

INFO-H-200

Programmation orientée objet

Séance d'exercices 1
Introduction au langage JAVA

Université libre de Bruxelles
École polytechnique de Bruxelles

Professeur : Hugues Bersini

2017-2018

Organisation

Travaux pratiques :

- 9 séances de 3 heures,
- Sur papier et sur machine.

Site web :

- <http://cs.ulb.ac.be/public/teaching/infoh200>
- Enoncés, corrections et projet.

Assistants :

- Dejaegere Gilles - gdejaege@ulb.ac.be
- Rosenfeld Jean - jean.rosenfeld@ulb.ac.be
- Bonatto Daniele - Daniele.Bonatto@ulb.ac.be

Java

Language de programmation orienté objet créé par Sun Microsystems

- Langage portable entre différents OS
- Bibliothèques standards (API Java) : String, Math, File, Network, Mail,...
- Le code est compilé en un langage intermédiaire appelé "bytecode" sur une machine virtuelle (la JVM) qui est multi-plateformes.
- Tout est objet (sauf les types primitifs) !

Terminologie

- **JVM** : Java Virtual Machine
- **JRE** : Java Runtime Environment
 - Ensemble d'outils pour exécuter des programmes Java sur toutes les plateformes supportées.
 - Constitué d'une JVM et d'une bibliothèque standard à toutes les plateformes.
- **JDK** : Java Development Kit
 - JRE + des outils de développement comme le compilateur (javac -> bytecode)
- **IDE** : Integrated Development Environment

Eclipse IDE

Premier projet

L'environnement de développement

L'IDE **Eclipse** est présent sur les machines des salles infos !

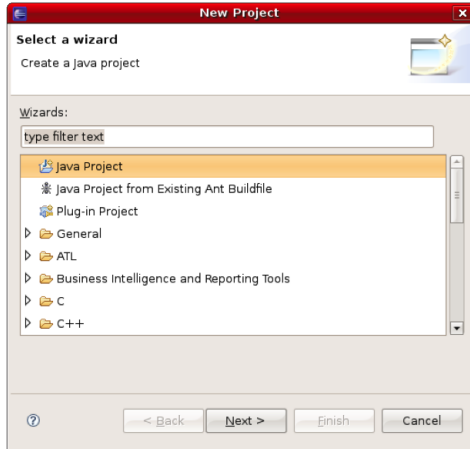
Vous pouvez aussi le télécharger et l'installer sur vos ordinateurs respectifs. Des versions Windows, Linux et OSX existent !



» [Télécharger](#) «

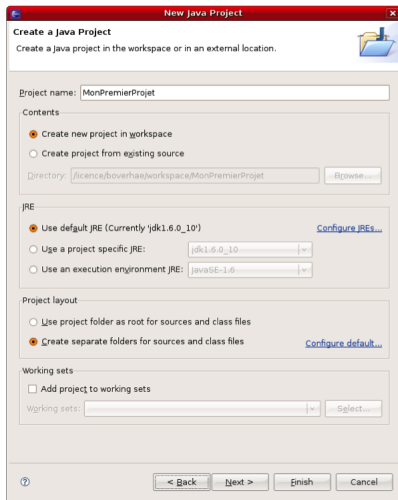
Eclipse : Créer un nouveau projet

- **File -> New -> Project -> Java Project**



Eclipse : Créer un nouveau projet

- Donner un nom au projet



The screenshot shows the 'New Java Project' dialog box in Eclipse. The title bar is red and says 'New Java Project'. The main area is titled 'Create a Java Project' and contains the instruction 'Create a Java project in the workspace or in an external location.' with a folder icon. The 'Project name' field is filled with 'MonPremierProjet'. Under 'Contents', the radio button for 'Create new project in workspace' is selected. The 'Directory' field shows the path '/science/boverhae/workspace/MonPremierProjet' with a 'Browse...' button. Under 'JRE', the radio button for 'Use default JRE (Currently jdk1.6.0_10)' is selected, with a 'Configure JREs...' link. Other options include 'Use a project specific JRE' (with a dropdown showing 'jdk1.6.0_10') and 'Use an execution environment JRE' (with a dropdown showing 'JavaSE-1.6'). Under 'Project layout', the radio button for 'Create separate folders for sources and class files' is selected, with a 'Configure default...' link. Under 'Working sets', the checkbox for 'Add project to working sets' is unchecked. The 'Working sets' field has a dropdown and a 'Select...' button. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: MonPremierProjet

Contents

☒ Create new project in workspace

☐ Create project from existing source

Directory: /science/boverhae/workspace/MonPremierProjet [Browse...](#)

JRE

☒ Use default JRE (Currently jdk1.6.0_10) [Configure JREs...](#)

☐ Use a project specific JRE: jdk1.6.0_10

☐ Use an execution environment JRE: JavaSE-1.6

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

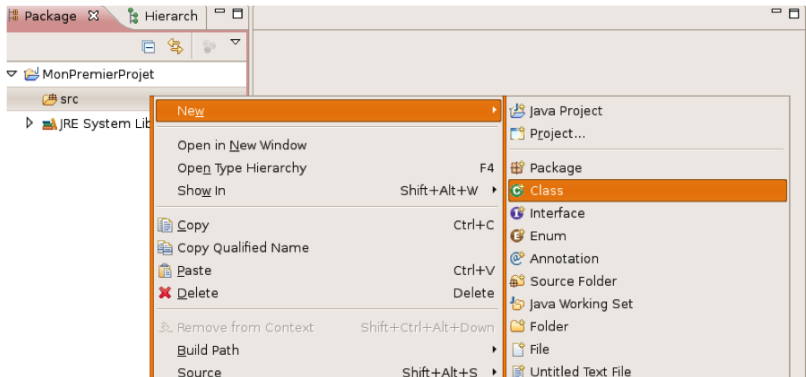
☐ Add project to working sets

Working sets: [Select...](#)

[? < Back](#) [Next >](#) [Finish](#) [Cancel](#)

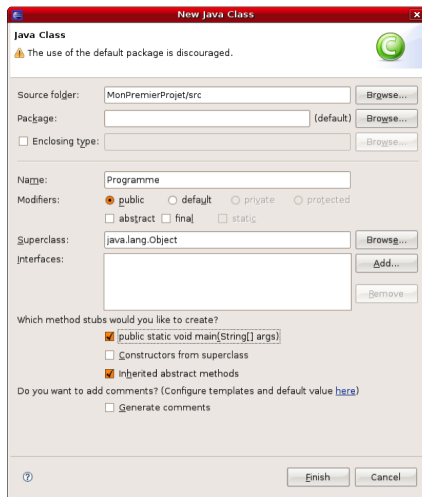
Eclipse : Nouvelle classe

- Clic droit sur le dossier 'src' -> New -> Class



Eclipse : Nouvelle classe

- Nommer la classe



The screenshot shows the 'New Java Class' dialog box in Eclipse. The title bar is red and says 'New Java Class'. The main area is light gray. At the top, it says 'Java Class' and has a warning icon with the text 'The use of the default package is discouraged.' Below this, there are three text boxes for 'Source folder:', 'Package:', and 'Enclosing type:'. The 'Source folder:' box contains 'MonPremierProjet/src'. The 'Package:' box is empty and has '(default)' next to it. The 'Enclosing type:' box is empty. To the right of each box is a 'Browse...' button. Below these is a 'Name:' text box containing 'Programme'. Under 'Name:', there are 'Modifiers:' with radio buttons for 'public' (selected), 'default', 'private', and 'protected'. There are also checkboxes for 'abstract', 'final', and 'static'. Below the modifiers is a 'Superclass:' text box containing 'java.lang.Object' and a 'Browse...' button. Below that is an 'Interfaces:' text box, an 'Add...' button, and a 'Remove' button. At the bottom, there is a section 'Which method stubs would you like to create?' with checkboxes for 'public static void main(String[] args)' (checked), 'Constructors from superclass' (unchecked), and 'Inherited abstract methods' (checked). Below this is a section 'Do you want to add comments? (Configure templates and default value [here](#))' with a checkbox for 'Generate comments' (unchecked). At the very bottom, there are 'Finish' and 'Cancel' buttons.

Java Class

⚠ The use of the default package is discouraged.

Source folder: MonPremierProjet/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: Programme

Modifiers: ☒ public ☐ default ☐ private ☐ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)

☐ Constructors from superclass

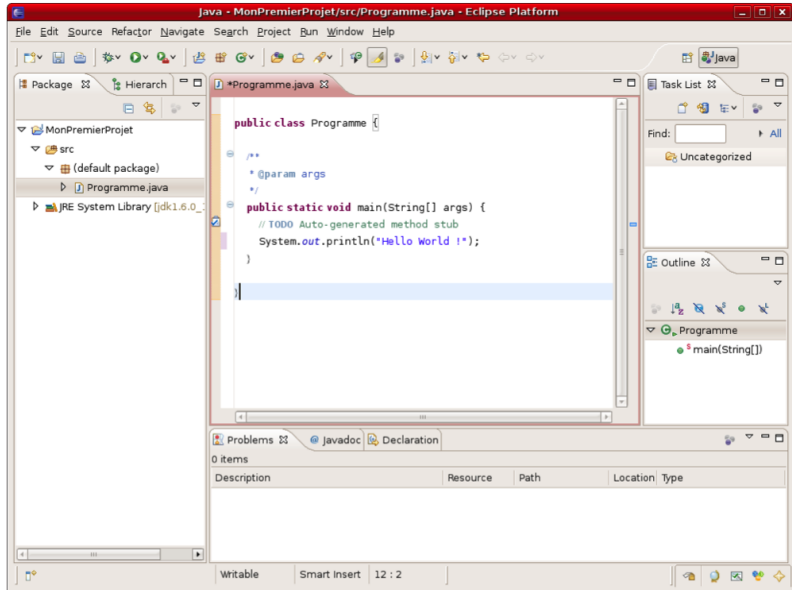
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Finish Cancel

Eclipse : Vue générale



Eclipse : Salles informatiques

Pour lancer Eclipse dans les salles informatiques, ouvrez un Terminal et tapez la commande suivante :

`/serveur/logiciels/eclipse/eclipse`

De Python à Java

Quelques notions

Premier exemple

Ecrivez un programme qui demande à l'utilisateur d'entrer le rayon d'un cercle, et qui s'en sert pour calculer l'aire et la circonférence du cercle. Pour ce faire, utilisez la valeur de π trouvée dans le module *math*.

```
import math
rayon = input("Entrez le rayon d'un cercle : ")
rayon = float(rayon)
aire = 2 * math.pi * rayon
circ = math.pi * rayon ** 2
print("L'aire est : " + str(aire))
print("La circonference est : " + str(circ))
```

```
import java.util.Scanner;
import java.lang.Math;

public class Example {
    public static void main (String args[]) {
        Scanner scanner = new Scanner( System.in );
        System.out.println("Entrez le rayon d'un cercle : ");
        String input = scanner.nextLine();
        Double rayon = Double.parseDouble(input);
        Double aire = 2 * Math.PI * rayon;
        Double circ = Math.PI * Math.pow(rayon,2);
        System.out.print("L'aire est : " + String.valueOf(aire));
        System.out.print("La circonférence est : " + String.valueOf(circ));
    }
}
```

Lancer ce premier programme Java

1 Fichier Example.java :

```
import java.util.Scanner;
import java.lang.Math;

public class Example {
    public static void main (String args[]) {
        Scanner scanner = new Scanner( System.in );
        System.out.println("Entrez le rayon d'un cercle : ");
        String input = scanner.nextLine();
        Double rayon = Double.parseDouble(input);
        Double aire = 2 * Math.PI * rayon;
        Double circ = Math.PI * Math.pow(rayon,2);
        System.out.print("L'aire est : " + String.valueOf(aire));
        System.out.print("La circonférence est : " + String.valueOf(circ));
    }
}
```

2 Compilation :

```
|      >> javac Example.java
```

3 Exécution :

```
|      >> java Example
```

ou

2 Compilation & Exécution



Les bases : Types primitifs et tableaux

- **Les types primitifs**
 - boolean : true / false
 - char : caractère encodé sur 16 bits
 - int : entier sur 32 bits
 - double : réel IEEE754 encodé sur 64 bits
- **La déclaration et l'assignation des variables**
 - char uneLettre = 'r';
 - double somme = 25.63;
- **Les tableaux**
 - **Allocation** : `int[] unTableau = new int[10];`
 - **Création** : `char[] unAutreTableau = {'J','a','v','a'};`
 - **Accès** : `unTableau[7] = 700;`
 - **Longueur** : `int length = unTableau.length;`

La notion de portée est la même qu'en Python. Les blocs étant ici délimités par des accolades. ({ })

Les bases : Opérateurs

- Les opérateurs (par ordre de priorité)
 - multiplication et division : $*$, $/$
 - addition et soustraction : $+$, $-$
 - comparaison : $<$, $<=$, $>$, $>=$
 - égalité : $==$, $!=$
 - 'et' logique : $\&\&$
 - 'ou' logique : $||$
- L'ordre des priorité est le même qu'en algèbre, avec des parenthèses.

Les bases : Types Objets

- Les types "objets", wrappers des types primitifs

- Integer

- | `Integer value = new Integer(10);`
`Integer value = 10;`

- Float

- | `Float value = new Float(10.0);`
`Float value = 10.0;`

- Double

- | `Double value = new Double(10.0);`
`Double value = 10.0;`

- String

- | `String value = new String("Bonjour");`
`String value = "Bonjour";`

- Boolean

- | `Boolean value = new Boolean(true);`
`Boolean value = true;`

- Des listes ?

- ArrayList

- | `ArrayList<String> list = new ArrayList<String>();`

Les bases : String

- **Les chaînes de caractères sont des objets**

- **Création**

```
| String greeting = "Hello world !";
```

- **Longueur**

```
| int length = greeting.length();
```

- **Accès**

```
| char c = greeting.charAt(2);
```

- **Concaténation**

```
| String maChaine = "Hello, " + " world !";
```

Les bases : Conversions

Equivalents des fonctions 'str()', 'int()' et 'float()' vues en Python.

- **Convertir un String en une valeur numérique**

```
| Integer value = Integer.valueOf("20");  
| Float value = Float.valueOf("10.5");  
| Double value = Double.valueOf("10.5");
```

- **Convertir un String en valeur booléenne**

```
| Boolean value = Boolean.valueOf("true");
```

- **Convertir une valeur numérique en String**

```
| String value = String.valueOf(123);
```

Les bases : Instruction IF

- **IF**

Python

```
| if condition:  
|     instructions
```

Java

```
| if (condition) {  
|     instructions  
| }
```

- **IF ELSE**

Python

```
| if condition:  
|     instructions  
| else:  
|     instructions
```

Java

```
| if (condition) {  
|     instructions  
| } else {  
|     instructions  
| }
```

- **IF ELSE IF ELSE**

Python

```
| if condition:  
|     instructions  
| elif:  
|     instructions  
| else:  
|     instructions
```

Java

```
| if (condition) {  
|     instructions  
| } else if (condition) {  
|     instructions  
| } else {  
|     instructions  
| }
```

Les bases : Instruction WHILE

- **WHILE**

Python

```
| while condition:  
|     instructions
```

Java

```
| while (condition) {  
|     instructions  
| }
```

- **DO WHILE**

Python

```
| instructions  
| while condition:  
|     instructions
```

Java

```
| do {  
|     instructions  
| } while (condition);
```

Les bases : Instruction FOR

- **FOR**

Python

```
| while condition:  
|     instructions
```

Java

```
| for (initialisation ; condition ;  
|     increment) {  
|     instructions  
| }
```

- **FOR EACH**

Python

```
| for elem in list:  
|     instructions
```

Java

```
| int[] list = {10,1,2};  
|  
| for (int elem : list) {  
|     instructions  
| }
```

Quoi d'autre ?!

- **Scanner** : Pour récupérer des informations au clavier. (équivalent de 'input' en Python)

```
import java.util.Scanner;    // Librairie devant être importée.

Scanner scanner = new Scanner(System.in);
double doubleValue = scanner.nextDouble();
String stringValue = scanner.nextLine();
scanner.close();
```

- **ArrayList** : Pour manipuler des listes

```
import java.util.ArrayList;    // Librairie devant être importée.

ArrayList<String> list = new ArrayList<String>();

list.add("Bonjour");           // Equivalent au '.append' en Python
list.add("Au revoir");

list.add("Salut", 1);          // Equivalent au '.insert' en Python

String salut = list.get(1);     // Equivalent au '[1]' en Python

list.remove(1);                // Equivalent au 'del list[1]' en Python

int length = list.size()       // Equivalent au 'len(list)' en Python
```


Quoi d'autre ?! (suite)

- **Lecture de fichiers :** (équivalent à 'read')

```
import java.io.*;           // Librairie devant être importée.
import java.util.*;         // Librairie devant être importée.

String filename = "test.txt";
BufferedReader file = new BufferedReader(new FileReader(filename));
String text = file.read();
file.close();
```

- **Lecture de fichiers :** (équivalent à 'readlines')

```
import java.io.*;           // Librairie devant être importée.
import java.util.*;         // Librairie devant être importée.

String filename = "test.txt";
String line;
BufferedReader file = new BufferedReader(new FileReader(filename));
while ((line = file.readLine()) != null) {
    doSomethingWithThisString(line);
}
in.close();
```

- **Ecriture de fichiers :** (équivalent à 'write')

```
import java.io.*;           // Librairie devant être importée.
import java.util.*;         // Librairie devant être importée.

String filename = "test.txt";
String text = "some text";
BufferedWriter file = new BufferedWriter(new FileWriter(filename));
file.write(text);
file.close();
```

Documentation

- Accès à la documentation en ligne
- Librairie Integer
- Librairie Float
- Librairie Double
- Librairie String
- Librairie Math
- Librairie Scanner
- Librairie ArrayList
- Librairie BufferedReader
- Librairie BufferedWriter

Google peut aussi vous servir pour localiser plus vite une information dans la documentation Java !