

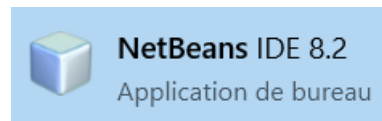
Dans ce TD vous trouverez une prise en main de l'environnement intégré Netbeans et vous réaliserez vos premiers programmes en Java.

Table des matières

1 Netbeans : environnement de développement intégré

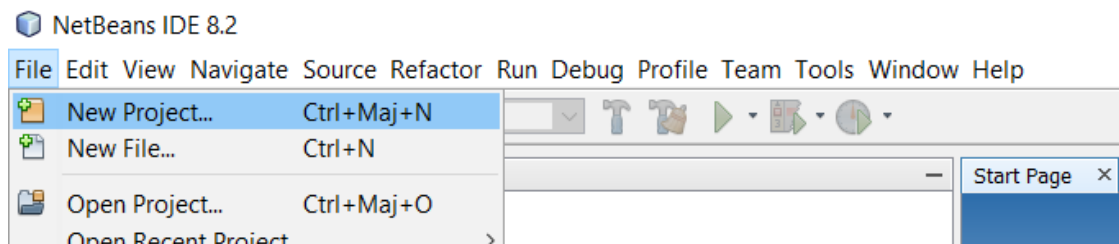
Vous allez être guidé pas à pas pour la création de votre tout premier programme Java.

1. Ouvrez NetBeans. L'icône de l'application se trouve sur votre Bureau.

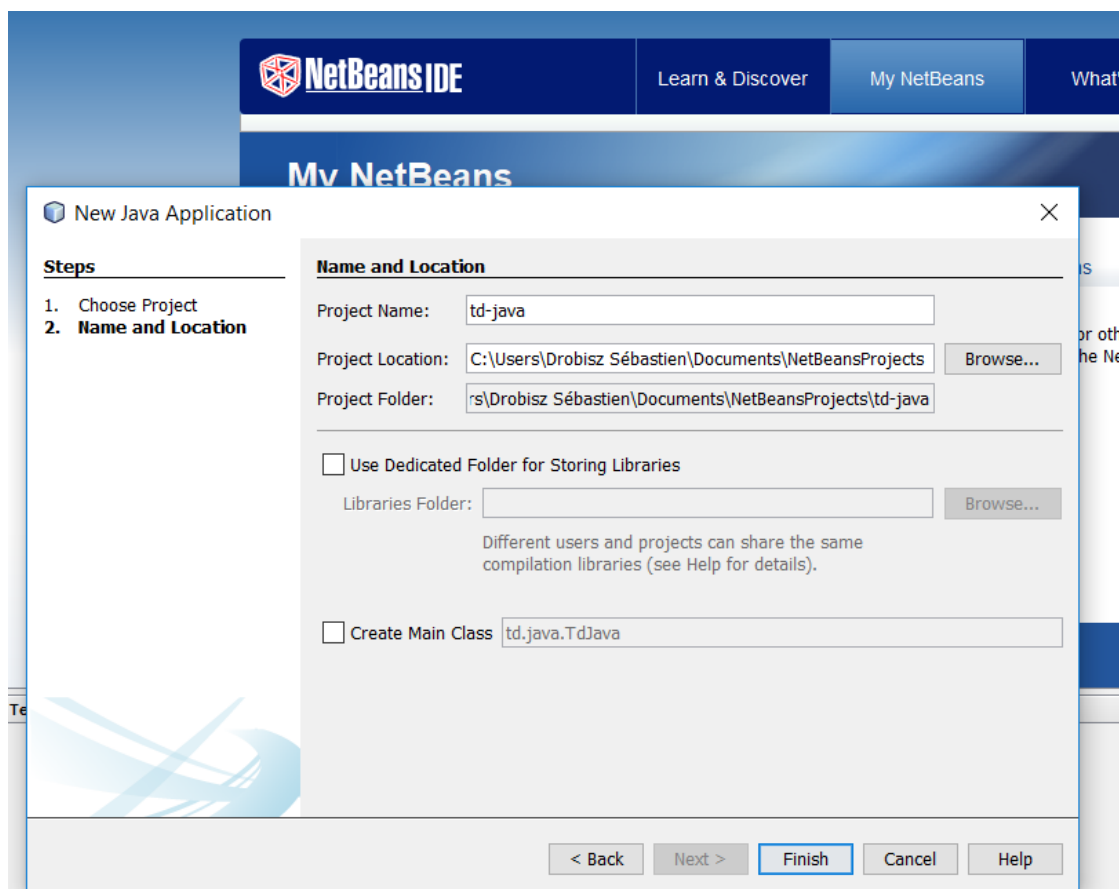


2. Créez un nouveau projet :

Pour cela, cliquez sur « File » en haut à gauche et ensuite, sur nouveau projet. Vous pouvez également utiliser le raccourci clavier CTRL + MAJ + N.



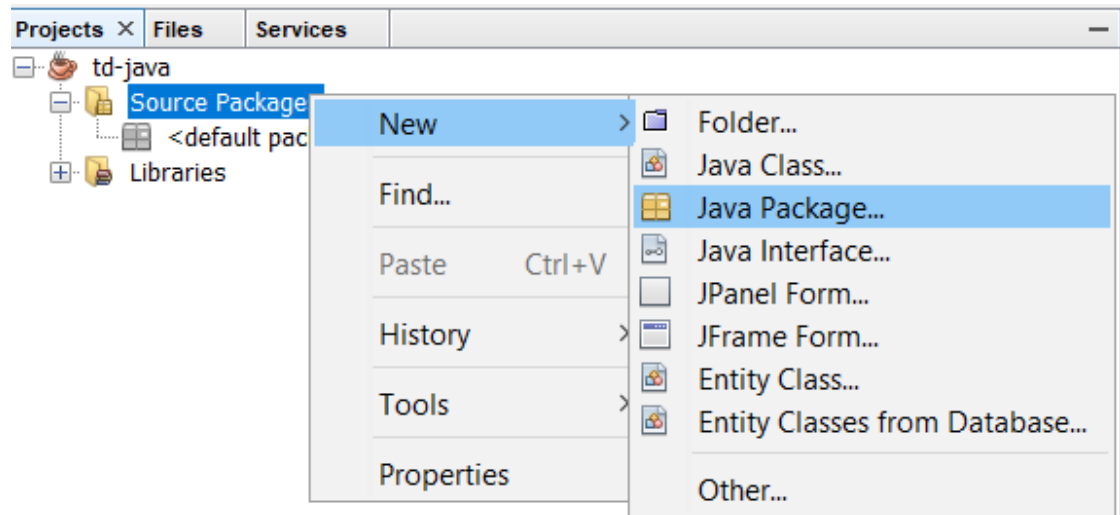
Nommez ce projet td-java et décochez la case Create Main Class.



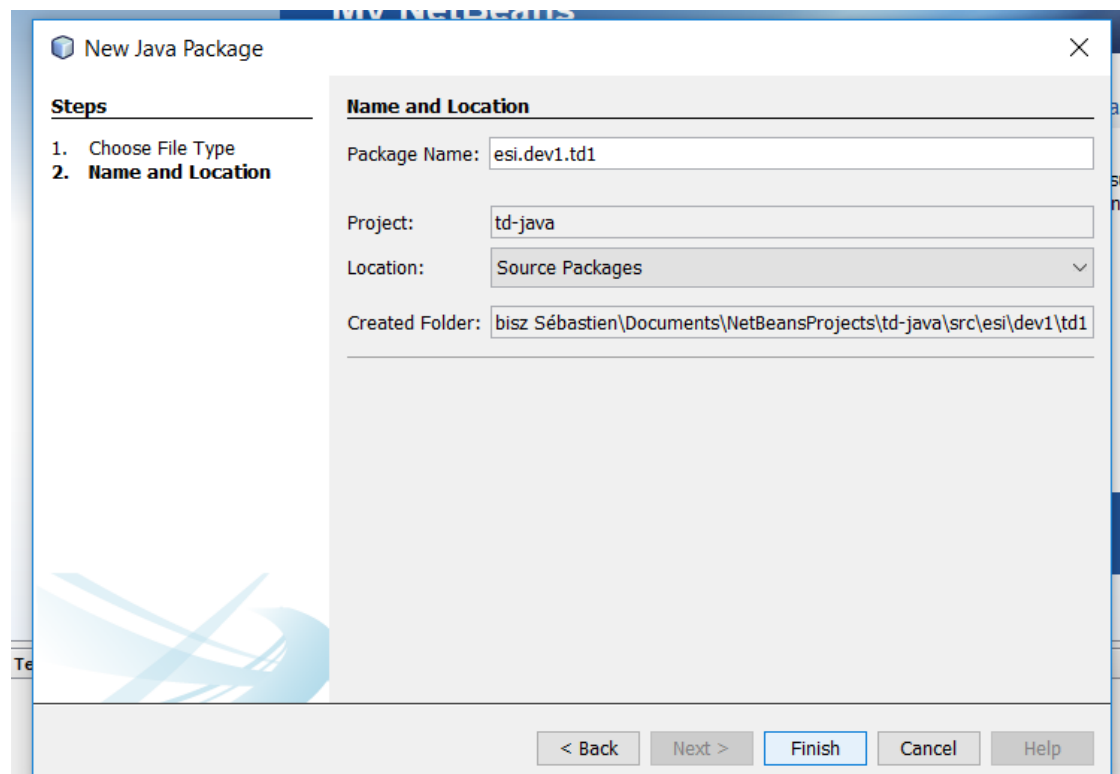
Dans Netbeans tout programme doit se trouver au sein d'un projet. Le projet contient le code de votre programme mais également les informations annexes comme le langage utilisé (ici Java), la version du langage (ici nous utilisons Java 8), et d'autres informations que vous découvrirez au fur et à mesure.

3. Créez un package :

Faites un clic droit sur le dossier contenant les sources de votre projet et ajoutez un nouveau package Java.

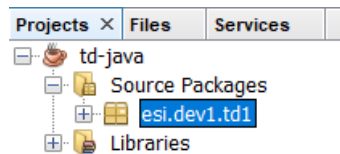


Nommez ce package g12345.dev1.td1 où vous remplacez g12345 par votre identifiant :

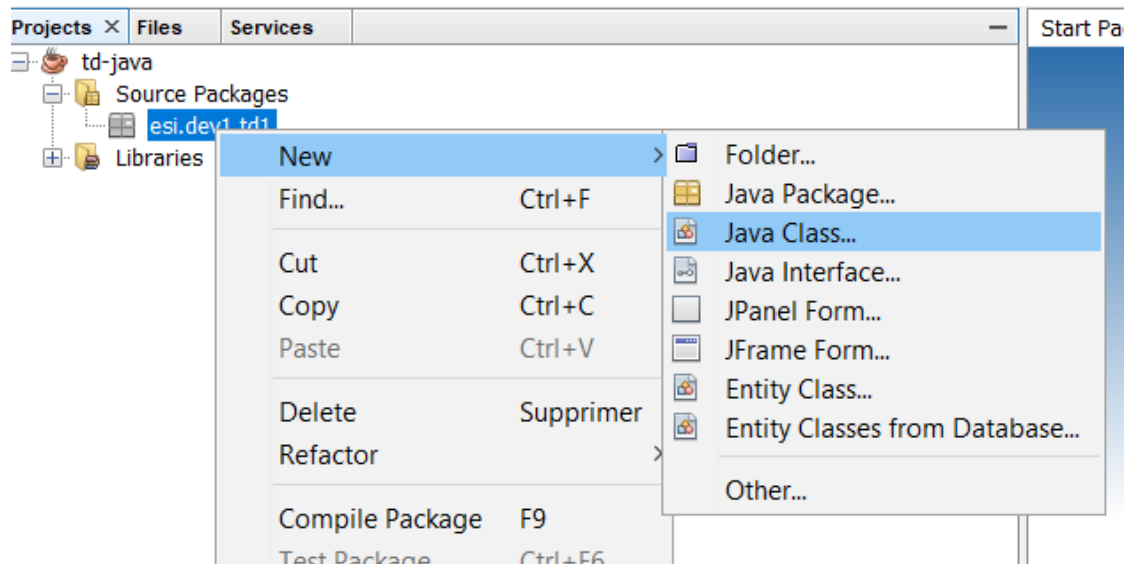


En Java, un package (paquet) permet de regrouper certains morceaux de votre code et ainsi d'ordonner votre projet.

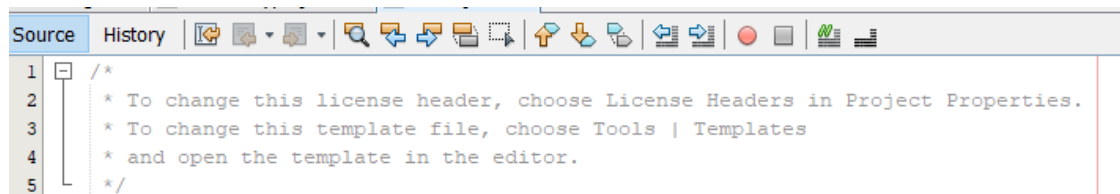
Remarquez que ce package a été ajouté dans les sources de votre projet.



4. Créez une classe : Dans NetBeans, faites un clic droit sur votre package et ajoutez une nouvelle classe ¹. Nommez cette classe Hello ²



5. Cliquez sur votre classe Hello. Si vous voyez le code suivant, veuillez à l'effacer ³.




Faites la même chose pour le code contenant `/**...@author...*/`. Maintenant que c'est fait, ajoutez le code suivant :

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Vous devriez obtenir ceci :

```
1 package esi.dev1.td1;
2
3 public class Hello {
4     public static void main(String[] args) {
5         System.out.println("Hello, world!");
6     }
7 }
```

6. Lancez le programme en cliquant sur la petite flèche *run* .
7. Modifiez votre programme pour que celui-ci affiche "Hello" suivi de votre prénom.

-
1. Pour le moment, on va simplement dire que c'est un fichier dans lequel se trouve du code Java.
 2. Si vous oubliez la majuscule au H, nous viendrons vous tirer les oreilles!
 3. Nous n'aimons pas le voir, il ne sert à rien

2 Affichage

Le programme suivant affiche Hello! suivi de Bonjour!.

```
1 package esi.dev1.td1;
2
3 public class HelloBonjour {
4
5     public static void main(String[] args) {
6         System.out.println("Hello!");
7         System.out.println("Bonjour!");
8     }
9 }
```

[code/HelloBonjour.java](#)

En java l’affichage se fait par l’*instruction* : `System.out.println("Hello!");`

Le texte entre guillemets sera affiché sur la *sortie standard*.

Exercice 1

Ligne

Dans votre package `g12345.dev1.td1` créez une classe `Ligne`. Dans cette classe écrivez un programme (et donc dans la fonction `main` de cette classe) qui affiche 10 tirets les uns à la suite des autres :

- - - - -

Exercice 2

Carré

Dans une classe `Carré`, écrivez un programme qui affiche un carré d’étoiles de 5 de côté :

```
*****
*****
*****
*****
*****
```

Exercice 3

Pyramide

Dans une classe `Pyramide`, écrivez un programme qui affiche une pyramide d’étoiles comme ceci :

```
  *
 ***
*****
*****
```

Exercice 4

Triangle

Dans une classe `Triangle`, écrivez un programme qui affiche un triangle d’étoiles comme ceci :

```
 *
 **
 ***
****
***
 **
 *
```

3 Expressions

Le programme suivant affiche la somme de 12345678 et 87654321.

```
1 package esi.dev1.td1;
2
3 public class Expression {
4
5     public static void main(String[] args) {
6         System.out.println("12345678+87654321 = ");
7         System.out.println(12345678+87654321);
8     }
9 }
```

[code/Expression.java](#)

- ▷ L'instruction de la ligne 6 affiche le texte entre guillemets : 12345678+87654321 =
- ▷ L'instruction de la ligne 7 affiche le *résultat du calcul* c'est-à-dire la somme de 12345678 et de 87654321 : 99999999
- ▷ Notez que l'*expression* 12345678+87654321 de l'instruction de la ligne 7 ne se trouve *pas* entre guillemets.

Exercice 5

Dans une classe **Calculs**, écrivez un programme qui affiche la somme la valeur des expressions suivantes :

- ▷ 10 + 32
- ▷ 2 * 21
- ▷ 234%57
- ▷ ((2*2)+(3*3))/25

4 Variables

Le programme suivant affiche l'aire d'un rectangle de longueur 12 et de largeur 4.

```
1 package esi.dev1.td1;
2
3 public class Variables {
4
5     public static void main(String[] args) {
6         int longueur = 12;
7         int largeur = 4;
8         System.out.println("L'aire du rectangle est : ");
9         System.out.println(longueur*largeur);
10    }
11 }
```

[code/Variables.java](#)

À la ligne 6 la variable `longueur` est *déclarée* avec le type `int`, elle peut donc 'contenir' des entiers. Sur cette même ligne on lui *assigne* la valeur 12.

À la ligne 7 la variable `longueur` est *déclarée* avec le type `int`. Sur cette même ligne on lui *assigne* la valeur 4.

À la ligne 9 on affiche le résultat de la multiplication de la valeur de la variable `longueur`, qui vaut 12, et de la valeur de la variable `largeur`, qui elle vaut 4.

Exercice 6

Dans un classe `VariablesExercice1` déclarez 2 variables : `x`, `y` et initialisez-les avec les valeurs 51 et 17.

Ensuite affichez sur la sortie standard la valeur de :

- ▷ `x+y`
- ▷ `x-y`
- ▷ `x*y`
- ▷ `x/y`
- ▷ `x%y`
- ▷ `x*x+y*y`

Exercice 7

Dans un classe `VariablesExercice2` déclarez 3 variables : `a`, `b` et `c` et initialisez avec les valeurs 2, 3 et 4 respectivement.

Ensuite affichez sur la sortie standard la valeur de :

- ▷ `4 * a * c`
- ▷ `b*b - 4*a*c`

5 Lecture au clavier

En java la lecture au clavier se fait en 3 étapes.

1. *Importer* le *lecteur* (`Scanner`) ligne 3 du code ci-dessous.
2. *Déclarer* et *initialiser* le lecteur : `Scanner clavier = new Scanner(System.in);`
3. La lecture proprement dite : `int longueur = clavier.nextInt();`

```
1 package esi.dev1.td1;
2
3 import java.util.Scanner;
4
5 public class AireRectangle {
6     public static void main(String[] args) {
7         Scanner clavier = new Scanner(System.in);
8         System.out.println("Entrez la longueur du rectangle: ");
9         int longueur = clavier.nextInt();
10
11         System.out.println("Entrez la largeur du rectangle: ");
12         int largeur = clavier.nextInt();
13
14         System.out.println("L'aire du rectangle vaut: ");
15         System.out.println(longueur*longueur);
16     }
17 }
```

[code/AireRectangle.java](#)

Exercice 8

Aire d'un carré

Dans une classe `AireRectangle`, écrivez un programme qui demande le côté d'un carré à l'utilisateur et affiche l'aire de ce carré.

Exercice 9

Dans une classe `LectureExercice1`, écrivez un programme qui demande deux nombres entiers, `x` et `y`, à l'utilisateur et afficher la valeur de :

- ▷ $x+y$
- ▷ $x-y$
- ▷ $x*y$
- ▷ x/y
- ▷ $x*x+y*y$

6 Exercices

Exercice 10

Aire d'un triangle

Dans une classe `AireTriangle`, écrivez un programme qui demande la base et la hauteur d'un triangle rectangle à l'utilisateur et affiche son aire.

Rappel : l'aire d'un triangle se calcule par la formule $(\text{base} * \text{hauteur}) / 2$

Exercice 11

HH:MM:SS en secondes

Dans une classe `Hms2s`, écrivez un programme qui demande une nombre d'heures, un nombre de minutes et un nombre de secondes et qui affiche le nombre de secondes totales.

Par exemple : si l'utilisateur entre 2 heures, 10 minutes et 27 secondes, le programme affiche 7827. En effet 2 heures donnent 7200 secondes, 10 minutes sont 600 secondes auxquelles il faut ajouter les 27 secondes : $7200 + 600 + 27 = 7827$.

Exercice 12

Secondes en minutes

Exercice 13

Secondes en heures

Exercice 14

Dans une classe **Felix**, écrivez un programme qui affiche le dessin suivant :

[illegible]