

INF1256 Informatique pour les sciences de la gestion
– Éléments de programmation –

Johnny TSHEKE, Ing. Jr.

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
DÉPARTEMENT D'INFORMATIQUE
TSHEKE_SHELE.JOHNNY@UQAM.CA

SÉANCE 02

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)
- 3 Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java
- 5 Environnement de développement

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)
- 3 Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java
- 5 Environnement de développement

Problème : convertisseur de température

On demande de développer une application permettant de convertir une température exprimée en degrés Celsius ($^{\circ}\text{C}$) en degrés Fahrenheit ($^{\circ}\text{F}$).

Solution théorique : conversion température

Fusion de l'eau $0\text{ }^{\circ}\text{C} = 32\text{ }^{\circ}\text{F}$

Ébullition de l'eau $100\text{ }^{\circ}\text{C} = 212\text{ }^{\circ}\text{F}$

Échelles : $^{\circ}\text{C} \rightarrow 100$, $^{\circ}\text{F} \rightarrow 180$

Convertir : $x\text{ }^{\circ}\text{C}$ en $y\text{ }^{\circ}\text{F}$ ou vice versa

$^{\circ}\text{C} \rightarrow ^{\circ}\text{F}$: $y\text{ }^{\circ}\text{F} = ((x * (9/5)) + 32)\text{ }^{\circ}\text{F}$

$^{\circ}\text{F} \rightarrow ^{\circ}\text{C}$: $x\text{ }^{\circ}\text{C} = ((y - 32) * (5/9))\text{ }^{\circ}\text{C}$

Conception : °C → °F

Déterminer : entrée, traitement, sortie

Entrée : x °C (Demander à l'utilisateur)

Traitement : Appliquer la formule y °F = $(x * (9/5)) + 32$

Sortie : Afficher le résultat y

Pseudo-code du programme

```
conversionCelciusAFahrenheit
```

```
DEBUT
```

```
  x (réel)
```

```
  y (réel)
```

```
  ECRIRE "Entrer la température en degrés Celsius svp "
```

```
  LIRE x
```

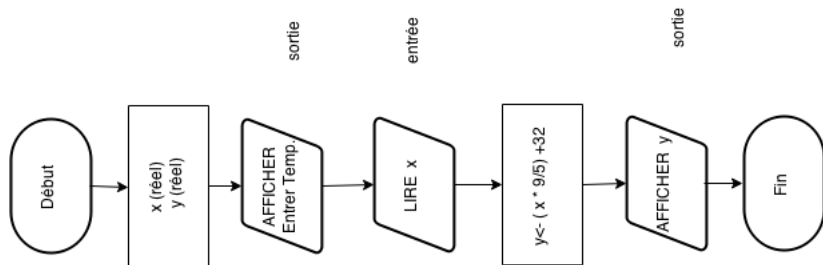
```
   $y \leftarrow (x * (9/5)) + 32$ 
```

```
  ECRIRE x, " degrés Celsius égal à ", y, " degrés Fahrenheit"
```

```
FIN
```

⇒ Comment représenter cet algorithme graphiquement ?

Organigramme (Flowchart) de l'algorithme



⇒ Il faut maintenant trouver un moyen d'exécuter cet algorithme sur machine

Comment exécuter cet algorithme sur machine ?

- Choisir un langage de programmation précis (éventuellement avec un environnement de développement)
- Convertir l'algorithme dans ce langage → programme
- (Au besoin compiler le programme)
- Exécuter le programme

⇒ Dans ce cours, nous utiliserons le langage **Java**

⇒ Version **Java 8** <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⇒ IDE (Integrated Development Environment) : **Eclipse**

⇒ Version **neon** www.eclipse.org (Eclipse IDE for Java EE Developers)

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)**
- 3 Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java
- 5 Environnement de développement

Nom

Les noms servent à identifier. Si plusieurs mots, alors attacher les mots en mettant la première lettre de chaque mot en majuscule et les autres en minuscule selon les règles suivantes :

- Sensible à la casse
- Classe : Première lettre de chaque mot commence avec Majuscule (ex : Conversion, ConversionEchelles)
- Méthode : Première lettre en minuscule, première Lettre de chaque mot en majuscule, (ex : main, calculerTemperature)
- les variables : Comme pour les méthodes sauf pour les constantes qu'il faut mettre en majuscule (voir section variables)
- ...

⇒ Tous ces noms sont des **identifiants**

Comment former un identifiant ?

- Commencer par une lettre ou un soulignement (_) ou un dollar (\$)
- Suivi d'un nombre quelconque de lettres, chiffre ou soulignement
- Sensibilité à la casse (Une lettre majuscule est différente d'une lettre minuscule)
- Pas d'espace blanc !
- Pas de mot réservé !

Exemples :

- y
- temp
- temp_celcius
- Temp
- valeur

Mots réservés

- font partie du Java
- Ne peuvent pas être utilisés comme d'autres noms ou identifiants ordinaires

<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert***</code>	<code>default</code>	<code>goto*</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum****</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp**</code>	<code>volatile</code>
<code>const*</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

* not used

** added in 1.2

*** added in 1.4

**** added in 5.0

Ainsi que `true`, `false` et `null`.

⇒ voir http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)
- 3** Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java
- 5 Environnement de développement

Comment se présente un programme Java

- Une classe contenant des instructions
- Enregistrer dans un fichier portant le même nom que la classe et ayant l'extension `.java`
- Compiler avec la commande :
`javac NomFichier.java`
- Si la compilation ok → exécution avec la commande :
`java NomFichier`
- `package` spécifie l'espace de nom (regroupement) d'appartenance de la classe

Exemple d'un premier programme Java `HelloWorld.java`

```
package inf1256s02;  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        // Premier programme Java  
        System.out.println(" Hello World!");  
    }  
  
}
```

Contrôle d'accès aux éléments d'une classes

Access Levels

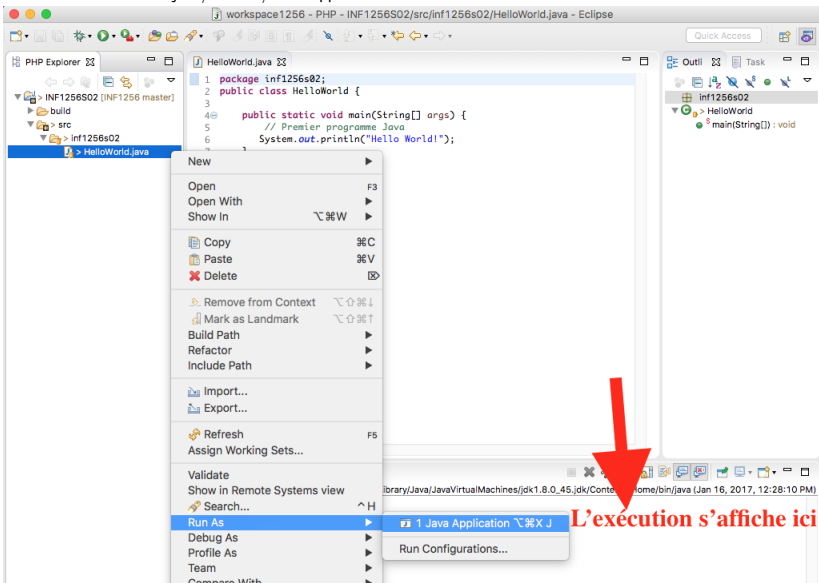
Modifieur	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
<code>private</code>	Y	N	N	N

- Niveau classe : `public` si classe accessible de partout. Si pas de protection explicite alors accessible seulement dans le package.
- Niveau attribut de classe : `public` , `protected`, `private`, défaut (pas de protection explicite) Telle que illustré dans le tableau

⇒ voir <https://docs.oracle.com/javase/tutorial/java/java00/accesscontrol.html>

Exécution en ligne de commande sur un terminal

Souris de droite sur le fichier .java/Run As/Java Application



The screenshot shows the Eclipse IDE interface. The main editor displays the code for `HelloWorld.java`:

```
1 package inf1256s02;
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5         // Premier programme Java
6         System.out.println("Hello World!");
7     }
8 }
```

The 'Run As' context menu is open, and the 'Run As' option is selected, which has opened a sub-menu with '1 Java Application' selected. A red arrow points to the 'Run As' option in the main menu, and a red text label 'L'exécution s'affiche ici' is positioned next to the arrow.

workspace1256 - PHP - INF1256S02/src/inf1256s02/HelloWorld.java - Eclipse

PHP Explorer

inf1256s02 [INF1256 master]

build

src

inf1256s02

HelloWorld.java

Quick Access

inf1256s02

HelloWorld

main(String[]): void

New

Open F3

Open With

Show In ⌘⇧W

Copy ⌘C

Paste ⌘V

Delete ⌘X

Remove from Context ⌘⇧↓

Mark as Landmark ⌘⇧↑

Build Path

Refactor

Include Path

Import...

Export...

Refresh F5

Assign Working Sets...

Validate

Show in Remote Systems view

Search... ⌘H

Run As ⌘X J

1 Java Application

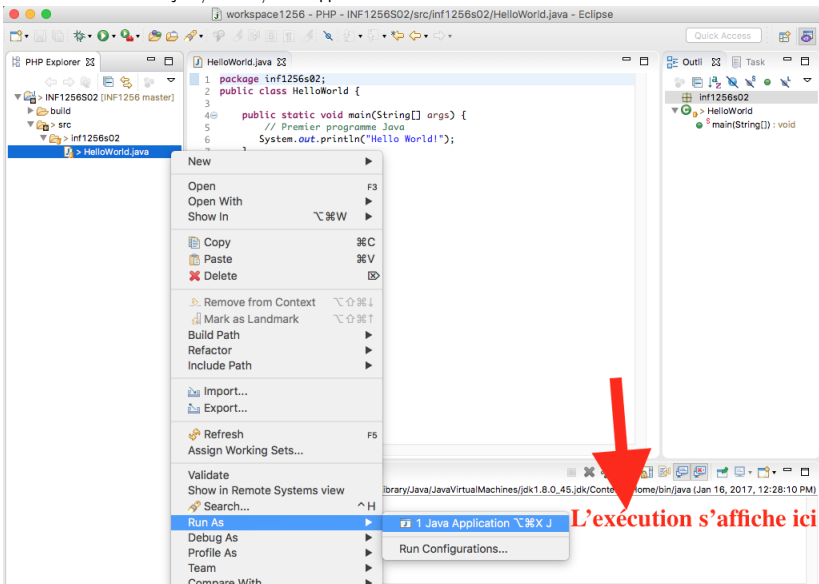
Run Configurations...

L'exécution s'affiche ici

Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/bin/java (Jan 16, 2017, 12:28:10 PM)

Exécution du programme sur eclipse

Souris de droite sur le fichier .java/Run As/Java Application



The screenshot shows the Eclipse IDE interface. The main editor displays the code for `HelloWorld.java`:

```
1 package inf1256s02;
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5         // Premier programme Java
6         System.out.println("Hello World!");
7     }
8 }
```

The 'Run As' context menu is open, and the 'Run As' option is highlighted. A red arrow points to the 'Run As' option, and the text 'L'exécution s'affiche ici' is written in red next to it.

The 'Run As' menu items are:

- New
- Open (F3)
- Open With
- Show In (⌘⌘W)
- Copy (⌘C)
- Paste (⌘V)
- Delete (⌘X)
- Remove from Context (⇧⌘↓)
- Mark as Landmark (⇧⌘↑)
- Build Path
- Refactor
- Include Path
- Import...
- Export...
- Refresh (F5)
- Assign Working Sets...
- Validate
- Show in Remote Systems view
- Search... (⌘H)
- Run As (⌘X J)**
- Debug As
- Profile As
- Team
- Compare With

The 'Run As' submenu is also visible, showing:

- 1 Java Application (⌘X J)
- Run Configurations...

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)
- 3 Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java**
- 5 Environnement de développement

Programme Java de Conversion de température

Exemple d'un programme Java de conversion de température **Conversion-CelciusAFahrenheit.java**

```

package inf1256s02;
import java.util.*; //importation des classes
public class ConversionCelciusAFahrenheit {

    public static void main(String[] args) {
        // ATTENTION: ecrire les nombres réels avec .
        Double tCel; //déclaration
        Scanner sc; //déclaration --pour lecture entrée
        sc=new Scanner(System.in); //affectation d'une instance
        System.out.println("Entrez la temperature en degre Celcius svp");
        tCel=sc.nextDouble(); //affectation nombre reel saisi au clavier
        Double tFah = (tCel * (9/5.0))+32;
        System.out.println("La temperature en degré Fahrenheit est: "+tFah);
        sc.close(); //fermeture scanner

    }
}

```

- 1 Un exemple de programmation
- 2 Éléments lexicaux (vocabulaire)
- 3 Structure d'un programme Java
- 4 Convertir un pseudo-code en programmes Java
- 5 Environnement de développement

IDE – Logiciel de développement

- Plateforme : Eclipse <http://www.eclipse.org/> (gratuit)
- Compilateur/ Interpréteur (java 8) : <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
prendre la version JDK.

⇒ Illustration en classe

Gestion de code source

`git https://git-scm.com/downloads`

Documentation : `https://git-scm.com/book/fr/v2`

⇒ Illustration en classe.