

TP 1 : Un début à la programmation Java

Ce TP aura plusieurs parties.

Premièrement on fera une première installation de Java Eclipse sur votre machine. Ceci vous montrera les étapes à suivre pour une installation complète de cet environnement de développement.

Comme une deuxième étape nous allons suivre un petit tutoriel pour concevoir un première programme très simple en Java.

Finalement on va faire les premiers pas pour créer nos pokemons, en apprenant également quelques étapes de base pour la programmation en chaque TP.

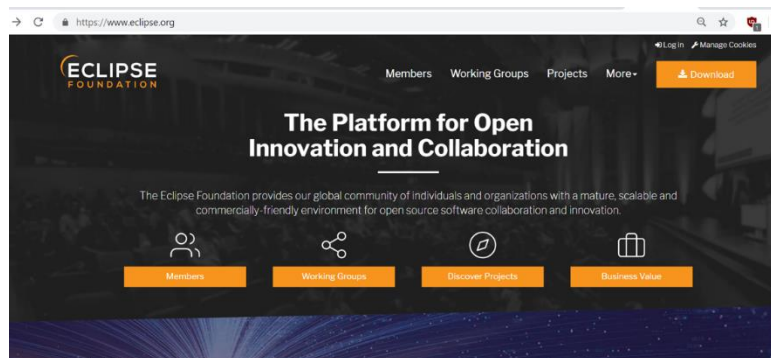
Prêts ?

Ben allons-y !

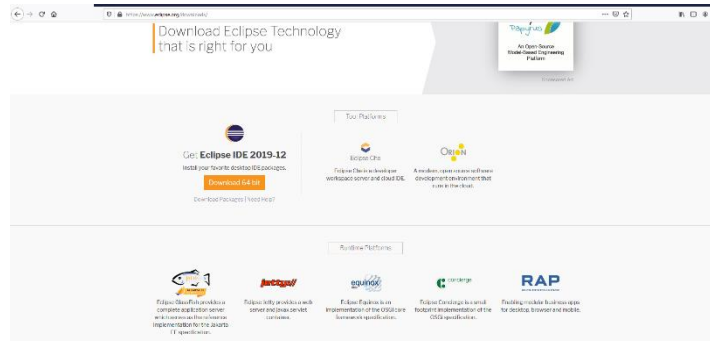
L'installation (sous Windows) de Java Eclipse

Eclipse est un IDE (Integrated Development Environment). Il contient un espace de travail qui permet le développement de programmes simples ou de plug-ins. Mais avant commencer le travail, il faut installer Eclipse avec les deux composantes qui nous intéressent : le JDK et le JRE.

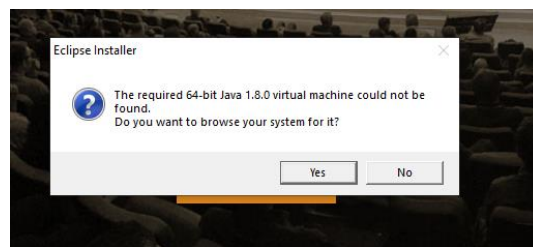
Nous allons trouver Eclipse sur le site <https://www.eclipse.org/> . Trouvez le bouton de Téléchargements en haut à droite sur l'écran :



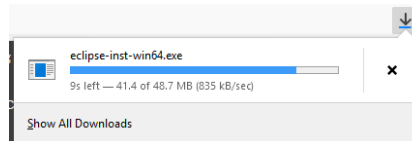
On peut choisir d'installer une des diverses versions d'Eclipse présentées sur l'écran suivant :



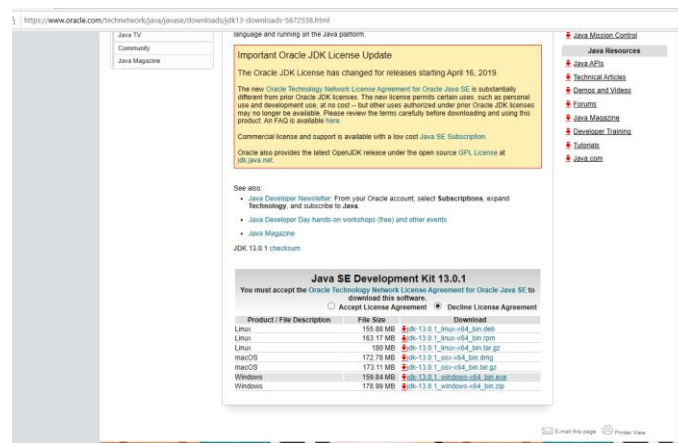
La version que vous allez installer dépend de votre ordinateur. Les instructions ci-dessous concernent la version sur 64 bits, mais vous pouvez toujours utiliser une autre si vous en avez besoin.



Après d'avoir téléchargé le fichier, faites-le exécuter. Ceci vous donnera une erreur, car vous n'avez pas encore le JDK (Java development kit) installé.

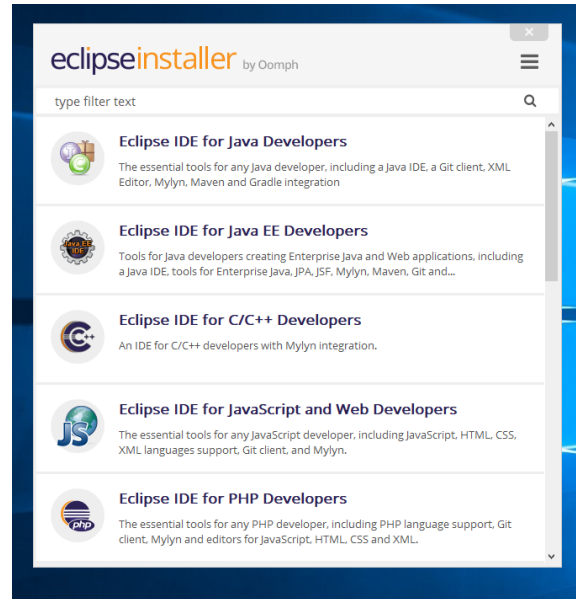


Cliquez sur Non dans le message d'erreur. Pour certaines ordinateurs et Systèmes d'Exploitation, votre navigateur vous redirigera directement vers un site Web où vous pourrez télécharger le JDK 13. Sinon, trouvez les sources en utilisant un moteur de recherche. Voici l'écran que vous devriez voir.

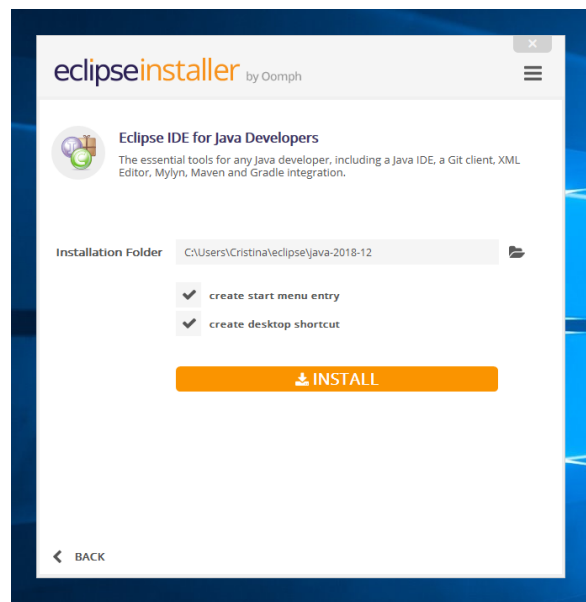


Acceptez les termes de la licence et trouvez la version exécutable pour votre système d'opération. Téléchargez ce fichier et faites-le installer.

Une fois le JDK installé, re-faites un click sur l'installer d'Eclipse. Après un délai, votre installation commencera en vous demandant quelle version de Java vous voulez installer.

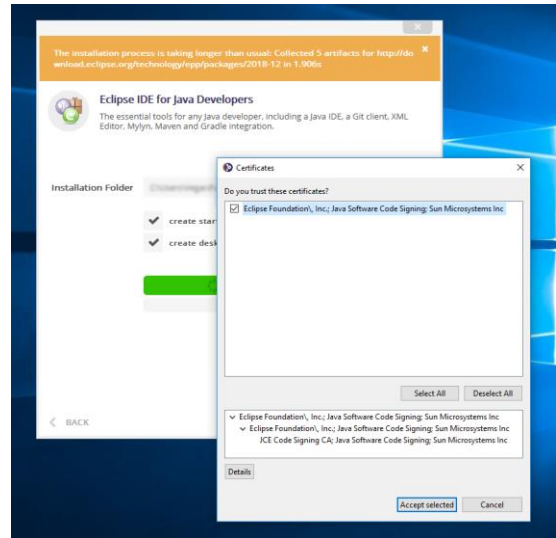
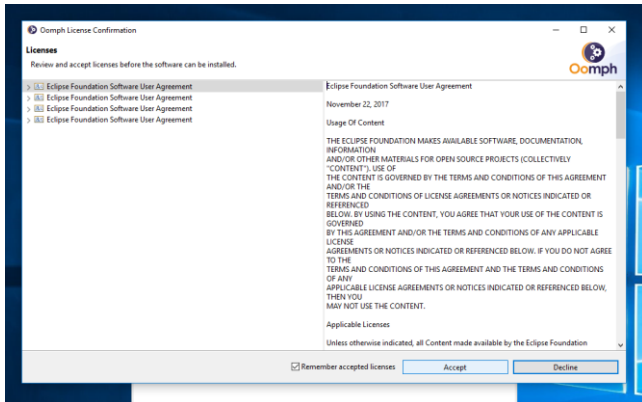


Pour cette installation nous allons choisir la première option, Eclipse IDE for Java Developers. Je préfère mettre des icons dans le menu de start ainsi que sur le desktop, mais c'est à vous à choisir.



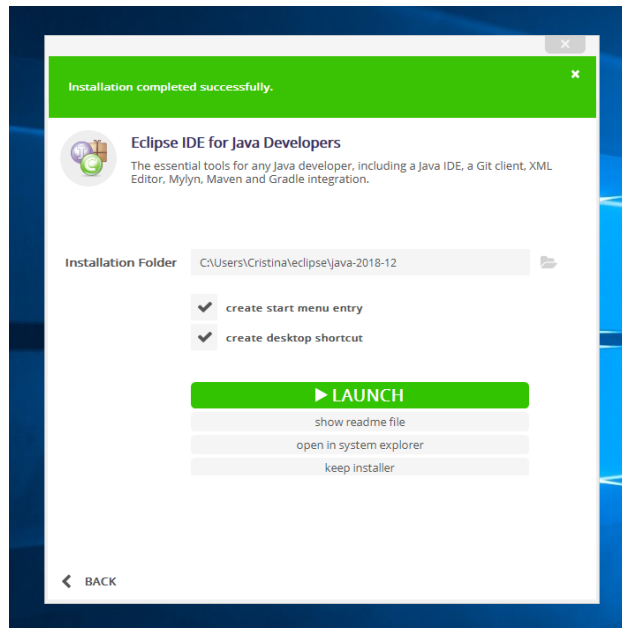
Après d'avoir accepté les conditions d'utilisation, l'installation commencera.

Pendant cette dernière vous aurez quelques messages qui interrompront l'installation, par exemple l'acceptation des conditions de la licence ou la demande d'acceptation du certificat pour Eclipse.



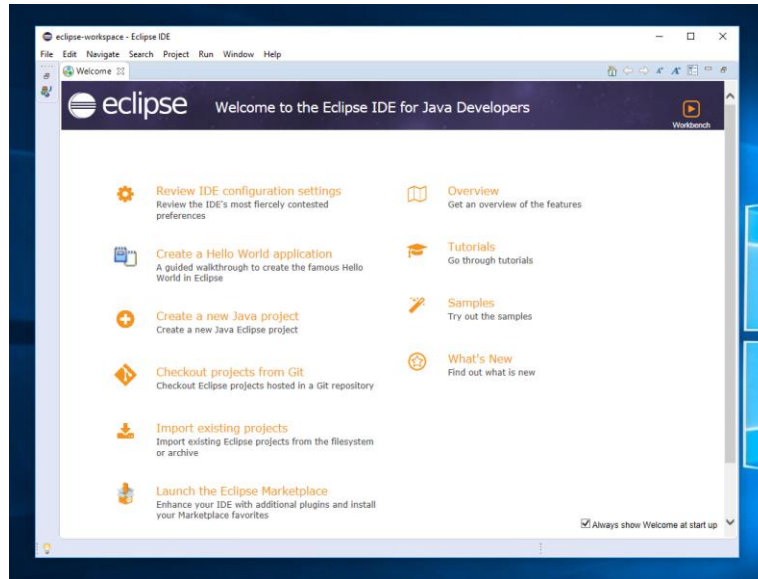
Vous allez devoir accepter les conditions et puis sélectionner le certificat et l'accepter.

Et finalement, vous avez installé Eclipse ! Il ne vous reste qu'à cliquer sur LAUNCH !



Il va falloir qu'on choisisse un espace de travail (workspace). Celui-ci sera un répertoire où on va sauvegarder tous nos projets. Je vous conseille de choisir un répertoire facilement accessible et de lui faire une shortcut sur votre desktop. Eclipse vous demandera à chaque fois quel répertoire vous voulez utiliser pour vos projets Java. Si vous ne voulez pas que ce message s'affiche à chaque fois que vous utilisez Eclipse, n'oubliez pas de cocher la boîte qui permet à Eclipse d'utiliser le répertoire donné en tant que l'espace de travail par défaut.

Si vous vous trouvez maintenant devant cet écran...



... alors félicitations ! Vous avez réussi à installer Eclipse correctement !

Un programme Hello World

Dans l'écran de début, vous verrez un tutoriel pour créer votre première application Hello World.

Faites clic sur ce dernier et suivez les étapes indiquées par Eclipse, qui seront, notamment :

1. S'assurer qu'on est dans le bon éditeur
2. Créer un premier projet (**attention, sur ce dernier ne créez pas des modules !**)
3. Créer une première classe
4. Créer une méthode principale dans laquelle on écrit un message
5. Faire exécuter son programme

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 (voire aussi les CMs 1 et 2). 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.

Routine en début de chaque TP

Au début de chaque TP vous allez créer un nouveau projet qui s'appellera TP<numéro du TP>, sans des modules dedans.

Nous allons apprendre un peu plus tard qu'une bonne organisation du code nous donnera des meilleurs résultats. Une façon de bien organiser son code c'est d'utiliser des packages.

En utilisant la suite de commandes `File >> New >> Package`, créez un nouveau package (une shortcut utile c'est d'utiliser `Alt+Shift+N` et puis choisir le mot package). Les noms de packages par convention avec une minuscule. Par exemple, j'ai mis comme package `tp1`. Cliquez sur Finish.

Pour les TPs prochains vous allez devoir utiliser le code du TP précédent. Nous allons apprendre comment faire cela lors du TP prochain. Pour l'instant, nous devons créer un premier bout de code.

1. Créez 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)
```

2. Dans la classe `Pokemon` : ajoutez une méthode `String toString()` qui retourne le texte suivant :

```
Pokemon[<nom>, <type du pokemon>, <niveau>].
```

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

```
nom : Piplup, type : EAU, 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.

3. Créez une nouvelle classe `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.

- Dans la méthode `main`, faites afficher dans la console le texte :

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

4. Maintenant, nous allons écrire dans la classe `Pokemon` une méthode avec la signature : `void 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.
5. 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()`.
6. 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.
7. 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 `periodeJournee`. Puis, au lieu d'utiliser la variable `periodeJournee` en entrée de la méthode `direBonjour`, on utilisera le premier élément de `args[]`.

- 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.