

1 Prise en main d'IntelliJ Idea

IntelliJ IDEA est un **environnement de développement** : un logiciel conçu pour pouvoir programmer (développer). Il permet entre autre :

- d'ouvrir, modifier et sauvegarder des fichiers de sources,
- de structurer les sources pour les présenter de façon lisible, en particulier les erreurs présentes (celle qu'il est capable de détecter tout du moins),
- de compiler les programmes et démarrer leur exécution,
- de proposer des modifications, des simplifications, des corrections,
- d'assister à la recherche d'erreurs.

C'est donc un logiciel très complexe et intimidant, dont nous apprendrons à utiliser les fonctionnalités petit-à-petit.

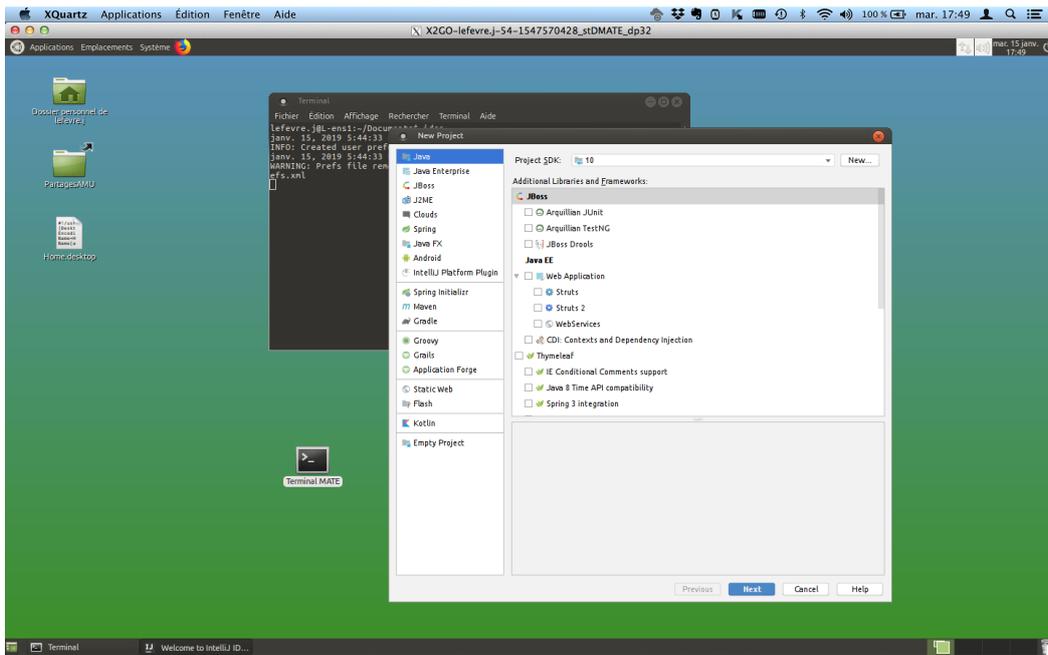
1.1 Création d'un projet

Un programme est constitué de multiples fichiers regroupés en un *projet*. Un projet est en général enregistré dans un répertoire dédié de votre espace personnel. Il vous est conseillé de créer un répertoire où vous placerez vos projets pour ce cours, par exemple `/home/XYZ/programmation1` (remplacez XYZ par votre identifiant). Nous commençons par créer un nouveau projet.

1. Commencez par ouvrir un terminal depuis le menu d'applications.
2. Déplacez-vous jusqu'à votre répertoire de travail, par exemple avec la commande : `cd ~/programmation1`
3. Entrez la commande `idea` qui va démarrer IntelliJ Idea.
4. Si c'est votre première utilisation d'IntelliJ Idea, le logiciel va vous poser plusieurs questions pour se configurer :
 - Une boîte de dialogue "IntelliJ IDEA User Agreement" apparaît. Cliquez sur "I confirm that I have read and accept the terms of this User Agreement"
 - Une autre boîte de dialogue vous demande "Are you a learner or an Educator?". Sélectionnez Learner.
 - Puis IntelliJ IDEA vous accueille.
5. Sélectionnez "Create new project"¹. Ceci aura pour effet de créer un nouveau projet en partant de rien.
6. Une fenêtre "New Project" s'ouvre. La configuration proposée doit faire apparaître Java dans la liste à gauche et 11 dans le "Project SDK"². Le 11 représente la version du langage Java qui sera utilisée pour le nouveau projet. Sélectionnez la valeur maximale proposée, et appuyez sur "Next".

1. Si vous avez déjà utilisé IntelliJ, il ouvre automatiquement le dernier projet modifié. Dans ce cas, l'option **Create new project** est dans le menu **File**. Ou bien vous pouvez fermer le projet avec le menu **File > Close Project**

2. SDK signifie *software development kit*, c'est un ensemble de librairie et d'outils pour le développement de projet en Java

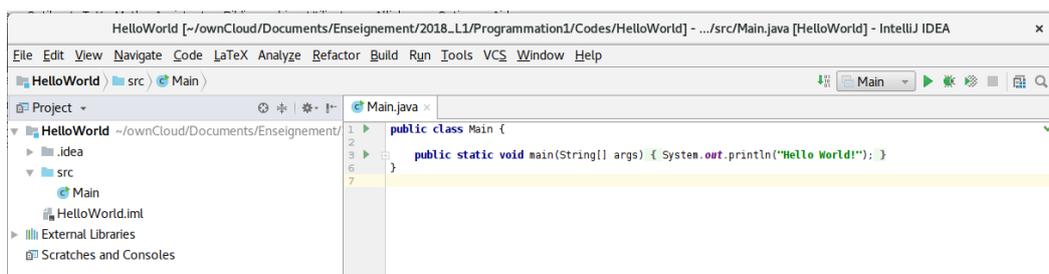


7. Cochez "Create a project from template", sélectionnez "Java Hello World" et appuyez sur "Next".
8. Écrivez HelloWorld pour "Project Name". Positionnez "Project Location" sur votre répertoire de projet, par exemple :
`/home/XYZ/programmation1/HelloWorld`
 puis validez.

Le projet est alors créé.

1.2 Compiler et exécuter le projet

1. Vous arrivez ensuite sur ce type d'interface :



À gauche, vous trouvez la structure du projet HelloWorld³. À ce stade, IntelliJ a créé un programme Java minimaliste complet. Ce programme ne fait rien d'autre qu'écrire "Hello World!" en sortie, mais vous pourrez le modifier par la suite. IntelliJ a organisé le répertoire HelloWorld avec plusieurs sous-dossiers. `src` contient les sources Java, `bin` et `out` (qui seront créés bientôt) contiennent les fichiers compilés, le répertoire `.idea` contient des fichiers de configurations propres à IntelliJ, et le fichier `HelloWorld.iml` contient une description du projet. Vous n'avez pas besoin de vous soucier d'autres choses que des fichiers présents dans le répertoire `src`.

Le fichier `Main.java`, qui se situe dans le répertoire `src`. C'est le contenu de ce fichier qui est affiché à droite. Vous pouvez le modifier et l'enregistrer, mais laissez-le tel quel pour l'instant.

3. cliquez sur 1: Project si cela n'apparaît pas

2. Vous pouvez compiler et démarrer ce programme de plusieurs façons, au choix :
 - en utilisant le raccourci clavier, en appuyant simultanément sur **Maj+F10**,
 - en utilisant le menu **Run > Run 'Main'**
 - en cliquant sur la flèche verte à gauche de la première ou de la troisième ligne du programme.
 - en cliquant sur l'icône flèche verte en haut à droite de la fenêtre.
 Après une courte attente, vous verrez s'afficher des messages dans la partie basse de la fenêtre (la console). En particulier, le message écrit par l'exécution du programme.

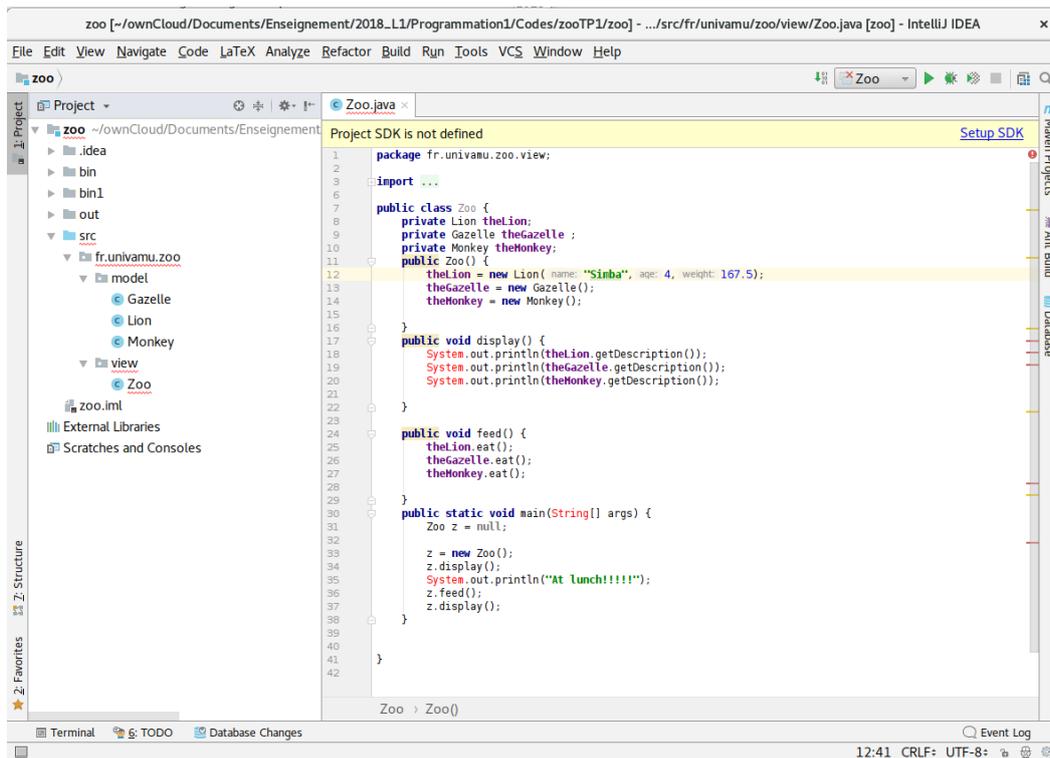


3. Vous pouvez essayer de modifier le programme, par exemple changer le message affiché. Relancez le programme pour vérifier que la modification a été prise en compte.
4. Essayer aussi d'introduire des erreurs. Comment IntelliJ signale-t-il les erreurs ? Que se passe-t-il si vous essayez de lancer un programme contenant des erreurs ?
5. Sauvegarder le fichier **Main.java** (apprenez à utiliser le raccourci clavier **Ctrl+S** pour cela). Puis fermer le projet (menu **File > Close project**).

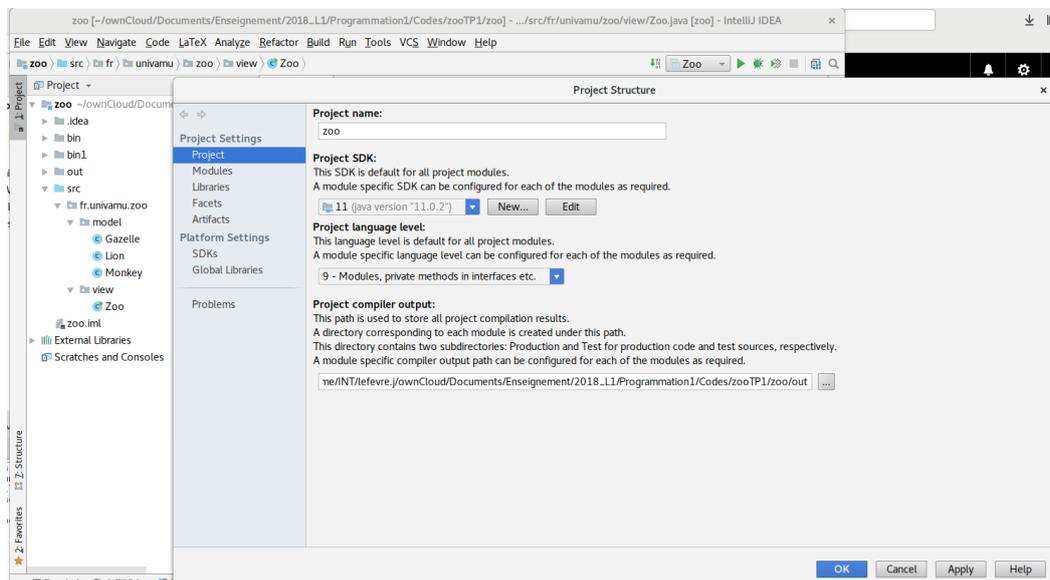
2 Premiers pas avec le TP du zoo

La suite du TP consiste à exécuter le programme du zoo vu en TD, et à ajouter l'animal éléphant.

1. Les fichiers sources du programme sont disponibles dans une archive sur Ametice. Téléchargez-les. Créer un répertoire `/programmation1/Zoo/`, puis extrayez-y les fichiers de l'archive. Vous devez avoir une hiérarchie de répertoires telle que la classe du lion, par exemple, soit `Zoo/src/fr.univamu.zoo/model/Lion.java`. Au besoin, ouvrir un explorateur de fichier pour organiser les répertoires correctement.
2. Créer un nouveau projet :
 - (a) Cette fois, le projet créé utilise des sources existantes, choisissez donc l'option "Import Project".
 - (b) Sélectionnez le répertoire **Zoo** que vous avez créé.
 - (c) Sélectionnez "Create from existing sources", puis "Next".
 - (d) Choisissez **Zoo** comme nom de projet, puis "Next".
 - (e) Cliquez sur "Next" plusieurs fois en laissant les choix par défaut.



- Si un bandeau jaune "Project SDK is not defined" apparaît, il faut choisir une version du kit de développement (par exemple la version 11). Cliquez sur le bandeau et choisissez la version. Utilisez le même numéro de version pour *Project language level*.



- Compiler et démarrer le programme, comme vous l'avez fait pour le premier projet. Vous devriez obtenir :

```
Run: Zoo
/usr/lib/jvm/jdk-11.0.2/bin/java -Didea.launcher.port=36181 -Didea.launcher.bin.path=/home/INT/lefevre.j/Documents/Enseignement/Java/ideaIU-2018.1.5/id
I'm a Lion my name is Simba, I'm 4 old, I weigh 167.5 kg .
I'm a Gazelle my name is Gazelle, I'm 1 old, I weigh 12.0 kg and my horns are 9 cm long
I'm a Monkey my name is Monkey, I'm 1 old, I weigh 1.0 kg .
At lunch!!!!!!
I'm a Lion my name is Simba, I'm 4 old, I weigh 169.4 kg .
I'm a Gazelle my name is Gazelle, I'm 1 old, I weigh 12.075 kg and my horns are 9 cm long
I'm a Monkey my name is Monkey, I'm 1 old, I weigh 1.3 kg .
Process finished with exit code 0
```

3 Ajout d'une classe Elephant

1. Créez la classe **Elephant** (la majuscule est importante!). Elle doit être ajoutée dans le dossier `model`. Pour cela, dans IntelliJ, faites un clic droit sur le dossier `model`, puis **New > Java class**. Entrez le nom de la nouvelle classe. IntelliJ crée une ébauche de la classe.
2. Ajoutez les propriétés et les méthodes de l'éléphant, vues en TD. Inspirez-vous des autres animaux pour respecter la syntaxe de Java. Remarquez comment IntelliJ indique les erreurs en soulignant en rouge⁴.
3. Intégrez cette nouvelle classe dans le zoo. Le zoo doit avoir un éléphant, qui agit en même temps que les autres animaux.
4. Relancer à nouveau le zoo pour vérifier que votre éléphant mange comme les autres animaux et qu'il grossit conformément au cahier des charges.
5. Faites que le programme endorme tous les animaux après les avoir nourri. Vérifier que les animaux ont bien dormi.

4. Ce n'est pas toujours bien visible. Conseil : modifiez la façon dont les erreurs sont affichées, menu **File > Settings**, onglets **Editor > Color Scheme > General** à gauche, puis **Errors and warnings > Error** au centre. choisissez un fond (background) rouge pale. Modifiez dans la même catégorie **Unknown symbol** de la même façon.