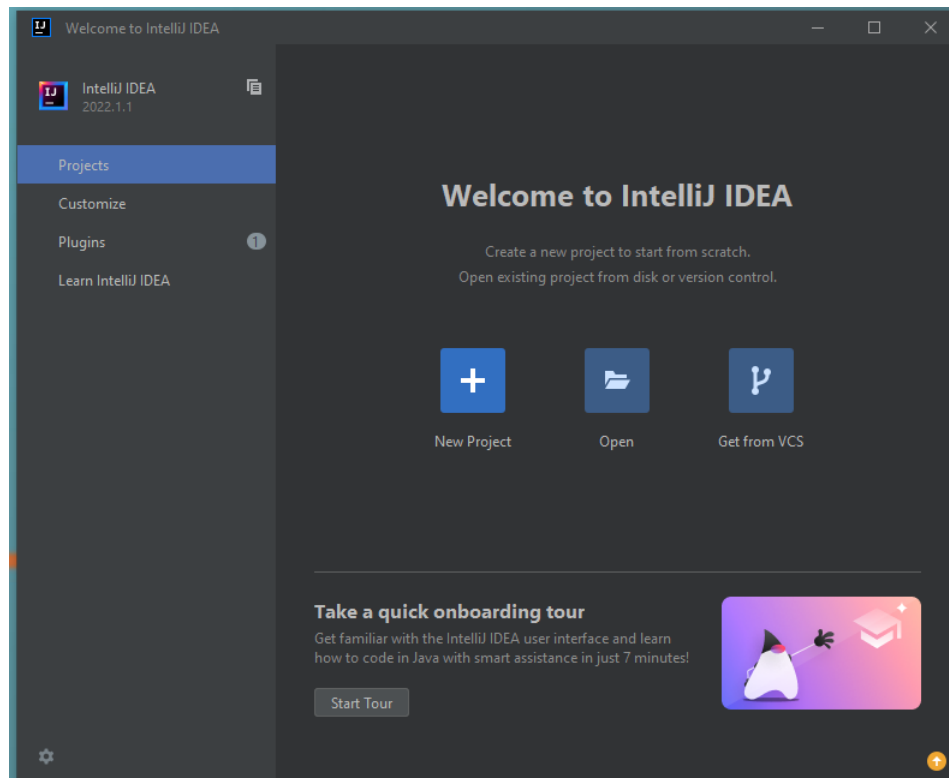
## Tutoriel 1 Premier programme Java avec IntelliJ

Vous allez être guidé pas à pas pour la création de votre tout premier programme Java.

- ✍ Ouvrez IntelliJ : l'icône de l'application se trouve sur votre Bureau.

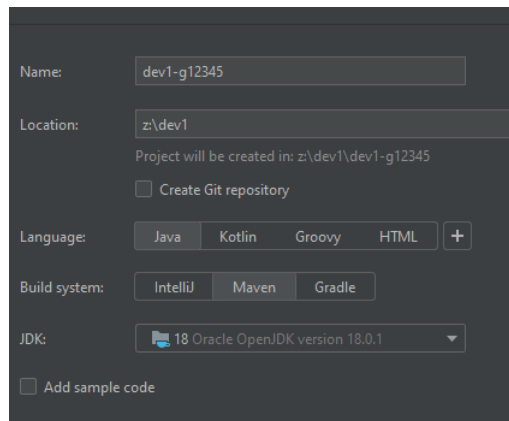


- ✍ L'IDE vous souhaite la bienvenue et vous propose plusieurs options pour continuer. Cliquez ici sur 'New Project' :

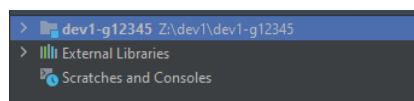


Dans IntelliJ, tout programme doit se trouver au sein d'un projet. Le projet contient le code de votre programme mais également les informations annexes comme le langage utilisé (ici Java), la version du langage (ici nous utilisons Java 18), et d'autres informations que vous découvrirez au fur et à mesure.

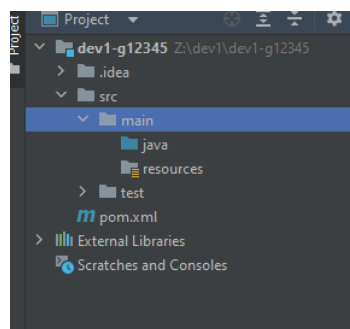
- ✍ L'environnement vous demande alors de choisir un nom pour votre projet, un emplacement sur le disque où stocker les différents fichiers de celui-ci ainsi que d'autres options techniques. Nous vous recommandons de stocker vos fichiers sur le disque partagé (Z) et de prendre pour nom de projet **dev1-g12345** en remplaçant 12345 par votre matricule étudiant. Complétez les informations restantes comme illustré ci-dessous : nous utiliserons le langage **Java**, avec le moteur de production **Maven** et la **version 18** du JDK (Java Development Kit).



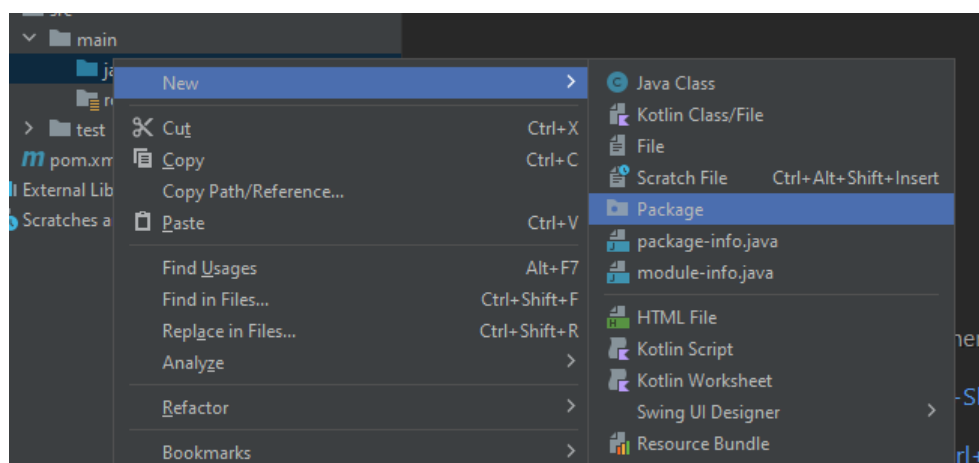
- ✍ Après avoir cliqué sur 'Create', le projet est créé et vous pouvez consulter les différents fichiers qu'il contient dans la barre latérale gauche :



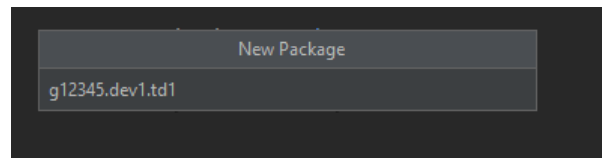
- ✍ Vous constatez en cliquant sur le nom du dossier de votre projet que IntelliJ a créé plusieurs dossiers, sous-dossiers et fichiers pour votre projet. Pour le moment, nous n'utiliserons que le dossier `Z:\dev1\dev1-g12345\src\main\java`, c'est celui qui contiendra le code Java de votre projet.



- ✍ Créez un *package* : faites un clic droit sur le dossier mentionné ci-dessus, comme illustré dans l'image ci-dessous et ajoutez un nouveau package Java.

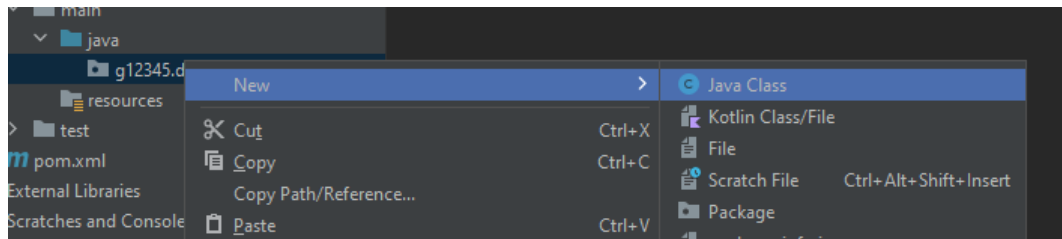


Nommez ce package `g12345.dev1.td1` où vous remplacez `g12345` par votre identifiant :



En Java, un package (paquet) permet de regrouper certaines parties de votre code (une sorte de classeur, qui contiendra des fichiers de code) et ainsi d'ordonner votre projet.

- ✍ Le package ainsi créé apparaît dans la fenêtre de gauche. Cliquez droit sur celui-ci pour ajouter une *classe* à votre package :



Nommez la **Hello**.

- ✍ Ouvrez le fichier **Hello.java** : double-cliquez sur votre classe **Hello**. Le code se trouvant dans ce fichier apparaît.

Ajoutez le code suivant en respectant bien les minuscules et les majuscules :

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Vous devriez obtenir ceci :



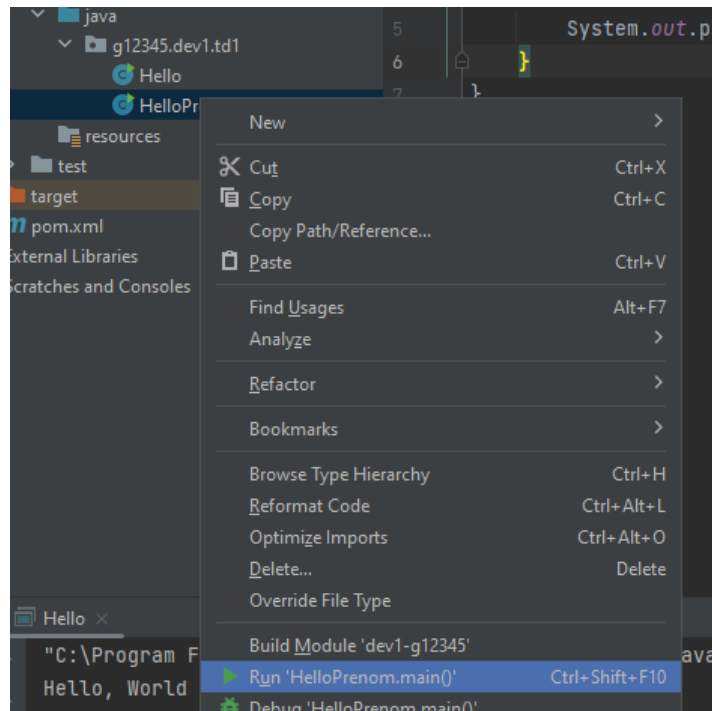
Enregistrez vos modifications (control-s).

- ✍ Lancez le programme en cliquant sur la petite flèche *run*



Le résultat s'affiche dans la fenêtre intitulée « Run ». Vous devriez voir "Hello, World!" s'y afficher.

- ✍ Créez une nouvelle classe qui se nomme **"HelloPrenom"**. Celui-ci doit afficher "Hello" suivi de votre prénom. Si vous essayez de le faire tourner en cliquant sur la flèche, vous ne verrez pas apparaître votre prénom. C'est parce que IntelliJ a retenu que le programme principal est le premier à avoir été lancé. Pour forcer IntelliJ à lancer le nouveau programme, faites un clic droit sur le fichier **HelloPrenom.java** et cliquez sur **Run HelloPrenom.main()**. Vous devriez voir "Hello, <votre prénom>" s'afficher dans l'output.



### Structure d'un programme Java sous IntelliJ

Nous venons de voir quelques points fondamentaux du développement Java que vous serez amené à utiliser tout au long de l'UE DEV1 pour créer des programmes Java. Voici un résumé de ces points :

1. Les différents fichiers et dossiers constituant une application Java sont regroupés au sein d'un projet,
2. Le code d'un programme est situé au sein d'une classe, contenue dans un fichier `.java` dont le nom est le même que celui de la classe,
3. Chaque classe est rangée dans un package (une sorte de classeur pour les fichiers contenant le code), ce package correspond à un sous-dossier du dossier `src/main/java` de votre projet.

## 2 Affichage

Le programme suivant affiche **Hello!** suivi de **Bonjour!**.

```

1 package g12345.dev1.td1;
2
3 public class HelloBonjour {
4
5     public static void main(String[] args) {
6         System.out.println("Hello!");
7         System.out.println("Bonjour!");
8     }
9 }
```

- ▷ La première ligne indique que la classe se trouve dans le package `g12345.dev1.td1`
- ▷ La ligne 3 déclare la classe `HelloBonjour`, remarquez que le nom d'une classe doit correspondre au nom du fichier dans lequel elle se trouve, ici `HelloBonjour.java`

- ▷ La ligne 5 déclare la *méthode principale*, *main* en anglais veut dire 'principale'. C'est ici que commence votre programme.
- ▷ En java l'affichage se fait par l'*instruction* : `System.out.println("Hello!");`
- ▷ Le texte entre guillemets sera affiché sur la *sortie standard*, dans IntelliJ la sortie standard est la fenêtre "Run".

Le code source de la classe `HelloBonjour` (ainsi que tous les codes de ce TD) se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/supports/dev1/tree/master/lab0-java/lab0-java-01-introduction-code/>

Prenez l'habitude de copier dans votre projet IntelliJ les codes présentés dans les TDs afin de les exécuter sur votre machine.

### Exercice 1 Ligne

Dans votre package `g12345.dev1.td1` créez une classe `Exercice1`. Dans cette classe écrivez un programme (et donc dans la méthode `main` de cette classe) qui affiche 10 tirets les uns à la suite des autres <sup>1</sup> :

```
- - - - -
```

### Exercice 2 Carré

Dans une classe `Exercice2` (dans votre package `g12345.dev1.td1`), écrivez un programme qui affiche un carré d'étoiles de 5 de côté :

```
*****
*****
*****
*****
*****
```

### Exercice 3 Pyramide

Dans une classe `Exercice3`, écrivez un programme qui affiche une pyramide d'étoiles comme ceci :

```
  *
 ***
*****
*****
```

## 3 Astuce pour coder rapidement

Devoir sans arrêt taper les lignes de code

```
public static void main(String[] args)
```

et

```
System.out.println(...)
```

devient vite lassant et chronophage. Pour vous aider, IntelliJ met à votre disposition des raccourcis clavier vous permettant d'aller plus vite. Pour la première ligne de code, tapez simplement `psvm` suivi d'une tabulation (la touche 'tab' de votre clavier) :

---

1. Basez-vous sur les précédents programme.

```

1 package TD1;
2
3 public class Exercice1 {
4     psvm|
5 }

```

IntelliJ ajoute alors automatiquement la ligne de code demandée. De même, pour la deuxième ligne de code, tapez simplement `sout` suivi d'une tabulation :

```

2
3 public class Exercice1 {
4     public static void main(String[] args) {
5         sout|
6     }
7 }

```

IntelliJ propose d'autres raccourci claviers du même genre pour aider les développeurs à coder plus rapidement, n'hésitez pas à consulter la documentation de l'IDE (en ligne) pour en savoir plus.

## 4 Expressions

Le programme suivant affiche la somme des *nombres entiers* 12345678 et 87654321.

```

1 package g12345.dev1.td1;
2
3 public class Expression {
4
5     public static void main(String[] args) {
6         System.out.println("12345678+87654321 = ");
7         System.out.println(12345678+87654321);
8     }
9 }

```

- ▷ L'instruction de la ligne 6 affiche le texte entre guillemets : 12345678+87654321 =
- ▷ L'instruction de la ligne 7 affiche le *résultat du calcul* c'est-à-dire la somme de 12345678 et de 87654321 : 99999999
- ▷ Notez que l'*expresssion* 12345678 + 87654321 de l'instruction de la ligne 7 ne se trouve *pas* entre guillemets.

Le nombre 12345678 est un nombre entier. Les nombres décimaux, aussi appelés flottants ou nombres à virgule, s'écrivent avec un point, par exemple : 12.3

### Exercice 4 Petits calculs

Dans une classe `Exercice4`, écrivez un programme qui affiche les expressions suivantes (sans effectuer le calcul) puis, à la ligne suivante, la valeur de l'expression, pour les expressions suivantes :

- ▷  $10 + 32$
- ▷  $10 - 32$
- ▷  $2 * 21$
- ▷  $234 \% 57$
- ▷  $((2*2)+(3*3)) / 25$
- ▷  $12.3 + 13.5$
- ▷  $12.3 - 13.5$
- ▷  $12.3 * 13.5$
- ▷  $2.0 / 3.0$

Remarque :  $\%$  est l'opérateur modulo c'est-à-dire le reste de la division entière.  $/$  est l'opérateur de division ici division entière. Par exemple,  $37\%10$  vaut 7 et  $37/10$  vaut 3.

#### **Exercice 5** Divisions entière et décimale

Dans une classe Exercice5, créez un programme qui affiche, de même, les expressions à évaluer puis leur valeur pour les expressions suivantes :

- ▷  $2.0 / 3.0$
- ▷  $2 / 3$
- ▷  $2 / 3.0$
- ▷  $2.0 / 3$
- ▷  $2.0 / 0.0$
- ▷  $2 / 0$

Notez la différence de résultat entre les expressions  $2.0/3.0$  qui est une division entre nombres décimaux et  $2/3$  qui est une division entre entiers (et donc une division entière).

Notez également la différence de résultat entre les 2 dernières expressions qui sont des divisions par zéro. La première est une division entre nombres décimaux, la seconde est une division entre entiers.