

TP1 – Partie 1

« NetBeans »

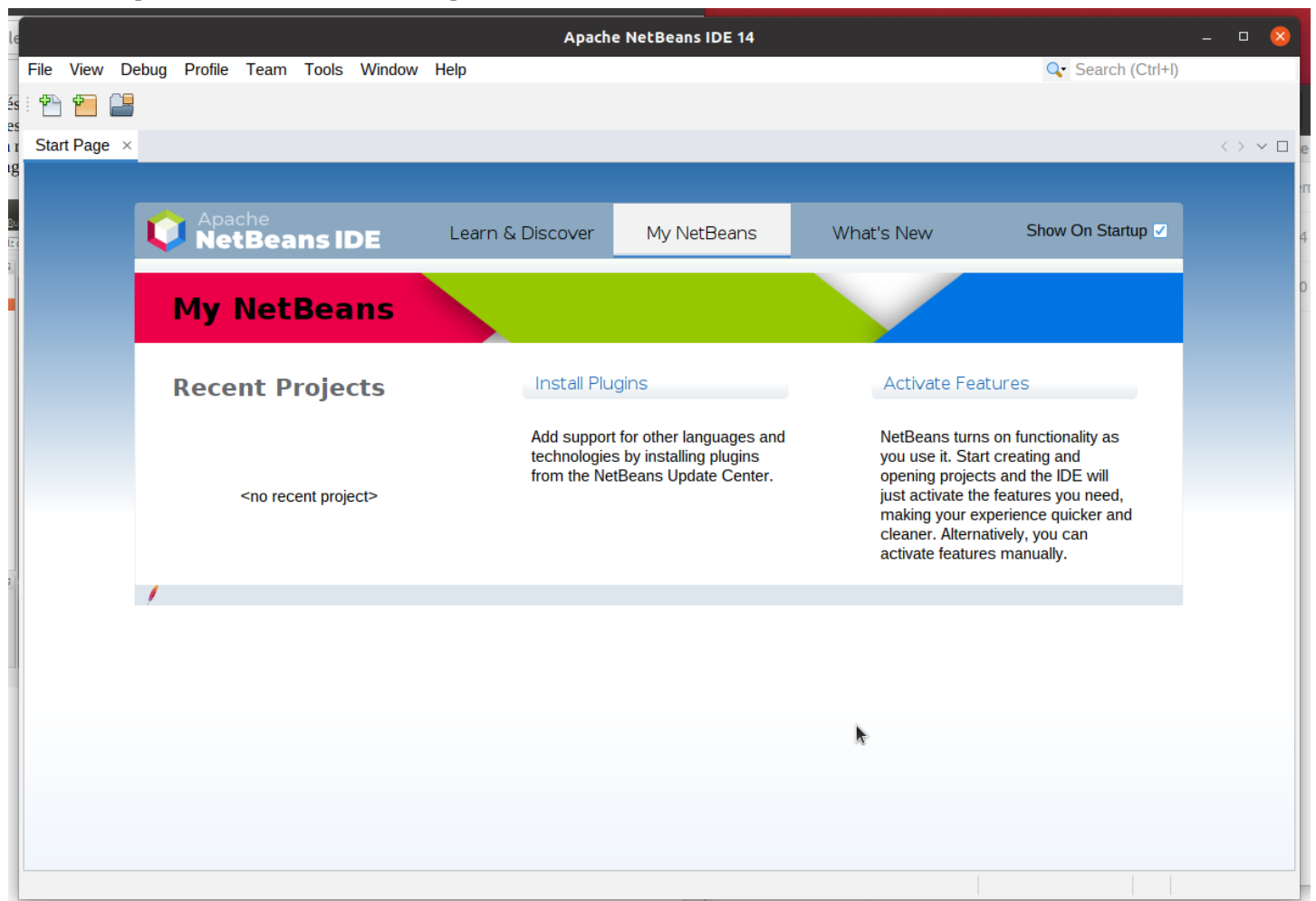
NetBeans est un environnement de développement intégré (*IDE*) adapté au développement d'applications *Java*, et permettant de réaliser des applications graphiques (bibliothèques *Awt*, *Swing* et *JavaFX*) d'une manière similaire à ce que vous avez vu avec *VS Code* l'an dernier pour le langage *Python*.

NetBeans est un logiciel libre (licences CDDL et GPL), vous pouvez donc l'installer sur vos machines personnelles. *NetBeans* est lui-même écrit en *Java*, il fonctionnera donc sur tout type de plateforme (*Windows*, *Linux*, *MacOS*) mais il faut auparavant y avoir installé un *JDK* (environnement de développement *Java*). Pour en savoir plus pour vos installations perso vous pourrez suivre le tuto « Installations *Java* sur vos ordinateurs personnels » sur la page e-campus du cours de *POO*.

Dans cette première partie de votre premier TP *Java* sur les ordinateurs de l'ENAC nous allons observer l'organisation et utiliser les mécanismes de base de *NetBeans*. De nombreux documents et tutoriels sont disponibles sur internet pour aller plus loin dans son utilisation.

Écran d'accueil

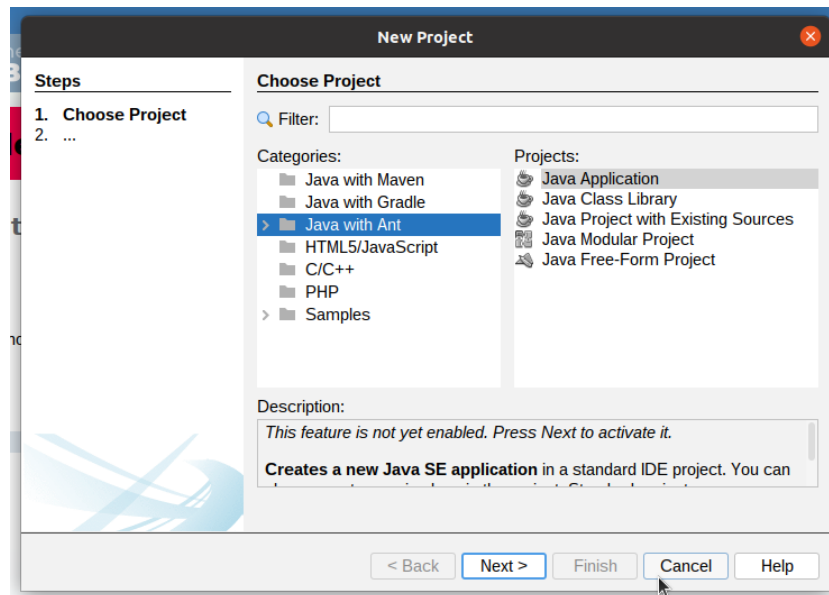
- Au premier lancement *NetBeans* présente l'écran d'accueil suivant :



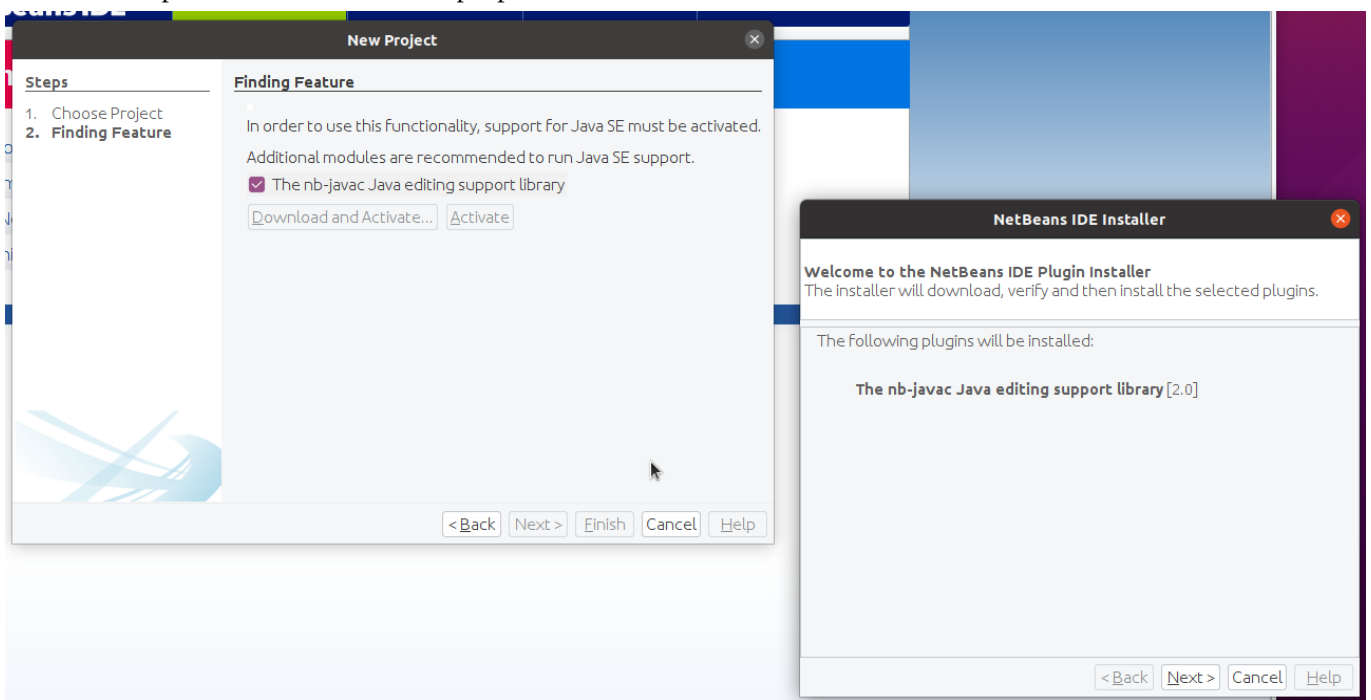
- Les menus et boutons ne sont pas au complet sur cet écran de lancement. Il faut activer les fonctionnalités associées au langage de programmation que vous allez utiliser (une fois pour toute, cette opération n'aura pas à être renouvelée pour ce langage). Pour ce faire il suffit de créer votre premier projet *Java*...

Créer votre premier projet

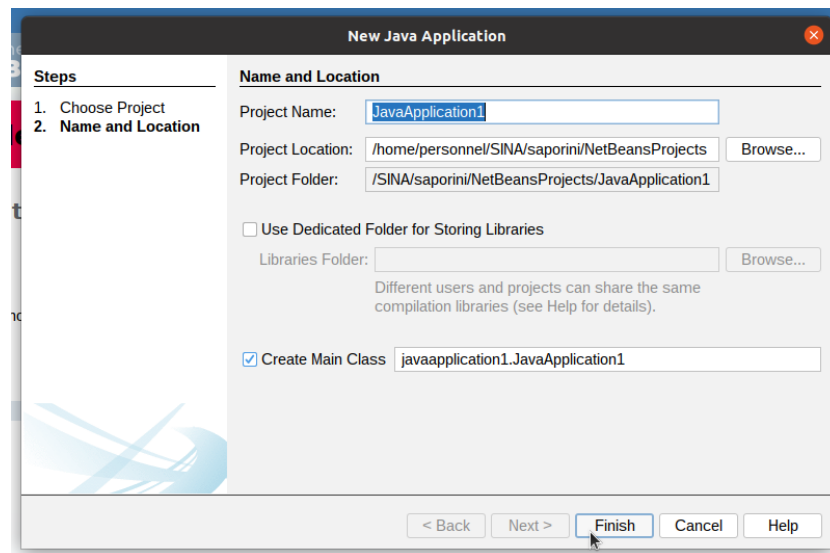
- Cliquez sur menu *File* puis sélectionnez *New Project*,
- Choisissez la catégorie *Java with Ant*, puis *Java Application* et cliquez sur le bouton *Next* :



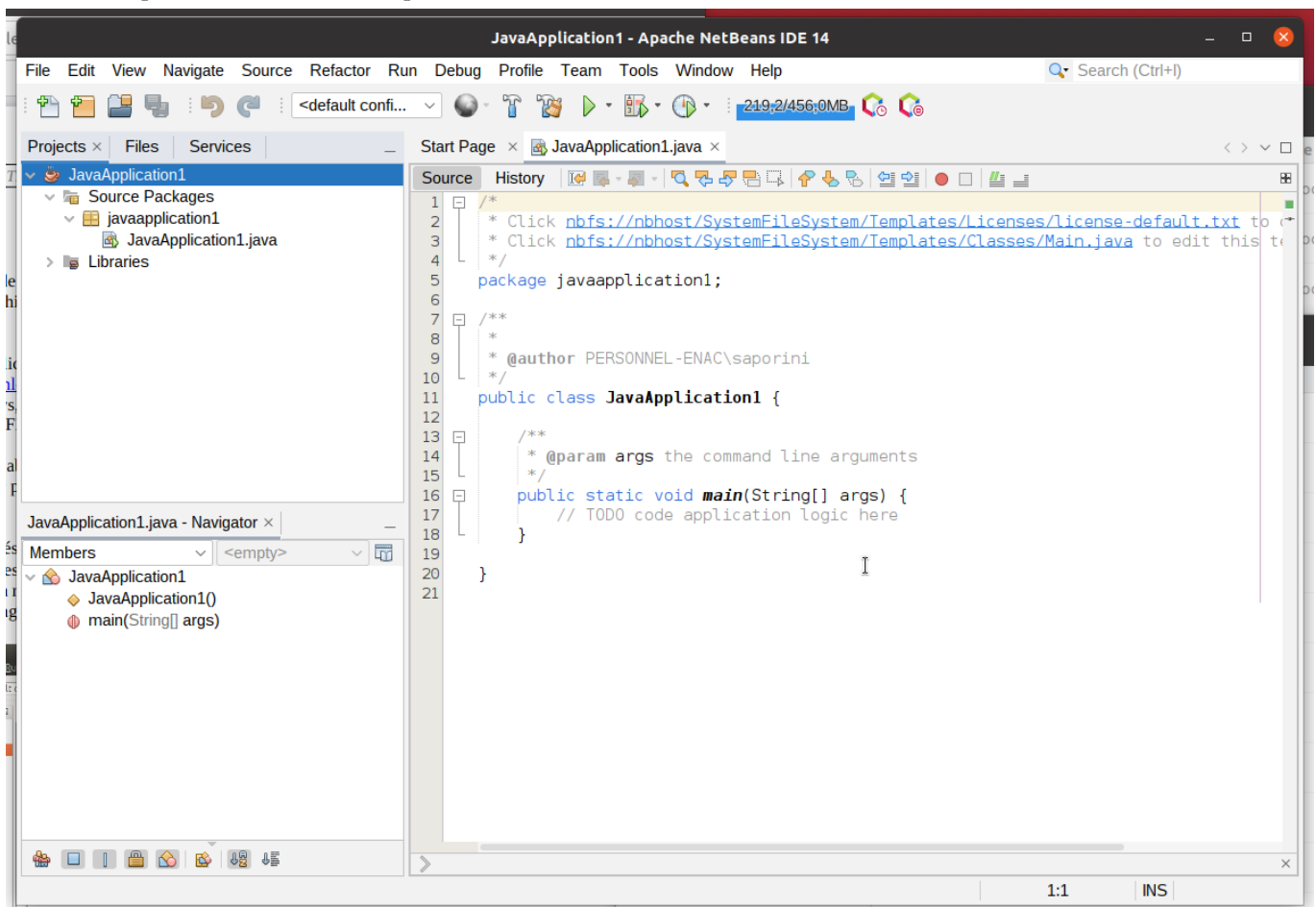
- *NetBeans* vous propose alors d'activer le support de *Java SE*, acceptez en cliquant sur le bouton *Activate*. L'activation peut durer un certain temps, patientez...



- Une fois l'activation réalisée *NetBeans* affiche une fenêtre pour vous permettre de saisir les paramètres du projet à créer. Par défaut une classe unique sera générée, portant le même nom que votre projet. Si vous voulez un autre nom, ou pas de classe du tout (dans le cas où vous souhaitez par exemple importer du code pré-existant), ou une imbrication particulière de paquetages (observez celui qui est proposé par défaut)... il faut modifier les paramètres de cette fenêtre de création du projet. Mais vous pouvez bien sûr modifier tout ça après coup en ajoutant/supprimant ce que vous voulez. Pour votre premier programme laissez tous les paramètres par défaut (screenshot ci-dessous) et cliquez sur le bouton *Finish*...



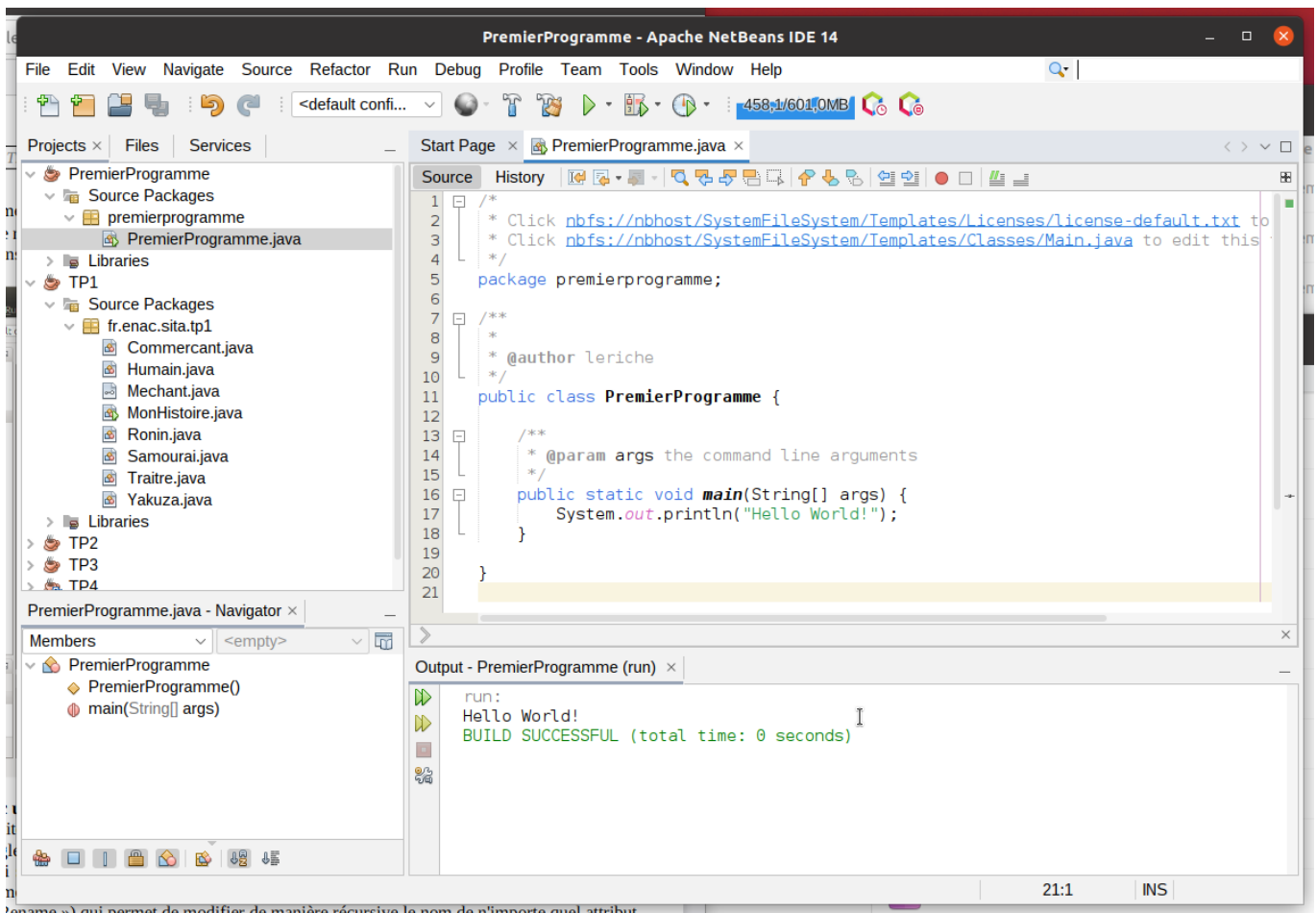
- *NetBeans* crée alors votre projet à l'emplacement choisi et génère le squelette de la classe principale, contenant une méthode `main(String[] args)` vide. Le screenshot ci-dessous montre ce que vous devriez obtenir en laissant les paramètres de création par défaut :



Exécuter une application

- Complétez le code de la méthode `main(String[] args)` pour afficher le message « Hello World ! », puis lancez l'application soit en cliquant sur la flèche verte de la barre de menus principale, soit en appuyant sur F6, soit en faisant un clic droit sur la classe principale et en sélectionnant l'item de menu *Run File*. Observez alors les opérations de compilations puis le lancement et les affichages correspondants dans la console (vue *Output* de *NetBeans*).

- Sur le screenshot ci-dessous vous pouvez voir le résultat de la question précédente dans un espace de travail montrant plusieurs projets ouverts, regroupés dans la vue *Projects*. Identifiez les symboles qui discriminent visuellement les projets, les paquetages, les classes java classiques et les classes contenant un `main()`. A quoi sert la vue *Members* ?

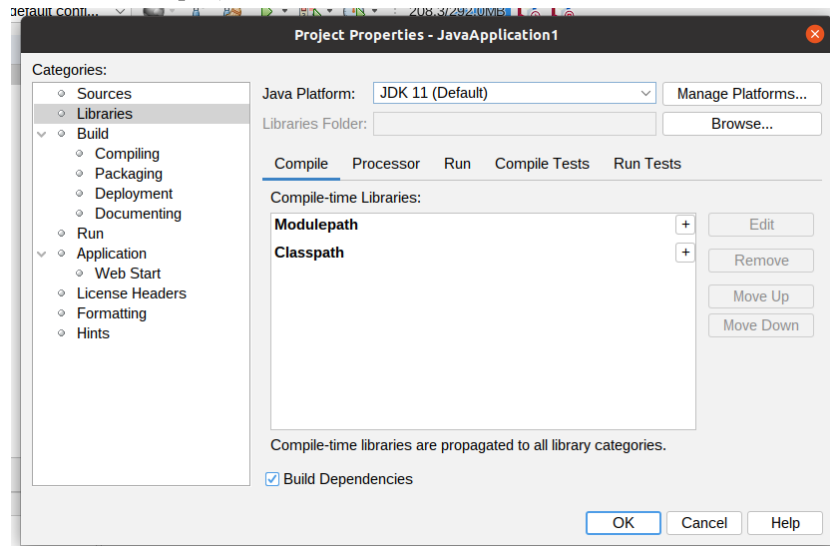


Augmenter sa productivité avec un IDE

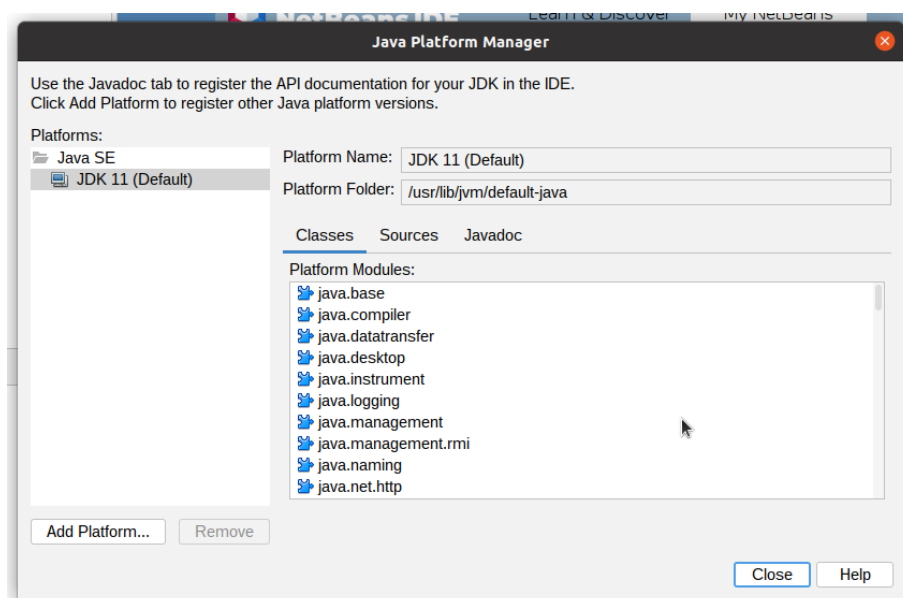
- Observez les fonctionnalités disponibles dans les menus *Source* et *Refactor* (également accessibles par clic droit depuis un onglet contenant du code). Vous utiliserez dans la suite **abondamment** la fonction de reformatage du code (qui suit les *guidelines* de Java), mais aussi la génération automatique de parties du code (*Insert Code*) qui permettent par exemple de générer des constructeurs à partir des attributs de la classe, ainsi que le renommage (*Refactor/Rename*) qui permet de modifier de manière récursive le nom de n'importe quel attribut/méthode/constante et toutes les références qui l'exploitent dans tout le projet.
 - Ajoutez un attribut entier bidon, générez un constructeur puis les accesseurs en lecture,
 - Renommez cet attributs.
- Enfin, l'*IDE* propose des fonctionnalités d'auto-complétion (ctrl+espace), de documentation (ctrl+maj+espace) et des suggestions d'amélioration lorsque votre code n'est pas optimal (warnings) ou des propositions de correction lorsque votre code ne compile pas.
 - Enlevez la majuscule de « System » et regardez la réaction de l'*IDE* lorsque vous passez votre souris sur le mot en erreur souligné.
 - Ajoutez une ligne de code commençant par « System. », observez qu'après un temps de latence court, l'*IDE* vous propose plusieurs possibilités de complétion (si ça ne se produit pas, forcez leur apparition en pressant ctrl+espace).

Changer de plateforme Java

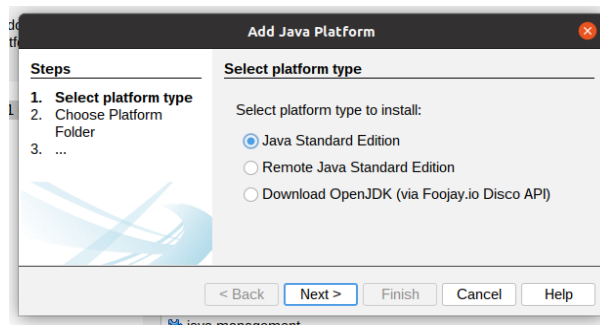
- Faites un clic droit sur le nom de votre projet dans la vue *Projects* et sélectionnez l'item *Properties*. NetBeans affiche alors toutes les propriétés de votre projet. Choisissez la catégories *Libraries*. Vous devriez voir que la plateforme Java utilisée pour votre projet (la version de Java) est la version par défaut, c'est à dire le *JDK11* sur les ordinateurs de l'ENAC (le *JDK18* sur vos ordinateurs personnels si vous y avez déjà installé Java en suivant les directives laissées sur e-campus).



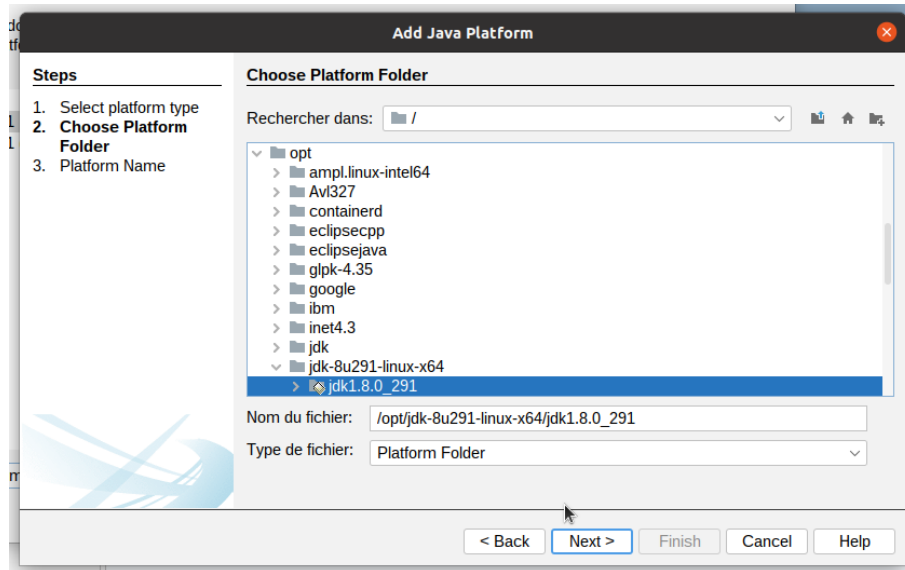
- Cette version de Java fonctionnera pour la plupart des TP, mais pas pour ceux comprenant des interfaces graphiques en *JavaFX* (nous verrons plus tard pourquoi). Heureusement, on peut installer autant de versions de Java que l'on veut, et en utiliser des différentes dans différents projets. Vous allez donc créer une nouvelle plateforme Java en cliquant sur le bouton *Manage Platforms* (ou en sélectionnant le menu *Tools > Java Platforms*). Vous obtiendrez la fenêtre suivante :



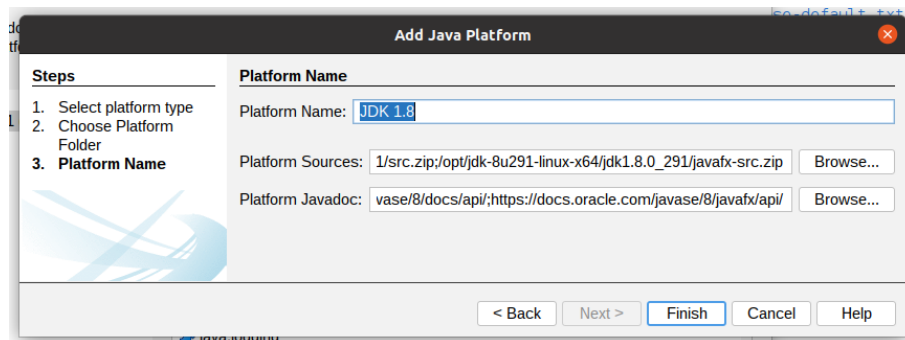
- Cliquez sur le bouton *Add Platform* en bas à gauche. Laissez le choix par défaut *Java Standard Edition* dans la fenêtre qui apparaît, et cliquez sur le bouton *Next* :



- Choisissez le dossier /opt et sélectionnez le *JDK8* visible en bleu sur le screenshot ci-dessous, puis cliquez sur le bouton *Next* :



- Une fenêtre récapitulative identifiant un « JDK 1.8 » doit apparaître. Cliquez sur le bouton *Finish* :



- Dans tous les TP à venir vous utiliserez cette version de Java en vérifiant, et en modifiant le cas échéant la plateforme Java dans les propriétés de vos projets.
- Vous pouvez maintenant passer à la partie 2 du TP et coder les histoires de samourais vues en TD de COO...