

# II.1102 / II.1202 TP1: Découverte de l'environnement de programmation

Sylvain Lefebvre

*<2015-09-23 mer.>*

## 1 Objectifs du TP

1. se familiariser avec l'environnement de développement
2. se familiariser avec la syntaxe du langage Java
3. appliquer la méthode vue en cours

## 2 L'environnement de programmation: l'IDE Eclipse

Dans un premier temps nous vérifier et configurer que l'environnement de développement nécessaire à la programmation Java est bien présent sur vos machines.

### 2.1 Sur les machines de l'ISEP

- Tout est déjà là ! Vous pouvez passer à l'étape lancement d'eclipse.

### 2.2 Sur vos machines personnelles

#### 2.2.1 Vérifier la présence de Java

- ouvrir un terminal / rechercher et taper `javac + ENTREE`
- Si une erreur apparaît c'est que kit de développement Java n'est pas installé. Il est disponible à l'adresse suivante: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

#### 2.2.2 Télécharger et installer Eclipse

<https://www.eclipse.org/downloads/>

- lancer l'installateur et sélectionner "eclipse IDE for Java developers"

## 3 Lancement d'Eclipse

### 3.1 Création d'un répertoire de travail

### 3.2 Création d'un projet

1. sur l'écran d'accueil d'eclipse, sélectionner "Go to workbench"

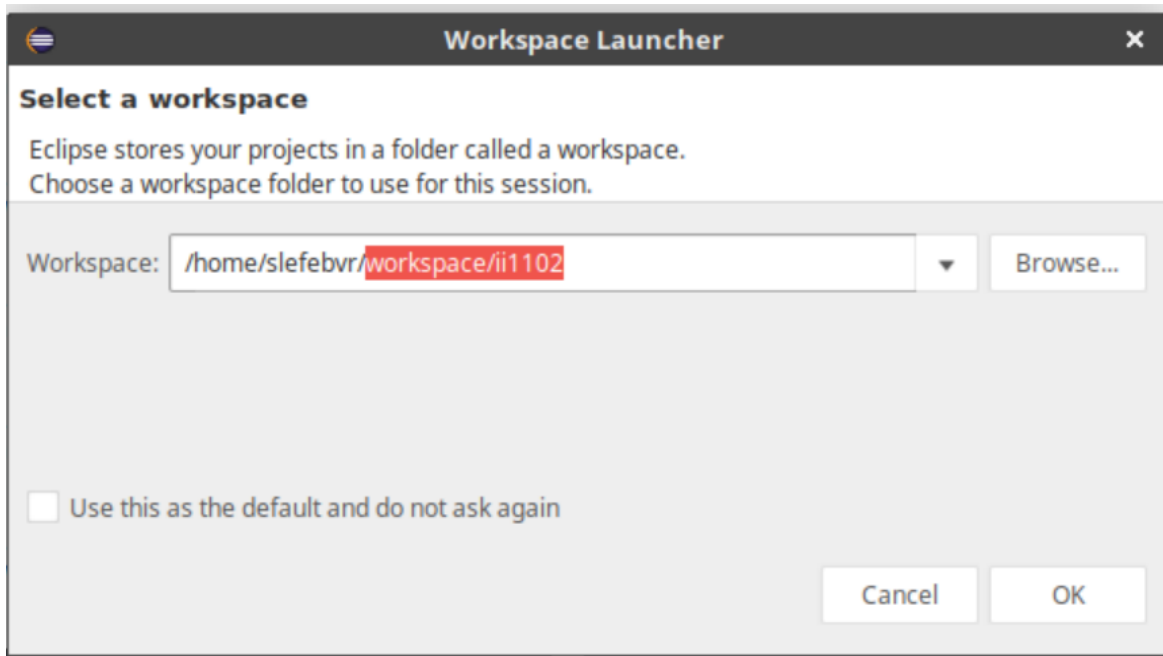


Figure 1: Choix du répertoire de travail

2. Puis aller dans le menu "File / Fichier" et sélectionner **New / Nouveau** → **Java Project**
3. Nommer le projet "tp1-vos-noms"
4. Vérifier grâce à l'explorateur de fichier que le projet s'est bien créé

## 4 Premiers programmes

Dans cette partie du TP, nous allons rédiger et analyser un premier programme JAVA.

### 4.1 Bonjour le monde

#### 4.1.1 Classe principale

Java est un langage **orienté objet**. La conséquence de cette approche est que les programmes Java sont organisés sous forme de **classes** (cf. Vocabulaire) et chacune de ces classes doit être décrite dans un fichier différent, portant le nom de la classe suivi de l'extension `.java`. Afin de commencer l'exécution d'un programme, le système doit donc déterminer quelle classe utiliser et quelle portion de code exécuter en premier.

L'utilisateur d'un programme Java doit donc spécifier lors du lancement de son programme quelle classe doit être lue par le système. Le système va chercher une **méthode** particulière dans cette classe: la méthode `static void main(String [] args)` (les détails de cette syntaxe seront vus lors de la prochaine séance de cours). Cette méthode est donc **nécessaire** et doit être présente au moins une fois dans l'ensemble du code pour pouvoir exécuter le programme.

Dans la suite, nous allons voir comment créer une première classe ainsi que sa méthode principale.

### 4.1.2 Création d'un premier programme

A partir de votre projet Eclipse, suivez le processus de création d'une classe principale décrit dans la vidéo ICI. Au terme de cette vidéo, le code suivant est inscrit dans le fichier `Hello.java`. (Vous pouvez le recopier d'ici, si nécessaire)

```
1 public class Hello {  
2  
3     public static void main(String[] args) {  
4  
5         System.out.println("Hello le monde");  
6  
7     }  
8 }
```

Ce programme très simple n'a qu'un seul objectif: afficher la chaîne de caractère "Hello le monde" sur la sortie standard du système, il ne prend aucun paramètre en entrée.

### 4.1.3 Execution du programme

Exécuter le programme à l'aide de l'environnement eclipse : vidéo. Le résultat du programme s'affiche dans l'onglet "console" en bas de la fenêtre. L'environnement de programmation a géré toute la chaîne allant de la compilation à l'exécution du programme. Nous allons voir chaque étape de cette chaîne individuellement.

#### 1. Compilation & exécution manuelles du programme

- (a) Utiliser l'explorateur de fichiers du système pour vous rendre dans le répertoire de votre projet
- (b) On observe la présence de 2 répertoires différents: `bin/` et `src/`.
  - Le répertoire `src/` contient le fichier `Hello.java` tel qu'il a été édité via l'environnement Eclipse.
  - Le répertoire `bin/` contient un fichier `Hello.class`, illisible par un éditeur de texte. Il s'agit de la classe compilée par le compilateur Java.

*Supprimer le fichier `bin/Hello.class`, et essayer d'exécuter le programme à nouveau, que constatez-vous ?*

- (a) Lancer une invite de commande dans le répertoire `ii1102-tp1/` (clic droit -> ouvrir un terminal ici)
- (b) Lancer la commande suivante:

```
$ javac src/Hello.java -d bin/
```

L'outil "javac" est le JAVA Compiler, cette commande va **compiler** le fichier `.java` et copier le résultat de cette compilation dans le répertoire `bin/`. Le fichier `.class` est de nouveau présent. Tous les programmes java doivent passer par cette étape pour pouvoir être exécutés.

La commande `java` permet l'exécution du code compilé. Il s'utilise comme suit:

```
$ java -cp bin/ Hello  
Hello le monde
```

## 4.2 Interaction avec l'utilisateur

Le langage Java possède un ensemble d'outils pour faciliter la tâche du développeur. Afin de permettre des entrées au clavier par l'utilisateur, il faut créer, puis utiliser un **objet** de type Scanner.

Le code ci-dessous montre comment déclarer un objet de type Scanner:

```
1 import java.util.Scanner;
2
3 public class TestScanner {
4     public void main(String [] args) {
5         /**
6          * Declaration d'un scanner pour lire des donnees depuis la console
7          */
8         Scanner scan = new Scanner(System.in);
9         long entierLu = scan.nextLong(); // Lecture d'un entier long
10        double doubleLu = scan.nextDouble(); // Lecture d'un flottant en double precision
11        String test = scan.nextLine(); // Lecture d'une chaine de caractères
12
13        // System.out.println permet d'AFFICHER des variables a l'ecran
14        System.out.println(" J'ai lu :" + entierLu + " et " + doubleLu + " et " + test + " !");
15    }
16 }
```

### 4.2.1 Modifier votre programme Hello.java pour demander et afficher le nom de l'utilisateur

- attention à bien ajouter la ligne "import java.util.Scanner;" au début du programme !

## 5 Premier algorithme

### 5.1 Calculer le volume d'un pavé droit

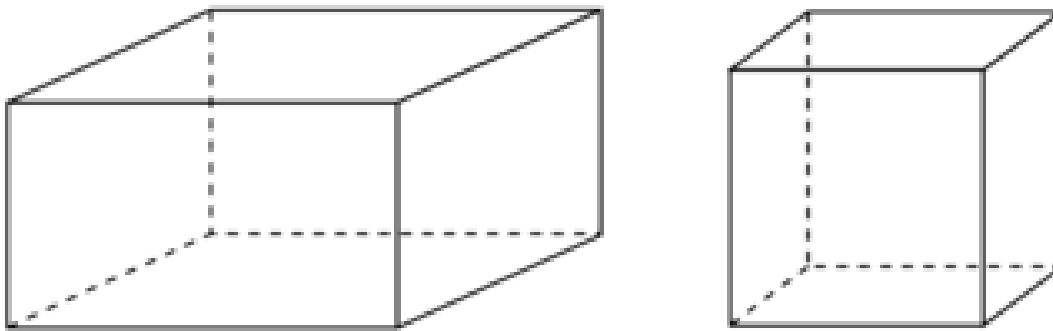


Figure 2: Exemples de pavés droits

On souhaite écrire un programme permettant de calculer le volume d'un pavé droit (exemple en Figure 3). Suivez les étapes décrites en cours pour mettre au point ce programme.

#### 5.1.1 Etape 1: Délimitation du problème

1. De quelle(s) valeur(s) a-t-on besoin pour faire ce calcul ?
2. Comment les obtenons-nous ?

- (a) On ne sait pas
- (b) D'un ensemble de variables
- (c) Du clavier
- (d) D'un fichier
- (e) Du réseau

3. Que doit-on faire du résultat ?

- (a) Le renvoyer au code appelant
- (b) L'afficher
- (c) l'écrire dans un fichier
- (d) On ne sait pas

4. Quelle est la formule à utiliser pour ce calcul ?

#### **5.1.2 Etape 2: Ecrire des exemples de calculs**

- que faire si des valeurs sont négatives ?
- quel type de variable utiliser pour ces valeurs ?

#### **5.1.3 Etape 3: Ecrire les étapes nécessaire au calcul**

#### **5.1.4 Etape 4: Ecrire le code java nécessaire à ce calcul.**

- Créez une nouvelle classe avec une méthode main dans le même projet. Aidez vous des exemples vus dans le cours pour écrire ce code.

#### **5.1.5 Etape 5: Exécuter le code ainsi créé**

#### **5.1.6 Etape 6: Repérer les erreurs éventuelles**

- les corriger

#### **5.1.7 Etape 7: Documenter le code**

### **5.2 Interaction avec l'utilisateur**

Modifier le programme de la question précédente pour demander les valeurs des côtés du pavé à l'utilisateur. Aidez vous des exemples vus en cours pour cela.

## **6 Résolution d'équation du 2nd degré**

### **6.1 Proposer un algorithme permettant de résoudre une équation du second degré dans l'ensemble des réels**

- Commencer par initialiser les variables nécessaire directement dans le programme.
- Suivez la méthode donnée en cours pour résoudre ce problème.

## 6.2 Modifier votre programme pour demander les variables nécessaires à l'utilisateur.

# 7 Vocabulaire

## 7.1 Classe

En programmation orientée objet, une classe regroupe des membres, méthodes et propriétés (attributs) communs à un ensemble d'objets. La classe déclare (décrit), d'une part, des attributs représentant l'état des objets et, d'autre part, des méthodes représentant leur comportement. Une classe représente donc une catégorie d'objets.

## 7.2 Objet

Un objet est une **instance** de classe. Un objet contient des attributs (tels que décrits par sa classe) auxquels des valeurs ont été affectés. Ces valeurs spécifient l'état de l'objet. Les méthodes qui peuvent être appelées sur l'objet utilisent ou modifient ces valeurs.

## 7.3 Méthode

En programmation orientée objet (POO), une méthode est une routine ou une fonction membre d'une classe. Emacs 24.5.1 (Org mode 8.3.3)