

Manuel IntelliJ

Ce petit guide a pour but de vous aider à prendre en main le logiciel IntelliJ. IntelliJ est un IDE (Integrated Development Environment, ou Environnement de Développement Intégré en français) pour Java, c'est-à-dire un ensemble d'outils réunis en une application pour aider à la programmation en langage Java. Comme cet outil est très complet, on peut donc vite s'y perdre.

Ce manuel vous guidera pour 1. Installer et configurer IntelliJ, 2. Créer des projets java et y ajouter les fichiers donnés en cours, et 3. Exécuter (lancer) votre code.

1. Installer et configurer IntelliJ

1.1 Télécharger et installer

1.1.1 Commencez par télécharger Java [sur cette page](#). Choisissez l'*Installer* qui correspond à votre système d'exploitation. Vous devrez créer un compte Oracle pour commencer le téléchargement. Laissez le logiciel d'installation vous guider et laissez tout par défaut. Souvenez-vous toutefois de l'endroit où Java sera installé sur votre ordinateur, cela pourra être utile pour la suite.

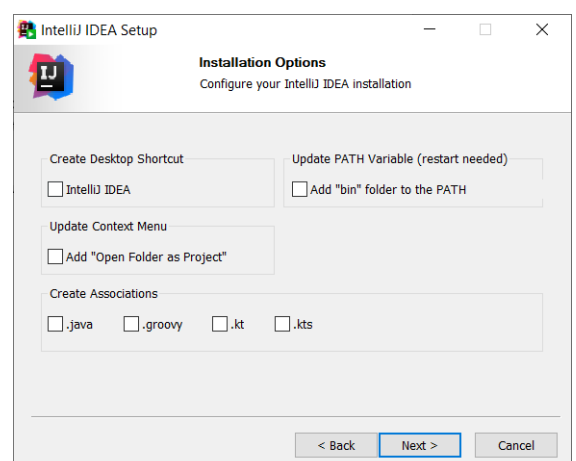
1.1.2 Créez un compte JetBrains [sur cette page](#) avec votre adresse mail **Vinci**. C'est important d'utiliser cette adresse en particulier car elle vous permettra d'avoir la licence Ultimate gratuitement.

1.1.3 Rendez-vous sur la [page de téléchargement de IntelliJ](#) et téléchargez la version **Ultimate** (celle de gauche), vous y avez droit grâce à votre compte Vinci étudiant. Si vous utilisez Ubuntu, vous pouvez également le trouver dans la Logithèque Ubuntu.

a) Windows

Ouvrez l'installateur que vous venez de télécharger. Cliquez sur « Next », choisissez un emplacement pour l'installation, et cliquez de nouveau sur « Next ». Le menu d'options d'installation apparaît.

- **Add « Open Folder as Project » recommandé :**
Permet d'ajouter une option au menu qui apparaît lorsque vous faites clic droit sur un dossier, permettant de l'ouvrir en tant que projet dans IntelliJ.
- **Create associations pas nécessaire :**
Les types de fichiers cochés s'ouvriront avec IntelliJ par défaut.
- **Update PATH Variable pas nécessaire :**
Permet de lancer IntelliJ via l'invite de commandes



Choisissez les options que vous préférez et cliquez sur « Next ». Vous pouvez choisir le dossier de menu de démarrage, puis cliquez sur « Install ».

b) MacOS

Ouvrez l'image disque et glissez IntelliJ dans votre dossier Applications.

c) Linux

Sur Ubuntu, vous pouvez trouver IntelliJ dans la logithèque Ubuntu. Dans ce cas, il suffit de l'installer par ce biais. Vous pourrez le lancer en utilisant l'image de bureau.

Sinon, ou si vous avez téléchargé l'installateur depuis le site d'IntelliJ, vous pouvez l'extraire de l'archive dans le dossier de votre choix grâce à la ligne de commande suivante :

```
$ sudo tar -xzf ideaIU.tar.gz -C /chemin/vers/le/dossier
```

Pour lancer IntelliJ, il vous suffit de vous rendre dans le dossier bin du dossier créé :

```
$ cd /chemin/vers/le/dossier/ideau-IU-<numero>/bin
```

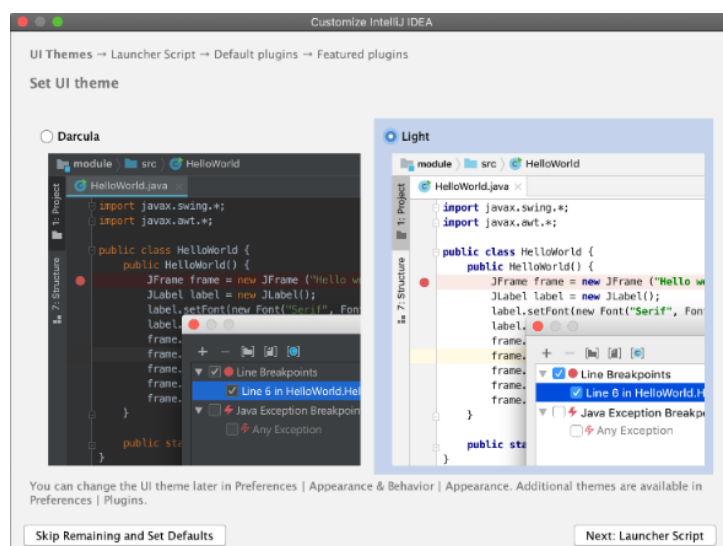
Puis d'exécuter le script idea.sh :

```
$ ./idea.sh
```

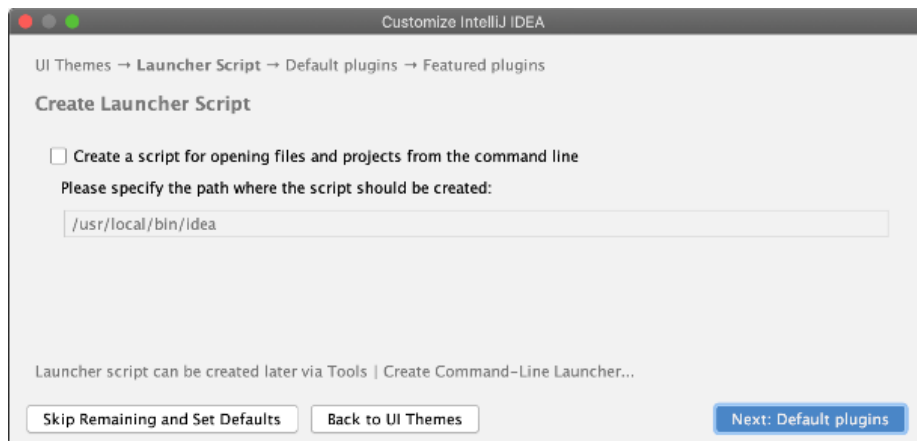
1.2 Premier lancement

Il est fort probable que toutes les étapes ne s'affichent pas automatiquement. Pas d'inquiétude, tout est optionnel.

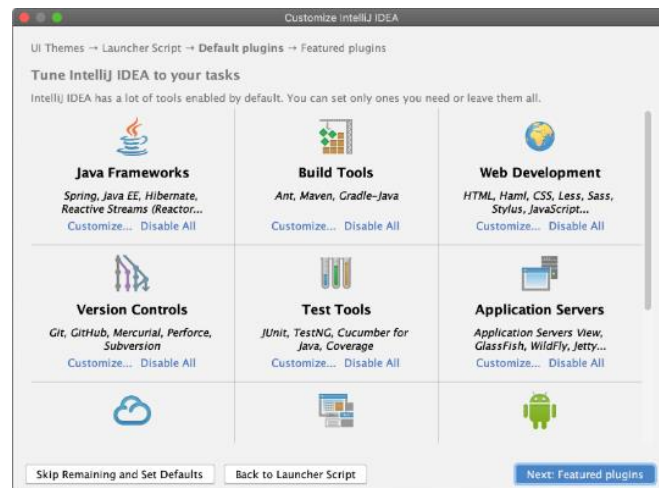
1.2.1 Lors du premier lancement, vous serez amené à choisir des options pour IntelliJ. Tout d'abord, vous devez choisir le thème (sombre ou clair). Choisissez selon votre préférence et cliquez sur « Next ». Le thème ne change rien à part l'apparence de l'éditeur. Si cette étape n'est pas affichée automatiquement, vous pourrez changer la couleur de la fenêtre au point **1.2.5** en cliquant sur « Customize ». L'option « Color theme » propose un menu déroulant avec les différents jeux de couleurs.



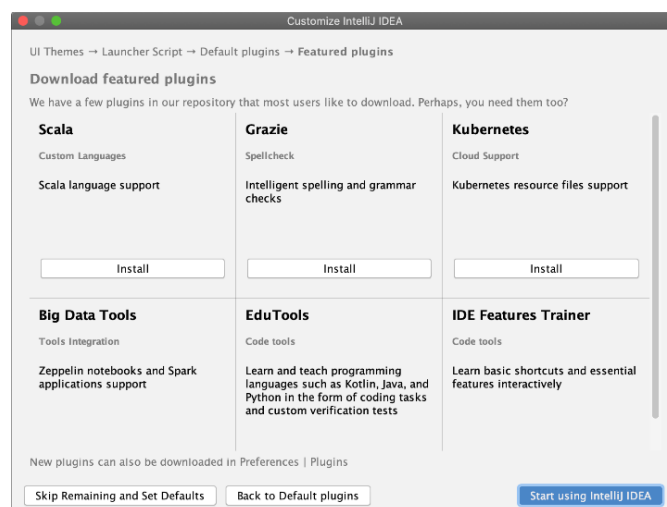
1.2.2 Si vous êtes sur Linux ou MacOS, IntelliJ vous propose ensuite de créer un script de lancement pour ouvrir les fichiers et les projets en lignes de commande. C'est utile **uniquement** si vous souhaitez effectivement utiliser le terminal. Dans ce cas, cochez la case et écrivez le chemin d'accès au script (vous pouvez choisir l'emplacement, même si le dossier n'existe pas encore : IntelliJ le créera automatiquement). **Si c'est flou pour vous, ne cochez pas la case !**



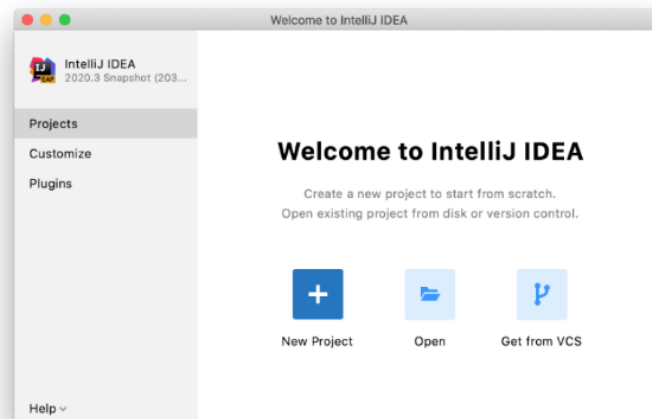
1.2.3 IntelliJ comprend de nombreux plugins par défaut. Vous n’aurez pas besoin de toutes les fonctionnalités proposées. La page ci-dessous peut s’afficher pour vous proposer de désactiver des plugins afin d’alléger l’application. Ne désactivez rien pour l’instant : si vous rencontrez des problèmes de performance pendant les exercices, vous pouvez envoyer un mail à baptiste.lapiere@vinci.be pour décider des plugins à désactiver.



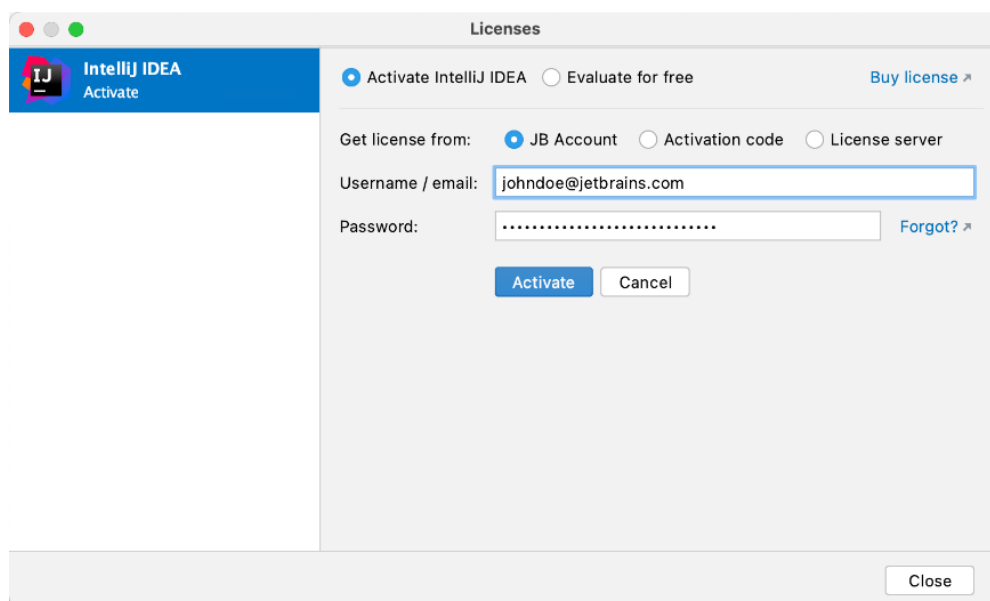
1.2.4 IntelliJ propose ensuite de télécharger et installer des plugins supplémentaires. Vous n’en avez pas besoin. Cliquez simplement sur « Start using IntelliJ IDEA » si cette page apparaît.



1.2.5 L'installation est complète, vous arrivez sur le menu d'accueil d'IntelliJ (attention, l'étape 1.2.6 peut se produire avant automatiquement) :



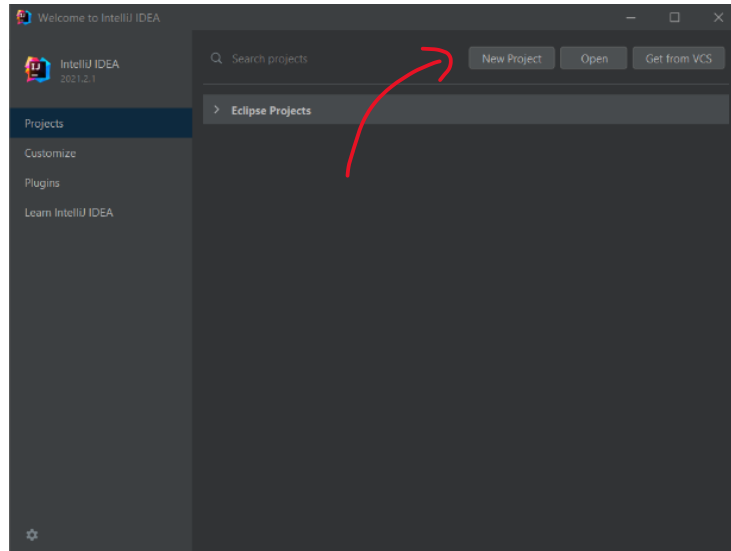
1.2.6 Cliquez sur « Help >> Register » pour vous connecter à JetBrains via IntelliJ. Cochez la case « Activate IntelliJ IDEA ». À côté de « Get license from », cochez « JB account ». Vous pouvez encoder votre adresse mail Vinci et votre mot de passe JetBrains et cliquer sur « Activate ». Notez que vous serez peut-être automatiquement dirigé vers une page de connexion avant l'étape 1.2.5.



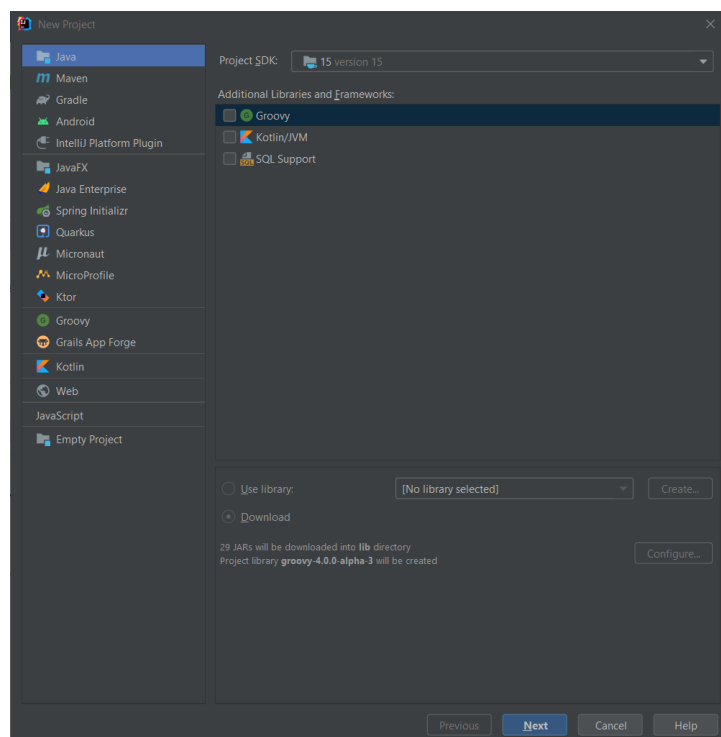
2. Créer un projet et y ajouter des fichiers

2.1 Création de projet

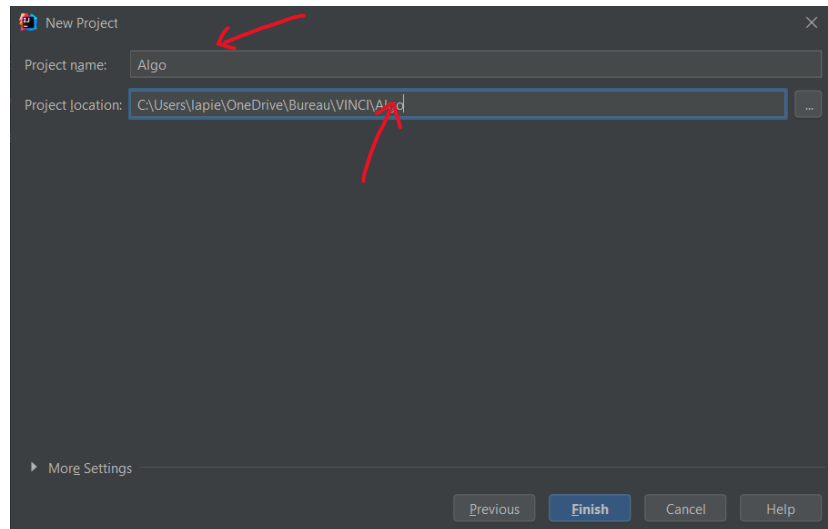
2.1.1 Commencez par cliquer sur « New Project » sur la page d'accueil :



2.1.2 Dans la fenêtre qui s'ouvre, Java est normalement sélectionné dans le menu de gauche (si ce n'est pas le cas, sélectionnez Java en haut à gauche). Choisissez une version de Java (vous pouvez laisser le choix par défaut) et ne cochez aucune case, nous n'aurons pas besoin de ces options. Cliquez sur « Next », puis encore sur « Next » dans la fenêtre suivante. **Si aucune version de java n'est proposée par défaut**, vous pouvez cliquer sur « Add JDK » dans le menu déroulant et sélectionnez le dossier où Java a été installé au point 1.1.1.



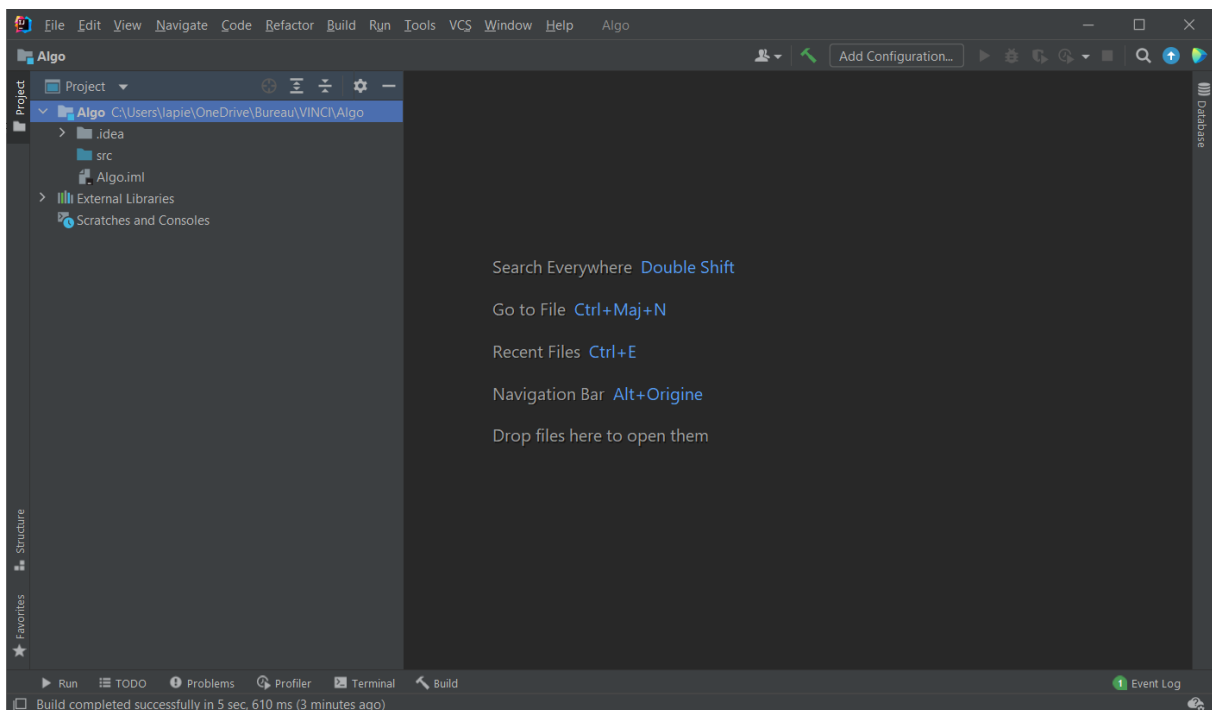
2.1.3 Vous pouvez ensuite choisir un nom pour votre projet (par exemple « APOO » ou « Algo » 😊) et l'endroit où il sera sauvegardé. Nous recommandons de choisir un emplacement que vous pourrez retrouver facilement, typiquement dans un dossier que vous prévoyez pour les projets du cours. Pour le nouveau projet « Algo », vous choisirez par exemple « .../IPL/Bloc 1/Algo ». Attention à bien faire correspondre le nom du projet et le nom de dossier dans *Project location* :



Vous pouvez cliquer sur les « ... » pour sélectionner le dossier dans votre explorateur de fichier. Si vous choisissez cette méthode, il faudra créer le dossier **contenant** le projet (dans l'exemple : VINCI, pour le projet Algo) manuellement.

Cliquez sur « Finish ».

2.1.4 Vous arriverez à la fenêtre de votre nouveau projet, actuellement vide.

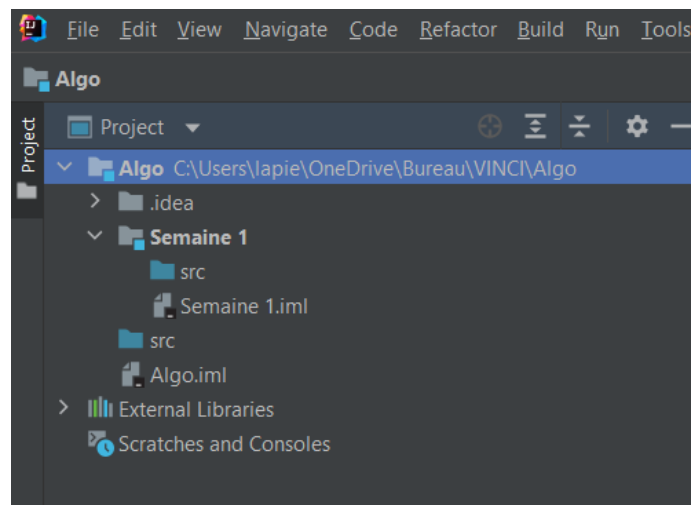


Dans l'espace de gauche se trouve la *hiérarchie* du projet. Vous y retrouvez le dossier contenant le projet, un dossier « .idea » qui contient des métadonnées nécessaire à IntelliJ auxquelles vous ne

devez pas toucher, un fichier « *nomDuProjet.iml* » qui contient des métadonnées du projet (vous n'avez pas besoin d'y toucher non plus) et un dossier vide : « *src* ».

Il se peut qu'un temps de chargement assez long s'écoule avant que vous ne puissiez faire quoi que ce soit. C'est normal, pas d'inquiétude.

2.1.5 Afin d'organiser vos exercices, vous pouvez créer un *module* pour chaque exercice. Pour cela, il vous suffit de faire un clic droit sur le nom de votre projet dans la hiérarchie, et cliquer sur « New... >> Module » dans le menu qui apparaît. Une fenêtre de création de projet apparaît. Comme pour la création initiale du projet, cliquez sur « Next » sans rien cocher. Choisissez un nom de module (exemple « Semaine 1 ») et laissez la destination intacte. Elle sera par défaut à l'intérieur du dossier de votre projet. Cliquez sur « Finish ». Le module apparaît maintenant dans la hiérarchie.



Organisation selon les cours : Attention, l'organisation des modules sera différente selon les cours. En **algorithmique**, on vous demande de créer **un module par semaine** (il se peut qu'occasionnellement vous deviez en créer deux. Si c'est le cas, vous serez prévenu en séance). En **Analyse et Programmation Orientée Objet (APOO)**, on vous demande de créer **un module par exercice**.

2.2 Ajouter les fichiers des exercices

Trois cas de figure peuvent se présenter lors d'un exercice :

- a) Vous recevez uniquement des fichiers « .java »
- b) Vous recevez une archive contenant un projet java
- c) Vous ne recevez aucun fichier « .java » et devez les créer vous-même

La procédure dans chacun des cas est décrite ci-dessous.

a) Vous recevez des fichiers java

Dans ce cas, vous devrez commencer par créer un projet avec IntelliJ comme décrit dans le point 2.1.

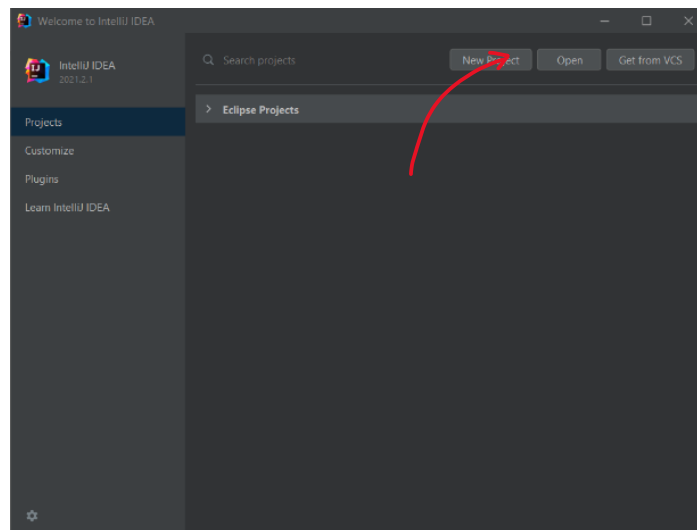
Il vous suffit de glisser-déposer les fichiers java dans le dossier *src* de la hiérarchie de votre projet. IntelliJ vous demandera de valider le *move refactor* (voir Notes, plus bas). Cliquez simplement sur OK.

Et c'est tout, vous pouvez commencer l'exercice. **Attention** : si vous procédez de cette manière, les fichiers seront déplacés, vous ne conserverez donc pas les fichiers d'origine.

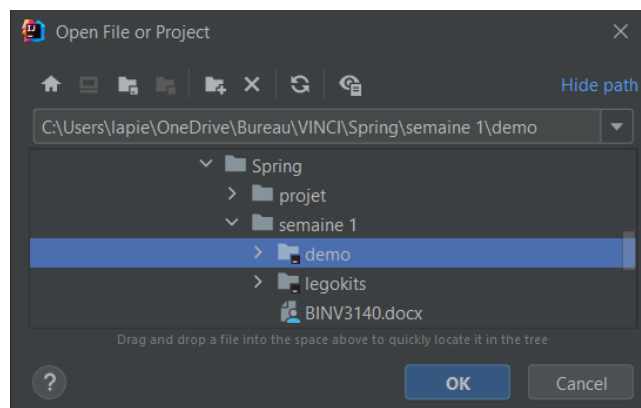
Une autre option est de copier les fichiers que vous avez reçus, de cliquer sur *src* dans la hiérarchie et de coller (CTRL + V ou clic droit >> « paste ») les fichiers. Vous devrez également confirmer en cliquant sur « Ok » dans la fenêtre qui s'ouvrira.

b) Vous recevez une archive de projet

Désarchivez le projet dans le dossier qui contient vos exercices du cours correspondant. Vous obtiendrez un dossier contenant toutes les données nécessaires à un projet java IntelliJ.



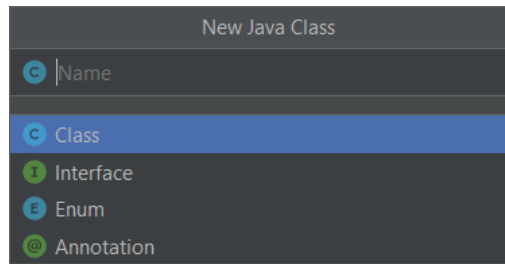
Choisissez simplement l'option « Open », et sélectionnez le dossier contenant le nouveau projet (il sera marqué d'un petit symbole indiquant que c'est un projet IntelliJ valide).



Ici, les dossiers « demo » et « legokits » sont deux projets valides, représentés par la petite icône noire avec un trait blanc.

c) Vous ne recevez aucun fichier

Effectuez un clic droit sur *src* (dans le module), et sélectionnez « New >> Java Class ». La petite fenêtre suivante apparaîtra alors :



Entrez simplement le nom de votre classe et appuyez sur Enter. Votre fichier est prêt, vous pouvez commencer à programmer !

Notes

- Par défaut, IntelliJ ouvre le dernier projet lancé au démarrage. Pour revenir à la page d'accueil, vous pouvez choisir File >> Close Project. Notez cependant que vous pouvez ouvrir ou créer un projet via le menu File également.
- Si vous vous trompez de nom en créant une classe ou un module, pas de panique. Faites un clic droit sur le nom à changer dans la hiérarchie, et choisissez « Refactor >> Rename... ». Entrez simplement le nouveau nom dans la petite fenêtre qui apparaît. Pour un module, un sous-menu apparaît avant, dans lequel vous devez choisir « Rename module and directory ». **Attention** : ne renommez **pas** vos fichiers via l'explorateur de votre ordinateur, cela risque de casser les dépendances des fichiers entre eux et de causer de nombreux problèmes de compilation pour votre programme !
- Une opération *refactor* consiste à modifier des parties du programme (noms de fichiers, noms de variable, positions de fichiers). Il faut être soigneux lors de ces opérations, car cela peut provoquer des erreurs dans les parties du programme qui utilisent ce qui a été modifié. IntelliJ propose donc d'effectuer ces opérations à votre place : quand vous déplacez un fichier, il regarde toutes les autres parties du code qui sont impactées et les modifie de manière à ce qu'aucune erreur ne se produise, par exemple.

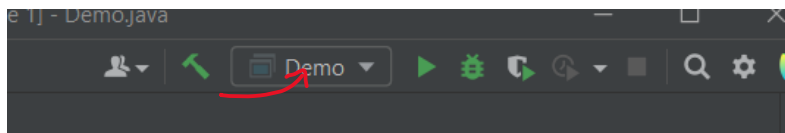
3. Exécuter un programme

Pour exécuter un programme, il suffit de cliquer le bouton « Play » vert en haut à droite de votre fenêtre. La première fois que vous voulez exécuter un programme, ce bouton est grisé, et vous ne pouvez pas cliquer dessus. C'est normal, car IntelliJ n'a pas encore pu configurer d'exécution.

La solution la plus simple est de vous rendre dans votre code.

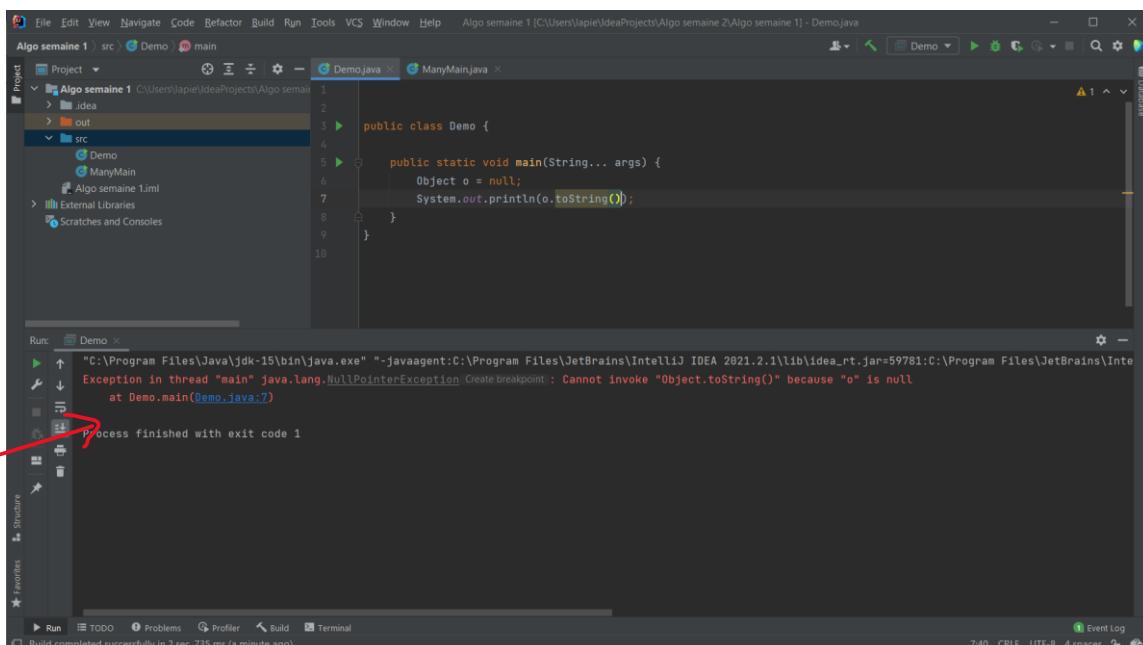
```
2  
3 ▶ public class Demo {  
4  
5 ▶ public static void main(String... args) {  
6     System.out.println("Hello, World!");  
7 }  
8  
9 }
```

Dans la marge de votre code, à côté de chaque nom de classe qui contient une méthode « main » ou à côté de la méthode main en question, il y a un petit bouton « Play » vert. Cliquer sur ce bouton permet d'exécuter la méthode main de cette classe (c'est-à-dire exécuter le programme). IntelliJ configurera automatiquement son environnement d'exécution pour que le bouton général en haut à droite corresponde à cette exécution.



Vous pouvez également retrouver toutes les exécutions déjà effectuées dans le menu de configuration en haut à droite (voir ci-dessus). Lorsque vous exécutez une méthode main par le bouton play de la marge, le nom de sa classe s'ajoute dans le menu déroulant. Vous pouvez ensuite choisir la classe que vous préférez exécuter via ce menu plutôt que retourner à chaque fois au bon endroit dans le code.

Lorsque vous exécutez votre programme, la fenêtre « Run » s'ouvre par défaut dans la partie inférieure de la fenêtre du projet.



C'est dans cette fenêtre que vous pourrez trouver les messages d'erreurs, les messages que vous voulez afficher (System.out.print), etc.

Les erreurs

Lorsque vous faites une erreur en programmant, elle peut, dans certains cas, être détectée automatiquement par IntelliJ avant même d'exécuter le code. C'est le cas par exemple de fautes de frappe, d'oubli de caractères spéciaux, etc.

Cependant, certaines erreurs ne sont pas détectables à l'avance et se produisent pendant l'exécution du programme. Comme dit plus haut, vous pouvez les trouver dans la fenêtre « Run » d'IntelliJ. Elles sont indiquées en rouge.

L'erreur indiquée précise de quel fichier elle provient, et à quelle ligne du fichier elle est intervenue, entre parenthèses, en bleu. Vous pouvez cliquer dessus pour être directement redirigé dans votre code à l'endroit où l'erreur est survenue.