

DEV1 – Laboratoires Java I**TD 1 – NetBeans**

Dans ce TD vous ferez connaissance avec l'environnement intégré NetBeans et vous réaliserez vos premiers programmes en Java.

Les codes sources de ce TD se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/tree/master/labo-java/td01-introduction/code/>

Les solutions se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/tree/master/labo-java/td01-introduction/solutions/>

Table des matières

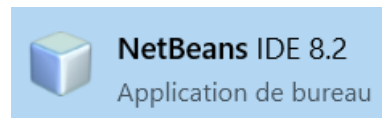
1	NetBeans : environnement de développement intégré	2
2	Affichage	6
3	Expressions	7
4	Variables	8
5	Lecture au clavier	9
6	Exercices Récapitulatifs	9

1 NetBeans : environnement de développement intégré

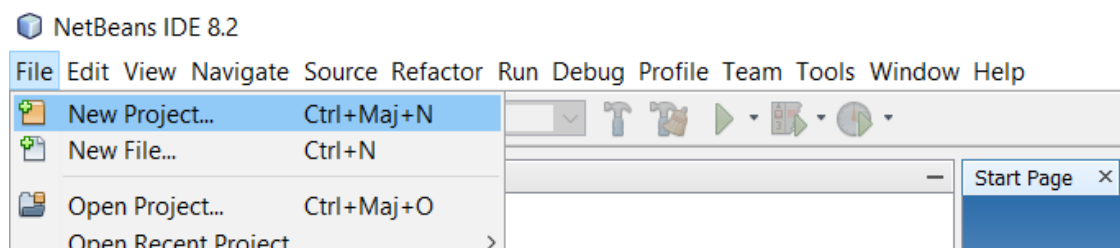
Tutoriel 1 Premier programme Java avec NetBeans

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

- ✍ Ouvrez NetBeans : l'icône de l'application se trouve sur votre Bureau.

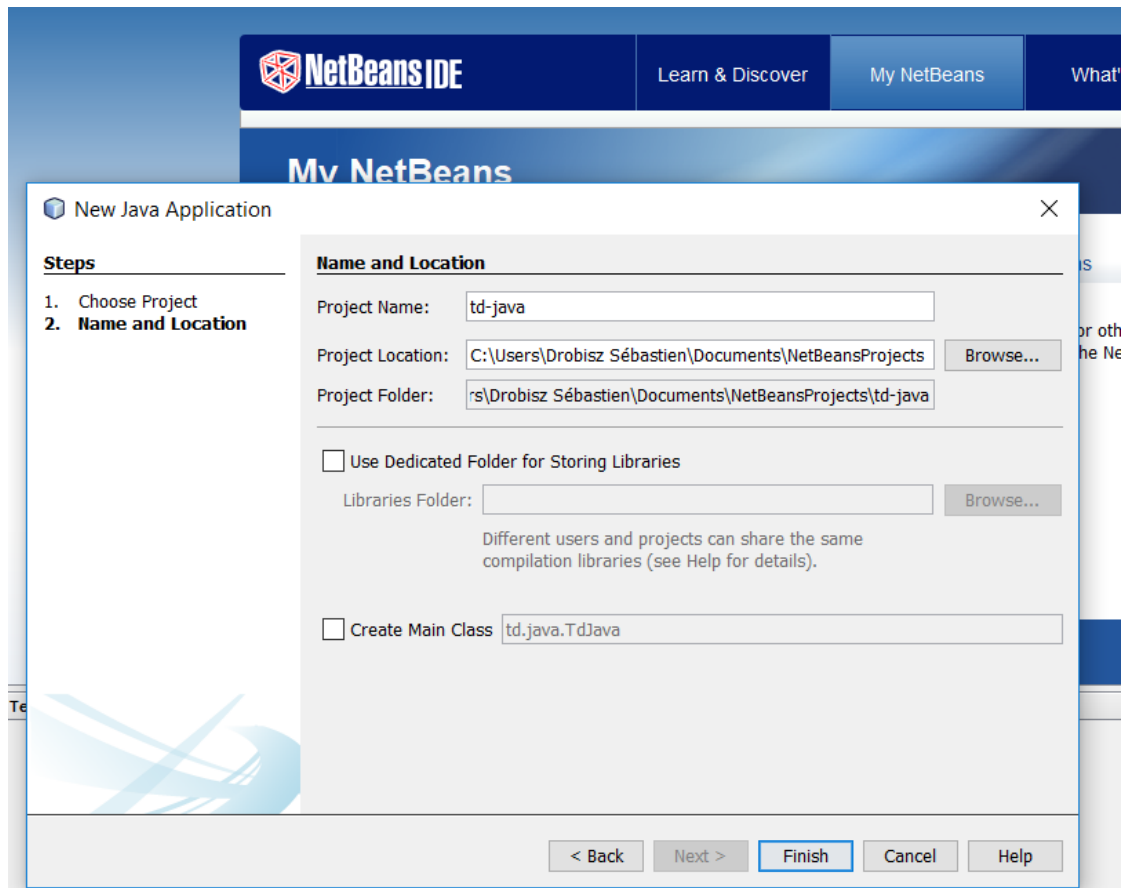


- ✍ Créez un nouveau projet :
Pour cela, cliquez sur « File » en haut à gauche et ensuite, sur nouveau projet.



Dans la fenêtre « New Project » qui s'ouvre, choisissez « Java Application » dans la liste « Projects » et cliquez le bouton « Next ».

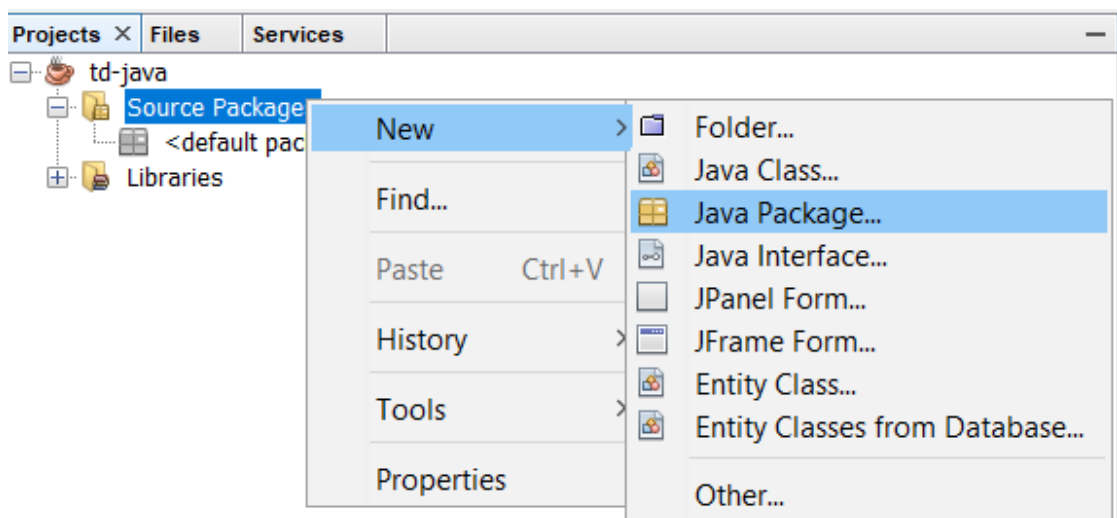
A l'écran suivant « New Java Application » illustré ci-dessous, nommez le projet `td-java`, décochez la case `Create Main Class` et cliquez sur le bouton « Finish ».



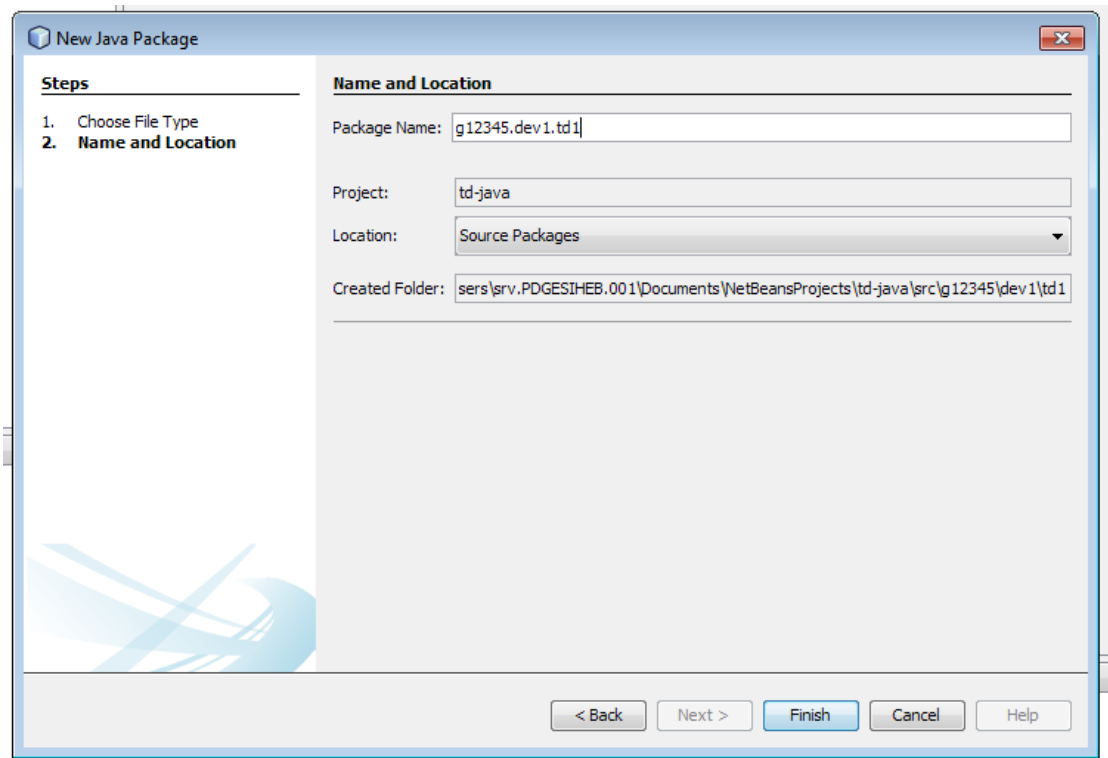
Dans NetBeans tout programme doit se trouver au sein d'un projet. Le projet contient le code de votre programme mais également les informations annexes comme le langage utilisé (ici Java), la version du langage (ici nous utilisons Java 8), et d'autres informations que vous découvrirez au fur et à mesure.

✍ Créez un package :

faites un clic droit sur le dossier contenant les sources de votre projet, comme illustré dans l'image ci-dessous, et ajoutez un nouveau package Java.

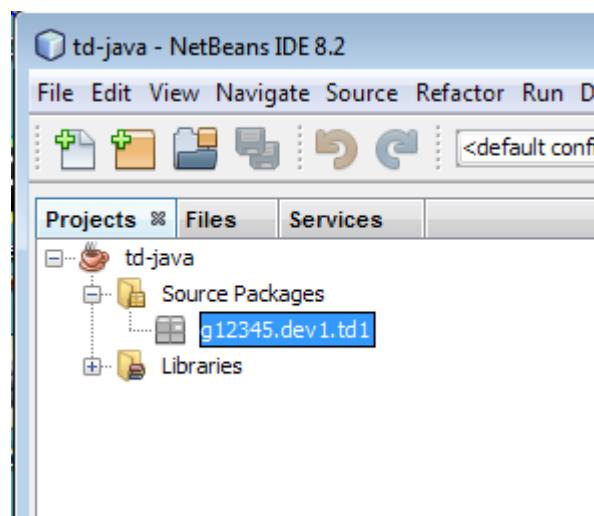


Nommez ce package `g12345.dev1.td1` où vous remplacez `g12345` par votre identifiant et cliquer sur le bouton « Finish » :



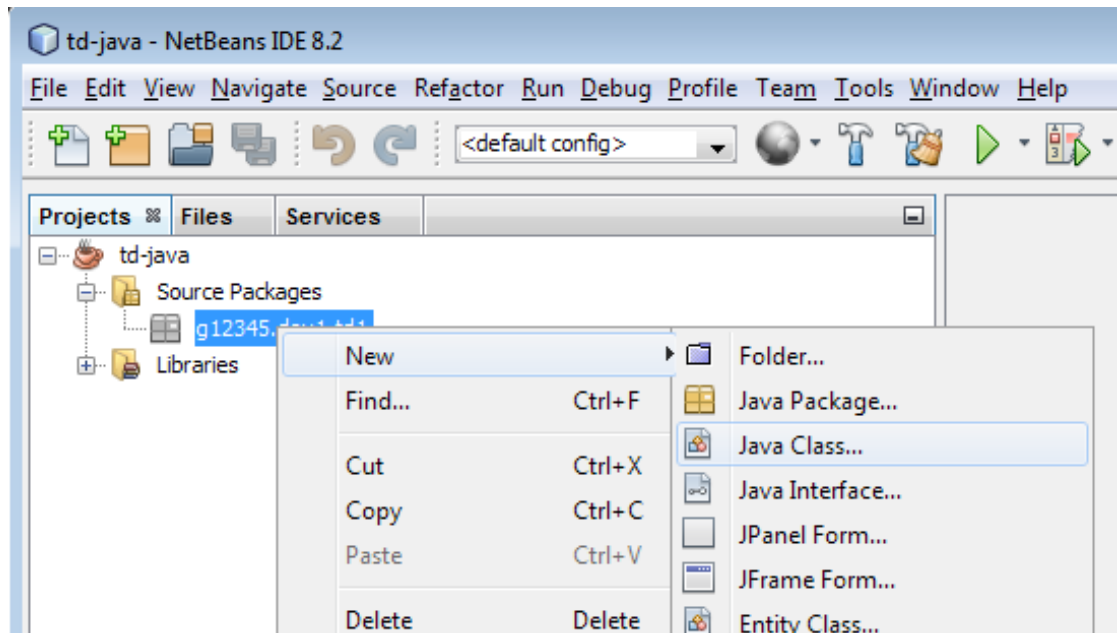
En Java, un package (paquet) permet de regrouper certaines parties de votre code et ainsi d'ordonner votre projet.

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



- ✍ Créez une classe :
comme illustré sur l'image ci-dessous faites un clic droit sur votre package et ajoutez une nouvelle classe ¹. Nommez cette classe Hello

1. Pour le moment, on va simplement dire que c'est un fichier dans lequel se trouve du code Java.



- ✎ Ouvrez le fichier `Hello.java` :
double-cliquez sur votre classe `Hello`. Le code se trouvant dans ce fichier apparaît.
Ajoutez le code suivant en respectant bien les minuscules et les majuscules :

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

Vous devriez obtenir ceci :

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

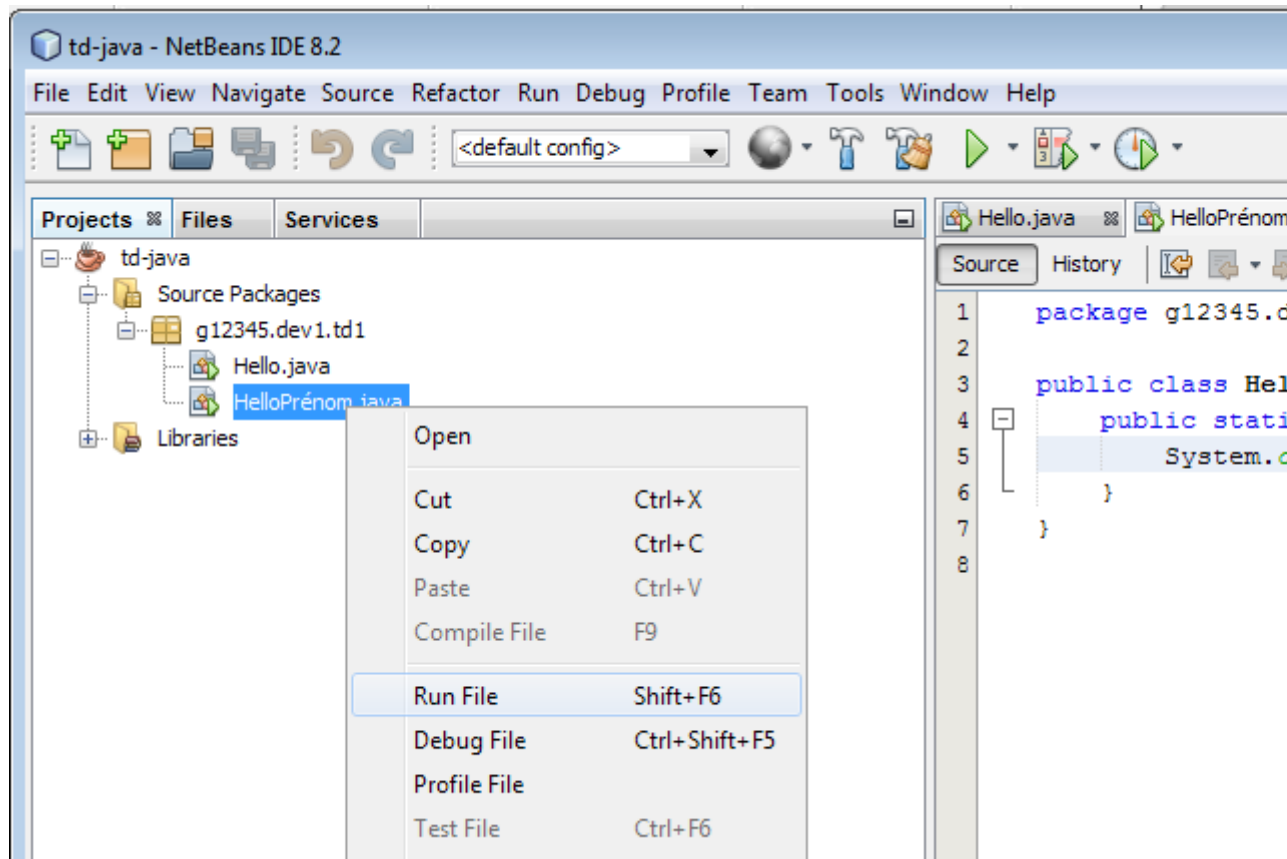
Enregistrez vos modifications (control-s).

- ✎ Lancez le programme en cliquant sur la petite flèche *run*



Le résultat s'affiche dans la fenêtre intitulée « Output ». Vous devriez voir "Hello, World!" s'y afficher.

- ✎ Créez un nouveau programme qui se nomme "`HelloPrénom`". Celui-ci doit afficher "`Hello`" suivi de votre prénom. Si vous essayez de le faire tourner en cliquant sur la flèche, vous ne verrez pas apparaître votre prénom. C'est parce que Netbeans a retenu que le programme principale est le premier à avoir été lancé. Pour forcer Netbeans à lancer le nouveau programme, faite un clique droit sur le fichier `HelloPrénom.java` et cliquez sur `Run File`. Vous devriez voir "Hello, John" s'afficher dans l'output.



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 }

```

HelloBonjour.java

- ▷ La première ligne indique que la classe se trouve dans le package `esi.dev1.td1`
- ▷ La ligne 3 déclare la classe `HelloBonjour`, remarquez que le nom d'une classe doit correspondre au nom du fichier dans lequel elle se trouve, ici `HelloBonjour.java`
- ▷ La ligne 5 déclare la *méthode principale*, *main* en anglais veut dire 'principale'. C'est ici que commence votre programme.
- ▷ En java l'affichage se fait par l'*instruction* : `System.out.println("Hello!");`
- ▷ Le texte entre guillemets sera affiché sur la *sortie standard*.

Le code source de la classe `HelloBonjour` (ainsi que tous les codes de ce TD) se trouvent à l'adresse :

<https://git.esi-bru.be/dev1/tree/master/lab0-java/td01-introduction/code/>

N'hésitez pas à tester les codes présentées dans les TDs sur votre machine.

Exercice 1 Ligne

Dans votre package `g12345.dev1.td1` créez une classe `Exercice1`. Dans cette classe écrivez un programme (et donc dans la méthode `main` de cette classe) qui affiche 10 tirets les uns à la suite des autres :

```
- - - - -
```

Exercice 2 Carré

Dans une classe `Exercice2` (dans votre package `g12345.dev1.td1`), écrivez un programme qui affiche un carré d'étoiles de 5 de côté :

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

Exercice 3 Pyramide

Dans une classe `Exercice3`, écrivez un programme qui affiche une pyramide 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 }
```

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 4 Petits calculs

Dans une classe `Exercice4`, écrivez un programme qui affiche la valeur des expressions suivantes :

- ▷ `10 + 32`

- ▷ 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 }
```

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 `largeur` 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 5 Petits calculs avec 2 variables

Dans un classe `Exercice5` déclarez 2 variables : `a`, `b` et initialisez-les avec les valeurs 51 et 17.

Ensuite affichez sur la sortie standard la valeur de :

- ▷ `a+b`
- ▷ `a-b`
- ▷ `a*b`
- ▷ `a/b`
- ▷ `a%b`
- ▷ `a*a+b*b`

Exercice 6 Petits calculs avec 3 variables

Dans un classe `Exercice6` 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*largeur);
16     }
17 }
```

AireRectangle.java

Exercice 7 Aire d'un carré

Écrivez un programme qui demande le côté d'un carré à l'utilisateur et affiche l'aire de ce carré.

Exercice 8 Petits calculs avec 2 nombres lus au clavier

Écrivez un programme qui demande deux nombres entiers, **a** et **b**, à l'utilisateur et affiche la valeur de :

- ▷ $a+b$
- ▷ $a-b$
- ▷ $a*b$
- ▷ a/b
- ▷ $a\%b$
- ▷ $a*a+b*b$

6 Exercices Récapitulatifs

Exercice 9 Centaines, dizaines, unités

Écrivez un programme qui demande à l'utilisateur un nombre **nb** compris entre 0 et 1000 et affiche la valeur des expressions suivantes :

- ▷ $nb\%10$ - les unités
- ▷ $(nb/10)\%10$ - les dizaines
- ▷ $(nb/100)\%10$ - les centaines

Par exemple, si l'utilisateur entre 362 votre programme affiche

2
6
3

Exercice 10 Miroir

Écrivez un programme qui demande à l'utilisateur un nombre `nb` compris entre 1000 et 9999 et affiche la valeur miroir :

Par exemple, si l'utilisateur entre 7362 votre programme affiche 2637.

Exercice 11 Secondes en minutes

Écrivez un programme qui demande un nombre de secondes à l'utilisateur et qui affiche le nombre de minutes que cela représente.

Par exemple : si l'utilisateur entre 217 secondes, le programme affiche 3, car 217 secondes correspondent à 3 minutes et 37 secondes.

Exercice 12 Temps en secondes

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 : $7200 + 600 + 27 = 7827$.

Exercise 13

Écrivez un programme qui affiche le dessin suivant :

[illegible]