

# TP 2 : variables, expressions et instructions

Informatique Fondamentale (IF1)

Semaine du 27 Septembre 2010

Commencez par créer un répertoire nommé `tp2` dans lequel vous travaillerez durant ce TP.

## 1 La classe Deug

Comme vous le savez, un programme Java commence par des instructions de la forme `■ import xxx.yyy.zzz.ttt; ■` qui permettent d'importer des paquetages, c'est-à-dire d'utiliser des ensembles de fonctions prédéfinies. On utilisera souvent le paquetage `Deug` en mettant au début des programmes

```
import fr.jussieu.script.Deug;
```

**Exercice 1.** Sans y passer trop de temps, jetez un coup d'œil à la documentation du paquet `Deug`, qui se trouve à l'adresse

<http://www.liafa.jussieu.fr/~yunes/deug/Deug/>

À quoi sert la fonction `Deug.println(int)` ? La fonction `Deug.readInt()` ?

## 2 Mise en jambes

**Exercice 2.** Écrivez un programme `Nom` qui affiche votre prénom. Compilez et testez le

## 3 Expressions arithmétiques

**Exercice 3.** Écrivez un programme `Moyenne` qui lit quatre notes et affiche leur somme et leur moyenne. Compilez et testez le.

**Exercice 4.** Écrivez un programme `Cercle` qui demande le rayon d'un cercle et affiche son périmètre et sa surface. Vous pourrez vous servir de la constante `Math.PI` dont la valeur est une approximation de  $\pi$ . Compilez et testez votre programme.

**Exercice 5.** Quel est le type et la valeur de chacune des expressions suivantes ?

<code>4 / 3</code>	<code>(4 - 3) * 5</code>	<code>1.3 / 0</code>
<code>4 / 3.0</code>	<code>117 % 7</code>	<code>(-1.3) / 0</code>
<code>4 - 3 * 5</code>	<code>0.3 - 0.2</code>	<code>0 / 1.3</code>
<code>4 - (3 * 5)</code>	<code>0.2 - 0.1</code>	<code>0 / 0</code>

Écrivez un programme `ExprArith` qui évalue et affiche la valeur de chacune d'elles pour vérifier vos hypothèses.

**Exercice 6.** Écrivez un programme `Conversion` qui demande une température en degrés Celsius et la convertit en degrés Fahrenheit. On rappelle la formule

$$f = \frac{9c}{5} + 32$$

où  $f$  est la température en degrés Fahrenheit et  $c$  en degrés Celsius. (Pensez au type des variables — ce ne sont pas forcément des entiers.) Compilez et testez votre programme.

**Exercice 7.** Écrivez un programme `TTC` qui à partir d'un prix hors taxes et d'un taux de TVA calcule le prix toutes taxes comprises correspondant. Compilez et testez le.

## 4 Opérateurs booléens

**Exercice 8.** Quelles sont les valeurs des expressions booléennes suivantes ?

<code>10 &gt; 5</code>	<code>false    (5 != 4)</code>
<code>10 == 5</code>	<code>false &amp;&amp; (5 != 4)</code>
<code>5 == 5</code>	<code>!(30 % 3 == 0)</code>
<code>5 == 11 - 6</code>	<code>0.3 - 0.2 == 0.2 - 0.1</code>

Vérifiez vos réponses en écrivant un programme qui les évalue.

**Exercice 9.** Écrivez un programme `Bissextile` qui détermine si une année est bissextile. (On rappelle qu'une année est bissextile si son numéro est multiple de 4, sauf si l'année est aussi multiple de 100 mais pas de 400.)

## 5 Compilation sous XEmacs

Dans le TP précédent nous avons appris à compiler un programme java à l'aide d'une fenêtre shell. Cela consistait à ce placer dans le répertoire où se trouve le fichier à compiler, puis à taper la commande `javac MonProgramme.java`. Il est également possible de compiler un programme directement sous XEmacs.

**Exercice 10.** Tapez le programme suivant :

```
import fr.jussieu.script.Deug;

public class ProgrammeBogue{
    public static void main(String[] args){
        int x, x_cube, x_carre;
        Deug.println("Entrez un entier");
        x = Deug.readInt();
        x_carre = x * x;
        Deug.println("la valeur de x carre est " + x_carre);
        x_cube = x_carre * x;
```

```

        Deug.println("la valeur de x cube est " + x_cube);
    }
}

```

1. Compilez le à l'aide d'XEmacs, pour cela appelez la commande `compile`, en tapant d'abord `M-x` (tenez appuyée la touche *Alt* pour `M-`, appuyez sur la touche `x`, puis relâchez la touche *Alt*). Le curseur passe alors dans le *mini buffer*, situé en bas de la fenêtre d'XEmacs, tapez alors `compile`.
2. Remplacez `make -k` par la commande de compilation appropriée

```
javac ProgrammeBogue.java
```

 puis appuyez sur la touche *entrée*.
3. Pointez sur une erreur, et cliquez sur le bouton du milieu de la souris, que se passe t-il ?
4. Corrigez les erreurs et recompilez jusqu'à ce le programme fonctionne.

Dans la suite des TP, c'est à vous de décider si vous préférez compiler directement à partir du shell, ou si vous trouvez que c'est plus pratique à partir d'Emacs.

## 6 Chaînes de caractères

**Exercice 11.** Écrivez un générateur de lettres de félicitations au chargé de TP. Ce programme, appelé `Lettre`, devra demander le prénom d'un chargé de TP et votre prénom et devra afficher un message de la forme suivante :

```

Cher chargé_de_TP,
J'adore vos TPs, ils sont tellement bien !
Signé étudiant.

```

Comme d'habitude, compilez et testez votre programme.

Modifiez ensuite votre programme pour insister sur la qualité des TPs en rajoutant des guillemets autour de `■ bien ■`, c'est-à-dire en affichant maintenant `■ ils sont tellement "bien" ! ■`. Comme d'habitude, compilez et testez votre programme.

**Exercice 12.** Quel est le type et la valeur des expressions suivantes ?

1. `'a'`
2. `"pif\npouf"`
3. `'\\'`
4. `'\"'`
5. `"aujourd'hui"`
6. `"dites \"Aaaaah\" !"`
7. `"un\ttext\tespacé"`
8. `"deux" + "deux"`
9. `"deux + deux"`
10. `'o' + "k"`

Vérifiez vos réponses à l'aide d'un programme.

## 7 S'il vous reste du temps

**Exercice 13.** Écrivez un programme qui convertit un nombre de secondes en une durée exprimée en heures, minutes et secondes.

**Exercice 14.** Écrivez un programme qui demande à l'utilisateur trois entiers consécutifs, affiche `true` s'ils sont donnés en ordre croissant, `false` sinon.