

Algorithmique et programmation

TP 1 : Le langage Java, structure d'un programme, compilation, types de base, affichage et lecture de données

Le langage de programmation utilisé pour les TP d'algorithmique et programmation est le langage Java. Le but de ce premier TP est d'écrire, de compiler et d'exécuter des premiers programmes simples.

1 Le premier programme : *Hello.java*

1.1 Organisation et saisie du premier programme

Avant tout, il est important de bien organiser vos travaux. Pour ce faire, commencer par créer un répertoire pour les TP dans votre répertoire personnel. Éventuellement, créer un sous-répertoire *TP1*.

Saisir le programme suivant dans votre éditeur de textes préféré. Dans le doute, utiliser l'éditeur recommandé, à savoir *emacs*.

```
/******  
* Informations sur le programme : auteur, date, etc. *  
*****/  
  
class Hello {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World !!!");  
    }  
  
}
```

Sauvegarder le programme sous *Hello.java*. NB : l'extension *.java* est importante pour indiquer qu'il s'agit d'un programme Java.

1.2 Compilation

Avant de pouvoir exécuter le programme, il faut le transformer en fichier exécutable. Cette opération, appelée *compilation* est réalisée par un *compilateur*. Le compilateur que vous allez utiliser s'appelle *javac* et s'utilise de la manière suivante :

```
javac Hello.java
```

Le résultat de la compilation se trouve dans le (nouveau) fichier *Hello.class*.

Charger le fichier *Hello.class* dans votre éditeur, la version binaire du programme apparaît. Cette version binaire est en réalité du *bytecode Java* destiné à être exécuté par une *machine virtuelle Java*. Ceci sera l'objet de la section suivante.

1.3 Exécution et compréhension du programme par déduction

- Après avoir compilé le programme, l'exécuter à l'aide de la commande :
`java Hello`
- déduire ce que fait l'instruction `System.out.println` ;
- insérer plusieurs instructions `System.out.println` ;
- remplacer `System.out.println` par `System.out.print` sur certaines lignes, et observer la différence.

2 Affichages de données

- Déclarer des variables dans *Hello.java* avec les types de base **int**, **double** et **char** ;
- initialiser les variables au niveau de leur déclaration (cf. exemple ci-dessous) ;
- afficher les valeurs des variables ;
- décommenter les deux dernières lignes de l'exemple, et observer la différence.

```
/******  
* Hello2.java : version 2.0 *  
* * * * *  
* Déclaration et affectation de variables *  
*****/  
  
class Hello2 {  
  
    public static void main(String [] args) {  
        int entier = 2014;  
        double reel = 3.1416;  
        char lettre = 'S';  
  
        System.out.println(entier);  
  
        System.out.println("entier : " + entier);  
        System.out.println("reel ..: " + reel);  
        System.out.println("lettre : " + lettre);  
  
        // System.out.println(entier + reel);  
        // System.out.println("" + entier + reel);  
    }  
  
}
```

3 Lecture de données au clavier

La lecture de données au clavier peut se faire à l'aide d'un objet spécial `Scanner`. Compiler et étudier l'exemple suivant (*Hello3.java*) pour en comprendre le fonctionnement.

```

/*****
 * Hello3.java : version 3.0
 *
 * Déclaration et affectation de variables
 *****/

import java.util.*;

class Hello3 {

    static final Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        int entier;
        double reel;
        String chaine;

        // input.useLocale(Locale.ROOT);

        System.out.println("Entrer un entier : ");
        entier = input.nextInt();
        System.out.println("Donner un mot : ");
        chaine = input.next();
        System.out.println("Valeurs saisies : " + entier + " " + chaine);

        System.out.println("Une autre valeur entière suivie d'un réel svp : ");
        entier = input.nextInt();
        reel = input.nextDouble();
        System.out.println("Valeurs saisies : " + entier + " " + reel);
        System.out.println("That's all folks");
    }
}

```

4 Influence du type des opérandes ; conversion de températures

On vous demande d'écrire un programme qui convertit une température donnée en degrés Fahrenheit en degrés Celsius.

- Pour ce faire vous devez demander à l'utilisateur d'entrer une température, puis la convertir en utilisant la formule suivante :

$$celsius = 5 \times (fahr - 32) / 9$$

- tester votre programme avec pour valeur de température 106 ;
- après avoir testé la formule précédente, modifier la formule comme suit :

$$celsius = 5/9 \times (fahr - 32)$$

- tester le programme avec la même température que précédemment, que constatez-vous ? Le type du résultat d'une opération dépend du type de ses opérandes, ce qui influe sur la valeur du résultat (division entière).

```
// Programme de conversion de degrés Fahrenheit vers Celsius
import java.util.*;
class Degres {
    static final Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        double fahr;
        double celsius;

        System.out.println("Programme de conversion deg. F vers deg. C");
        System.out.println("Entrer une température : ");
        fahr = input.nextDouble();

        celsius = 5 * (fahr - 32) / 9;

        // celsius = 5 / 9 * (fahr - 32);

        System.out.println("La température en Celsius est : " + celsius);
    }
}
```