

Une introduction à l'éditeur Java Eclipse

Hello World

Eclipse est un IDE (Integrated Development Environment). Il contient un espace de travail qui permet le développement de programmes simples ou de plug-ins. Dans le développement Java avec Eclipse, nous allons créer des *projets* Java. Un projet peut contenir plusieurs classes, dont une sera la classe principale (dans laquelle nous implémenterons notre méthode main). Lors de la compilation, notre code sera transformé dans du bytecode.

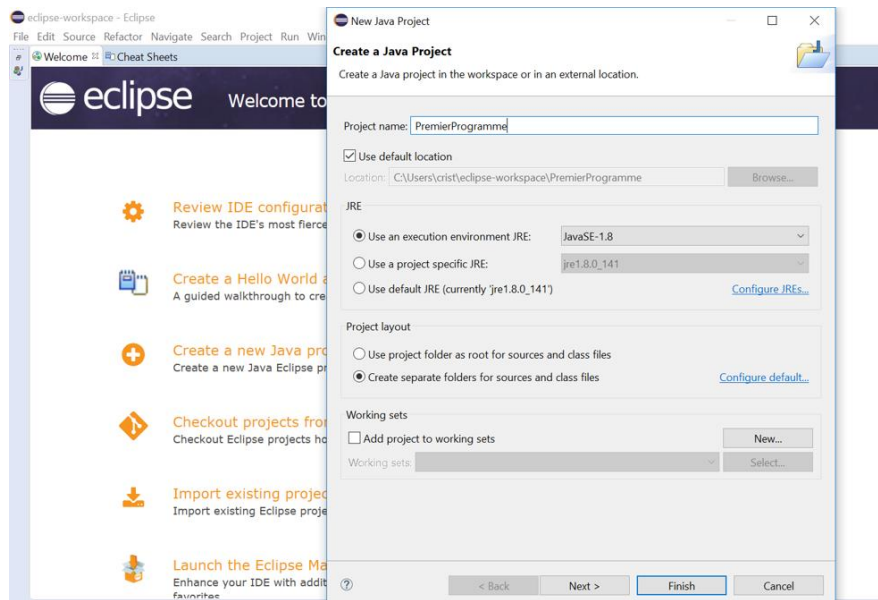
Pour ceux qui veulent installer Java Eclipse, il est à trouver ici : <https://www.eclipse.org/downloads/> .

Voire également le tutorial donné lors du CM.

Nous allons créer une première application, pour bien comprendre ce qu'est le bytecode avant de commencer créer le monde des pokémons !

Vous allez travailler dans une machine virtuelle. Choisissez notamment la première dans la liste. Celle-là aura Eclipse déjà installée.

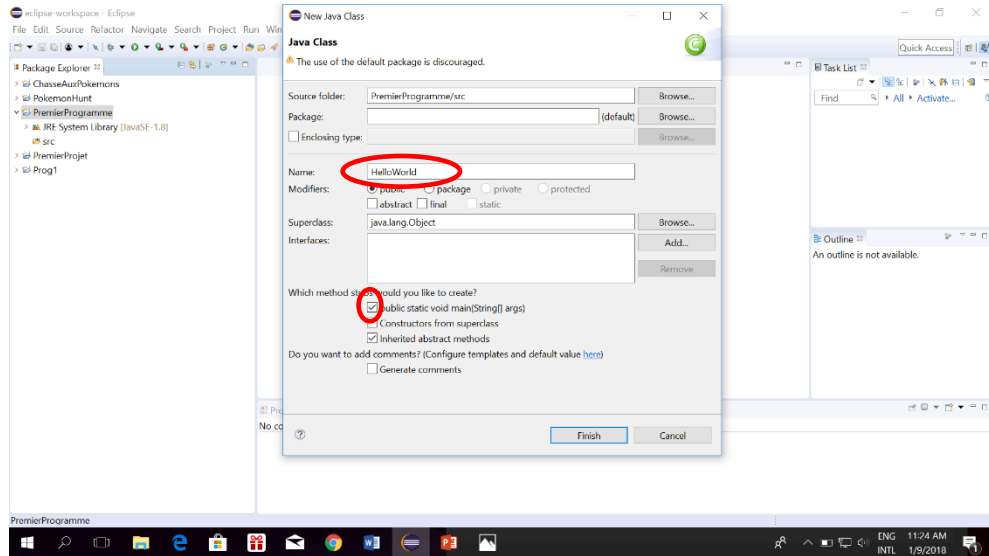
0. Nous allons créer un programme très simple, qui dit "Hello world !". Cela peut être réalisé en Java avec une seule classe (qui sera la classe principale). Ouvrez l'éditeur Eclipse. Puis, créez un nouveau projet Java et nommez-le "PremierProgramme". (File >> New >> Java Project)



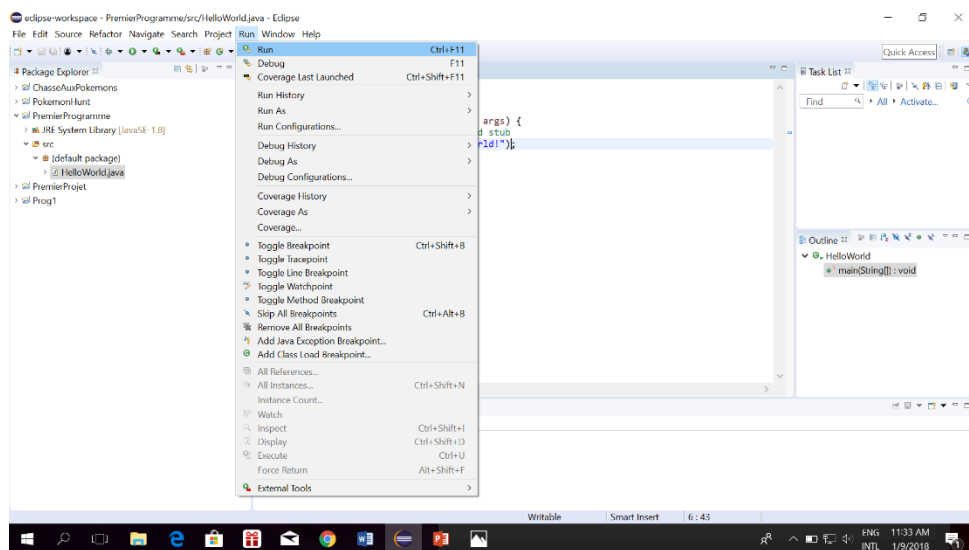
Cliquez sur Finish (ou, si vous voulez explorer l'environnement, vous pouvez également cliquer sur Next pour savoir ce qu'on peut configurer lors de la création d'un projet).

Maintenant votre projet est créé. Fermez les volets de Welcome et CheatSheets. Vous allez voir votre projet dans l'onglet à Gauche, qui s'appelle Package Explorer.

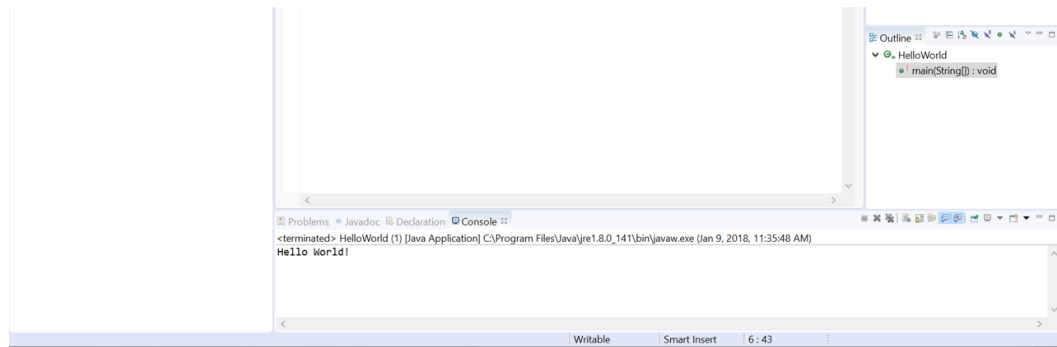
Vous allez créer une classe principale au sein du projet PremierProgramme (Rappel : c'est quoi une classe principale ?). Pour faire ceci, nous allons créer une nouvelle classe (File >> New >> Class), et nous allons cocher l'option pour ajouter la méthode public static void main lors de la création de la class. On va appeler notre classe principale "HelloWorld".



- Dans la méthode main, écrivez la commande qui affichera le message "Hello world !" lors de l'exécution.
- Pour compiler et exécuter votre programme, allez sur Run >> Run .



- Si votre programme marche, vous allez avoir un retour sur la console, en bas de l'éditeur.



- Nous allons maintenant regarder le bytecode de ce programme. Pour cela, il faut tout d'abord localiser le fichier javap.exe sur notre installation Java. Il sera localisé dans le jdk, dans le répertoire bin. Localisez-le.
- Ouvrez le répertoire correspondant à PremierProgramme. Vous allez voir qu'il est partagé en deux parties : src et bin. Ouvrez le répertoire src. Puis faites la même chose pour le répertoire bin. Que constatez-vous ?
- Ouvrez une console de commande (en utilisant la commande cmd). Dans ce terminal, allez dans le répertoire PremierProgramme\bin
- Maintenant exécutez la commande suivante : "...\javap" -c HelloWorld.class
Attention : au lieu des ... il faut écrire le chemin complet jusqu'à javap.exe.
Le texte affiché est le bytecode de votre programme.

TP 1 : Les pokemons se disent Bonjour !



Dans ce TP nous allons commencer notre chasse aux pokemons. Nous débutons avec des pokemons très simples qui vont se dire bonjour. Après, nous ajouterons un attribut diurne/ nocturne qui va dicter quand les pokemons seront réveillés pour se dire bonjour. Finalement, nous terminerons par créer des joueurs, qui auront un maximum de 5 pokemons chacun.

Pour les pokémons, nous allons utiliser le Pokedex : <https://pokemondb.net/pokedex/all> .

Les énoncés utilisent le Pokedex en anglais, mais si vous voulez utiliser les noms français des pokemons allez-y.

1. Créez un nouveau projet MaChasseAuxPokemons.
2. Nous allons organiser notre répertoire. Allez sur File >> New >> Package.

Les noms des packages commencent par convention avec une minuscule. Par exemple, j'ai mis comme package `tpl`. Cliquez sur Finish.

Cherchez à gauche. Dans votre nouveau projet ChasseAuxPokemons, vous devriez voir dans les sources (src) le nouveau package `tp1` . Cliquez là-dessus.

Puis, cliquez sur File >> New >> Class (et lisez l'énoncé de l'exercice numéro 3 pour continuer).

3. Ajoutez une classe Pokemon. Les objets de cette classe auront les attributs suivants :

Un attribut `nom` de type `String` ;

Un attribut `type` de type `String` ;

Un attribut `niveau` de type `int` .

- Déclarez ces attributs (rappel : c'est une bonne pratique de faire les attributs privés)
- Puis, créez un constructeur avec la signature :

```
Pokemon(String, String, int)
```

4. Dans la classe Pokemon : ajoutez une méthode `String toString()` qui affichera le texte suivant :
<nom du pokemon> est un pokemon de type <type du pokemon>, qui a le niveau <niveau>.

Par exemple si on a un pokemon avec : `nom = "Piplup"`; `type = "EAU"`; `niveau = 5`, la méthode `toString()` affichera :

Piplup est un pokemon du type EAU qui a le niveau 5.

Attention : nous avons simplifié l'univers des pokemons : on ne permet pas à un pokemon d'avoir plusieurs types (contrairement au Pokedex, qui précise qu'un pokemon pourrait avoir au plus deux types, ce qui est le cas de Rowlet, par exemple). Si vous voulez créer un pokemon du Pokedex qui a deux types, il faut choisir lequel de ces deux types vous voulez garder.

5. Créez une nouvelle classe principale ChasseAuxPokemons, avec une méthode main.
 - Dans le main de cette classe principale, créez trois pokémons : Piplup (type EAU, niveau 5) , Rowlet (PLANTE, niveau 10) et Totodile (EAU, niveau 8).
 - En utilisant la méthode toString() affichez les caractéristiques de ces trois pokemons.
 - Puis, sans utiliser la classe Pokemon, faites afficher dans la console le texte :

Piplup dit bonjour !
Rowlet dit bonjour !
Totodile dit bonjour !

6. Maintenant, nous allons écrire dans la classe Pokemon une méthode du type void avec la signature : direBonjour(). Celle-ci va afficher le salut de chaque pokemon : "<nom du Pokemon> dit Bonjour !"
 - Réécrivez la méthode main pour utiliser cette méthode là au lieu de System.out.println.
 - Exécutez votre code. Quel est le bytecode de la classe principale?
7. Ajoutez un attribut **diurne** à la classe Pokemon, qui va indiquer si un pokemon est diurne ou nocturne.
 - Quel type de variable utiliserons-nous ?
 - Modifiez le constructeur de la classe Pokemon pour prendre en compte l'ajout de ce nouvel attribut. Quelle est la signature de votre nouveau constructeur ? (Attention : modifier le constructeur aura des conséquences sur les autres classes et méthodes qui utilisent des objets de la classe Pokemon. En Eclipse cela sera indiqué par des petits croix rouges. Ne vous inquiétez pas, nous adapterons notre code ensuite.)
 - Modifiez la méthode direBonjour, qui va maintenant prendre en entrée une variable de type String. La signature sera : void direBonjour(String periode). Si la variable temps stocke la valeur "jour", alors les pokemons diurnes diront Bonjour, tandis que les pokemons nocturnes diront zzzzzz. Si la variable temps contient la valeur "nuit", alors les pokemons nocturnes diront Bonsoir et les pokemons diurnes diront zzzzzz.
 - Modifiez la méthode main pour utiliser correctement la nouvelle méthode direBonjour().

8. Nous allons exécuter notre programme pour simuler le salut des pokemons de jour et de nuit.
 - On commence par initialiser une valeur de type String dans la méthode principale, qu'on changera à la main. Nommez cette variable `periodeJournee` et initialisez-la à la valeur "jour" et exécutez votre programme pour cette valeur.
 - Puis changer la valeur de `periodeJournee` à "nuit" et exécutez le code encore une fois.

9. Cette fois-ci, nous allons utiliser la première composante de la variable `args[]` pour indiquer la période de la journée. Commentez la ligne / les lignes de code qui déclarent et initialisent la variable `tempsDuJour`. Puis, au lieu d'utiliser la variable `tempsDuJour` en entrée de la méthode `ditBonjour`, on utilisera le premier élément de `args[]`.
 - Quel est cet élément ?
 - Au lieu d'exécuter le programme normalement, nous allons ouvrir le menu d'exécution (Run >> Run Configurations), et nous allons choisir l'option d'exécuter une configuration. Dans le menu de la configuration, allez sur Arguments et mettez la valeur "jour" (avec des guillemets). Puis confirmez la modification en appuyant sur le bouton Appliquer. Ensuite exécutez le programme.
 - Quel est le bytecode de ce programme ?

10. À présent, nous voudrions ajouter des joueurs au jeu des pokemons. Nous avons besoin d'une nouvelle classe `Joueur`. Les attributs caractérisant les objets de cette classe seront :
 - Le nom
 - Le prénom
 - L'age
 - Un tableau contenant leurs pokemons. La taille de ce tableau sera limitée à 5.
 - Implémentez cette classe et ajouter un constructeur avec la signature :
`Joueur (String, String, int, Pokemon[])`