

# Algorithmes et Pensée Computationnelle

## Introduction à Java

### 1 Objectifs du document

Ce document constitue un guide pour débiter à programmer en utilisant le langage Java. Les objectifs de ce guide sont les suivants :

- Comprendre comment créer, éditer et lancer un programme en Java.
- Découvrir l'environnement de travail qui sera utilisé
- Se familiariser avec quelques notions de base du langage.

Le code utilisé dans ce document se trouve sur Moodle, dans le dossier *Exercices de Base (version hors ligne)/Code*.

### 2 Introduction


#### 2.1 Création d'un programme Java via le Terminal

Pour commencer à programmer en Java, il est nécessaire de créer un fichier ayant une extension `.java`. Pour ce faire, il suffit de créer un simple fichier java vide. La manière la plus simple est de passer par le terminal (ou invite de commande). Une fois ouvert, entrer la commande `nano monpremierfichierjava.java` pour créer un fichier vide nommé `monpremierfichierjava.java`.



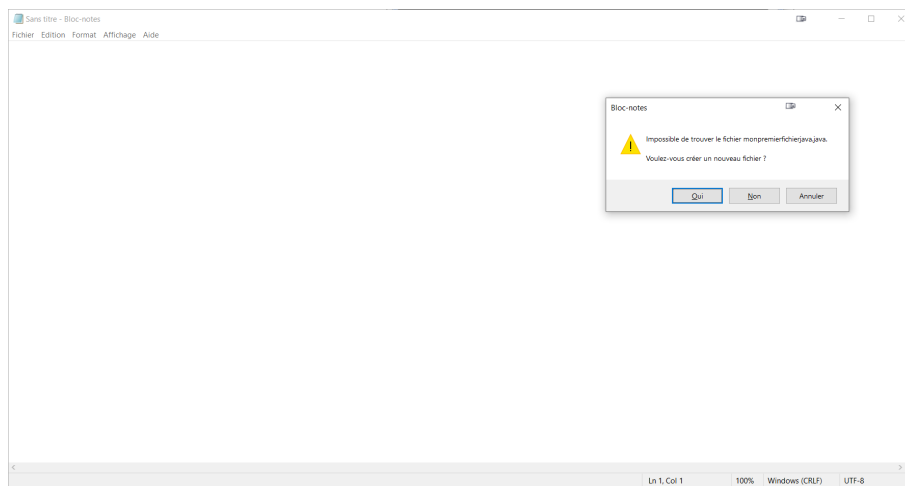
#### 2.2 Création d'un programme en Java sur Windows

Pour créer un fichier Python sur Windows, il suffit simplement d'entrer la commande `notepad monpremierfichier.java` dans le bloc-note. Vous pouvez alors simplement l'éditer et l'enregistrer.




```
Sélection Invite de commandes
Microsoft Windows [version 10.0.19042.1237]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Arnaud>notepad monpremierfichierjava.java
C:\Users\Arnaud>
```



Pour éditer votre programme Java, il suffit simplement de l'éditer avec n'importe quel éditeur de texte. Vous pouvez ensuite compiler et lancer votre programme Java en entrant les commandes **javac monpremierfichierjava.java** et **java monpremierfichierjava.java**. Ce dernier va s'exécuter dans le terminal.

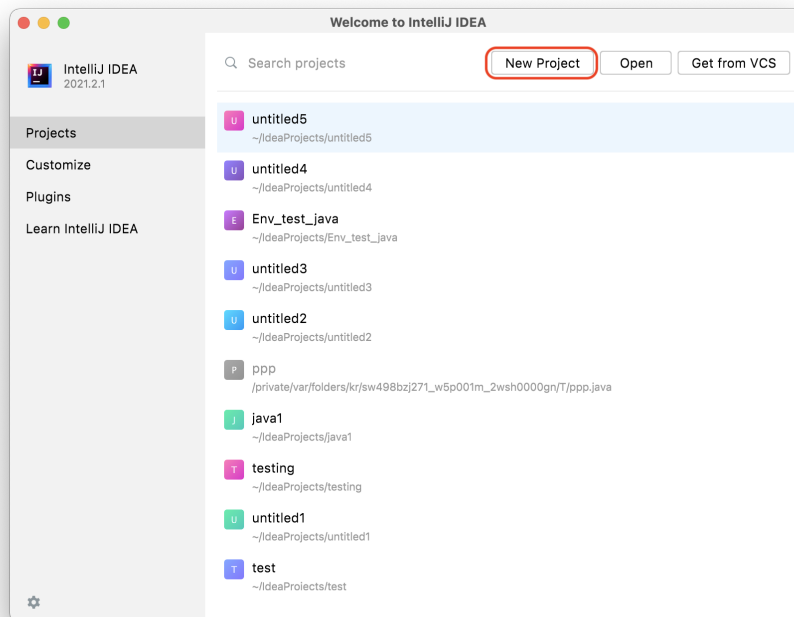


```
arnaud -- zsh -- 80x24
Last login: Fri Oct 1 13:27:23 on ttys000
arnaud@Arnauds-MBP ~ % javac monpremierfichierjava.java
```

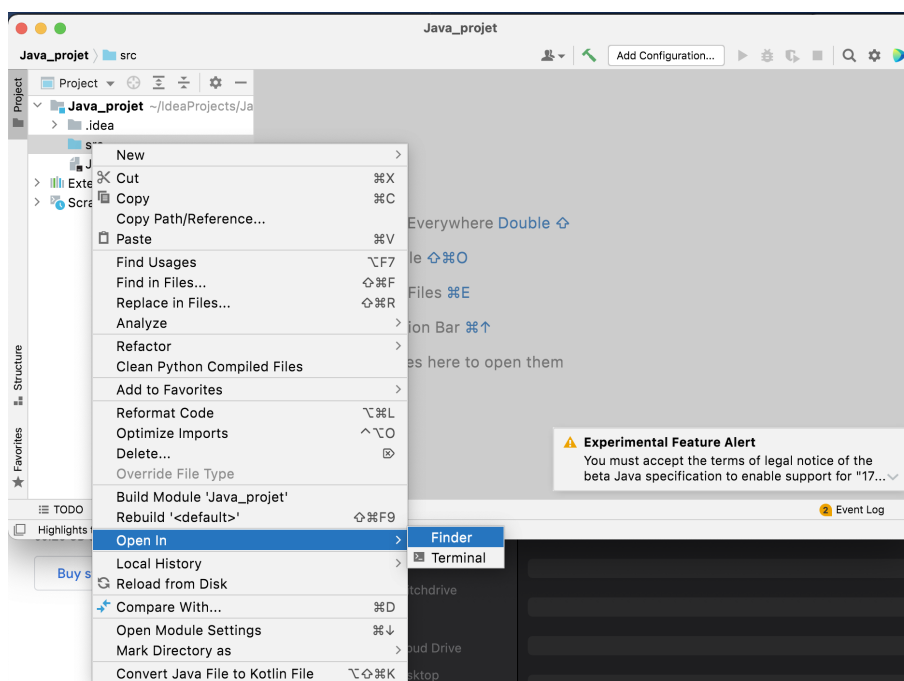
## 2.3 IntelliJ IDEA

Dans le cours, il vous a été demandé d'installer IntelliJ IDEA, l'un des environnements de développement le plus populaire de ces dernières années. Un environnement de développement ou IDE (Integrated Development Environment) est un programme qui combine différents outils de développement et qui facilite le travail d'un programmeur. Pour ouvrir un fichier `.java` avec le programme, il est nécessaire de créer un nouveau projet et de copier le fichier `.java` dans le dossier contenant le projet.

Commencez par ouvrir IDEA et créez un nouveau projet.



Dans le menu à gauche, vous allez retrouver un sous-dossier nommé `src`. Ouvrez l'emplacement de ce dossier puis copiez le fichier `.java` que vous souhaitez lancer.

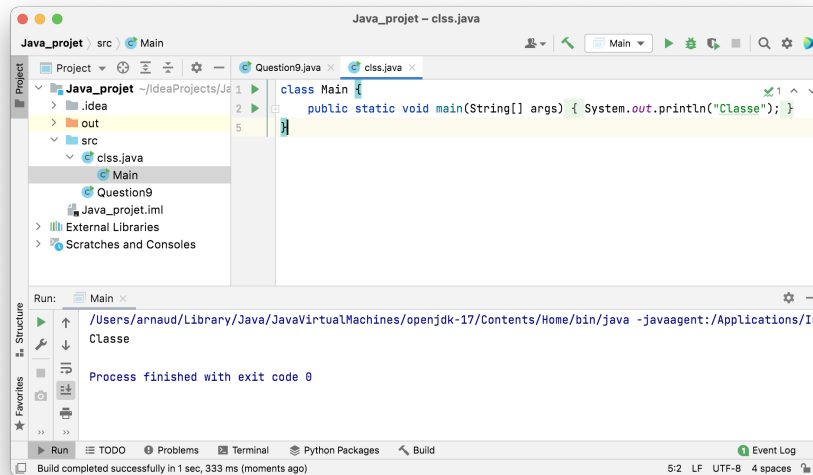


La fenêtre qui vient de s'ouvrir est similaire à un éditeur de texte basique. C'est ici que le code peut être entré et modifié.

Vous pouvez maintenant lancer le fichier en allant dans **Run... > Run** lenomdevotreclasse

## 2.4 Les Classes en Java

Contrairement à d'autres langages de programmation, tout code en Java doit être écrit à l'intérieur d'une classe. La notion de classe vous sera présentée plus en détails dans la suite du cours.



## 2.5 La méthode main

Dans tout programme Java, on doit toujours retrouver au moins une fois la fonction **public static void main(String[] args)** dans une de nos classes. Cette fonction est un point d'entrée dans le programme qui ne s'exécutera pas s'il ne retrouve pas cette fonction.

## 2.6 La création de variables

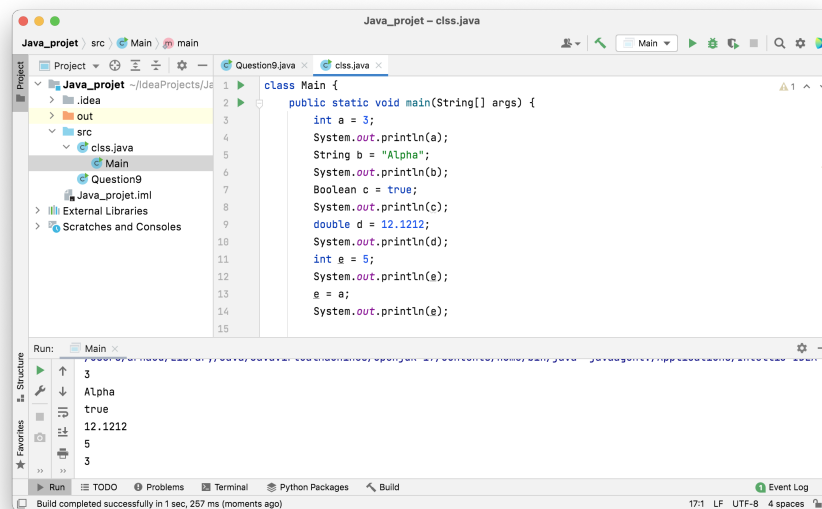
Les variables dans les langages de programmation sont similaires à des noms données à une valeur précise. Pour assigner une valeur à une variable en Java, il suffit de respecter le format suivant **type variable=valeur**.

### Conseil

- On peut assigner n'importe quelle suite de caractères non-réservee en tant que variable
- Il est aussi possible d'assigner des chaînes de caractères (strings en anglais) ou encore des valeurs booléennes à des variables
- En réassignant une nouvelle valeur à une variable déjà définie, la valeur de la variable va être écrasée et remplacée par la nouvelle valeur
- Il est possible d'additionner les variables du même type
- Plusieurs variables peuvent avoir la même valeur
- Le nom d'une variable doit toujours commencer par une lettre

À noter que les dernières versions de Java supportent l'inférence du type des variables. On pourrait autant écrire **String a="abc"** que **a="abc"** car le type de la variable serait automatiquement défini. Dans le cadre du cours, nous vous recommandons de toujours déclarer le type de vos variables.

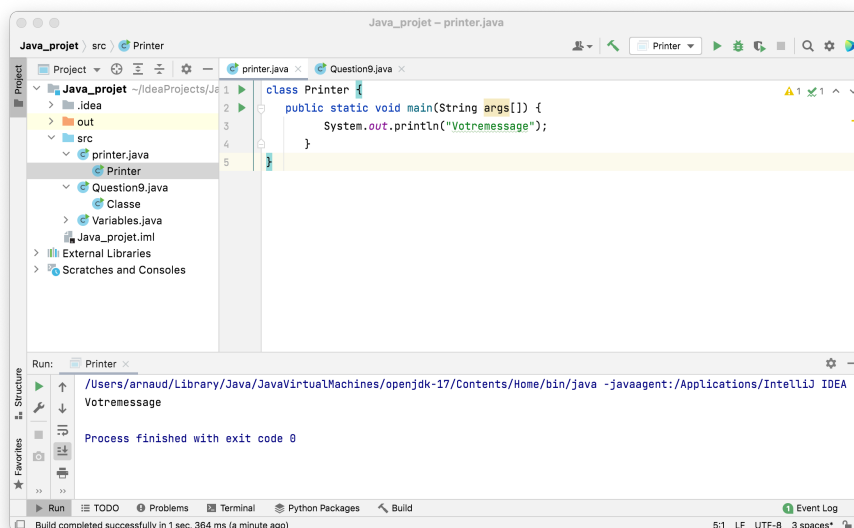
Si vous ouvrez et exécutez le programme Java **Variables.java** qui contient quelques exemples d'attribution de variables, vous pourrez observer comment sont créées différentes variables mais aussi comment le programme les traite.



## 2.7 Les fonctions

Dans les langages de programmation, on retrouve un très grand nombre de fonctions. Ces fonctions sont des blocs de code qui, lorsqu'ils sont invoqués avec certains paramètres, effectuent certaines actions. Une des fonctions basique et plutôt importante en Java est la fonction `System.out.println("Votremessage");`. Cette dernière permet d'afficher à l'écran le contenu de la parenthèse.

Si vous entrez la ligne de code `System.out.println("Votremessage");` et que vous lancez le programme, IntelliJ va vous afficher une ligne de texte. Le message qui apparaît est celui que vous avez entré entre guillemets (à la place de "Votremessage"). C'est le but d'une fonction `System.out.println("Votremessage");`;



En plus des fonctions incluses dans les bibliothèques de base, il est possible de créer des fonctions complètement personnalisées (généralement plusieurs fonctions sont réunies en une seule) au moyen de la fonction `static void nomdelafonction()` et de l'utiliser à n'importe quel moment en l'invoquant avec `nomdelafonction()`. Si vous ouvrez et exécutez le programme `mafonction.java`, vous pourrez voir l'exemple de la fonction personnalisée `mafonction` qui imprime 3 suites de caractères différentes.

