TP 1 : Variables, Expressions et Instructions

Informatique Fondamentale (IF121)

29 septembre – 3 octobre 2003

L'objectif de ce TP est d'écrire, compiler et exécuter ses premiers programmes Java. Les parties du cours abordées sont :

- les variables et les types
- les expressions arithmétiques
- les expressions booléennes (logiques)
- les entrées/sorties (interaction avec le terminal)
- les chaînes de caractères

Tous les TPs nécessitent l'utilisation de la classe `Deug` dont la documentation se trouve à l'adresse suivante : `http://www.liafa.jussieu.fr/~yunes/deug/Deug/`

Tous les fichiers Java contenant vos programmes devront commencer par un en-tête indiquant le nom de la classe, les auteurs, la date et une notice explicative, sous la forme :

```
/* MaSuperClasse
 *
 * auteurs : Gustave Martin et Antoinette Dupond
 *
 * date : le 25 septembre 2003
 *
 * Affiche le message ``Bonjour''.
 */
```

1 Ma première tasse de Java

Pour écrire votre premier programme Java[1], il vous faut :

- un éditeur de texte (kwrite, xemacs, pico, vi, ...)
- la plateforme Java contenant notamment le compilateur Java (commande `javac`) et la machine virtuelle Java (commande `java`).
- la classe `fr.jussieu.script.Deug`.

Exercice 0: Vérifier que les commandes `java` et `javac` sont disponibles.

Votre première application, `Hello`, affichera "Bonjour". Pour créer ce programme, vous allez :

1. Créer un fichier `source` Java. Un fichier source contient du texte, écrit en Java. Vous pouvez utiliser n'importe quel éditeur de texte pour créer et éditer des fichiers sources.

[1] http ://java.sun.com/docs/books/tutorial/getStarted/cupojava/unix.html#2

Exercice 1: Recopier le texte du code de la classe Hello présentée dans le TP0, exercice 3.4 et sauvegarder le fichier sous le nom `Hello.java`.

2. Compiler le fichier source dans un fichier de bytecode. Le compilateur Java, appelé `javac`, prend le fichier source `Hello.java` et le traduit en fichier `Hello.class` que la machine virtuelle Java peut comprendre : un fichier de bytecodes.

   Exercice 2: Placez-vous dans le répertoire contenant votre fichier `Hello.java` et exécuter la commande suivante : `javac Hello.java`.
   Si votre prompt réapparait sans message d'erreur, félicitations !, vous avez compilé avec succès votre premier programme Java. Sinon, lisez attentivement les messages d'erreur qui vous indiquent notamment la ligne suspecte dans le fichier source, ainsi que des informations sur la nature de l'erreur.
   Vérifiez que votre répertoire contient dorénavant le fichier `Hello.class`.

3. Exécuter le programme contenu dans le fichier de bytecode. La machine virtuelle Java (Java VM) interprète le fichier de bytecode et exécute les instructions.

   Exercice 3: Dans le même répertoire, exécuter la commande suivante : `java Hello`.
   Que se passe-t-il ?

Exercice 4: Le jeu des 4 erreurs
Télécharger le fichier `Hello2.java` disponible à l'adresse suivante :
`www-poleia.lip6.fr/~lepape/vrac/hello2.html`.
Ce fichier contient 4 erreurs. Corrigez-les de façon à ce que le programme fonctionne correctement.

2 Variables et types

Exercice 5: Ecrivez un programme `Typage` qui déclare une variable `x` de type `int`. Affectez `x` avec la valeur `1234` et compiler. Que se passe-t-il ?
Recommencer successivement d'affecter `x` avec les valeurs `-2`, `3.14f`, `4.5`, `"toto"` et `false`. Que se passe-t-il ?
Pour chaque valeur proposée, quels sont les types possibles qu'il aurait fallu donner à la variable `x` ?

Exercice 6:

1. Ecrivez un programme `Hello3` qui déclare une variable `message` de type String, l'affecte avec la valeur `"coucou !!!"` et affiche le message dans le terminal.
2. Ecrivez un programme `Echo` qui lit un message saisi par l'utilisateur dans le terminal et affiche ce message dans le terminal quand l'utilisateur appuie sur la touche Entrée.
3. Ecrire un programme Hello4 qui lit un nom saisi dans le terminal et affiche le message "Bonjour ... !", ou ... est remplacé par le nom.

3 Opérateurs arithmériques

Exercice 7: Ecrire un programme `ExprArith` qui évalue et affiche les expressions arithmétiques suivantes :

a) 4 / 3
b) 4 / 3.0
c) 4 - 3 * 5
d) 4 - (3 * 5)
e) (4 - 3) * 5
f) 117 % 7
g) 0.3 - 0.2
h) 0.2 - 0.1
i) 1.3 / 0
j) (-1.3) / 0
k) 0 / 1.3
l) 0 / 0

Exercice 8: Qu'affiche le programme suivant ?

```
int i = (int)3.14;
Deug.println("i = "+i);
```

Exercice 9: Ecrire un programme `Moyenne` qui lit 4 entiers, affiche leur somme et leur moyenne :

1. en utilisant quatre variables différentes pour les 4 entiers
2. en utilisant une seule variable pour la lecture des entiers
   indice : exploiter chaque nombre après l'avoir lu et réutiliser la même variable pour lire l'entier suivant.

Calculer la moyenne de 3, 5, 1 et 55 des deux façons.

Exercice 10: Conversion des degrés Celcius/Farenheit
Ecrire un programme qui :

1. demande à l'utilisateur de donner une température en degrés Celcius
2. effectue la conversion de cette température en degrés Farenheit grâce à la formule :
   `farenheit = celcius * 9/5 + 32`
3. affiche la température en degrés Farenheit

Exercice 11: Calcul d'un prix TTC
Ecrire un programme qui calcule le prix toutes taxes comprises (TTC) à partir d'un prix hors taxe (HT) et d'un taux : le prix TTC est égal au prix HT plus le prix HT multiplié par le taux divisé par 100.

Exercice 12:
a) Qu'affiche le programme suivant ?

```
int i = 10;
int n = i++%5;
Deug.println("i = "+i);
Deug.println("n = "+n);
```

b) Qu'afficherait-il si au lieu d'utiliser l'opérateur d'incrémentation postfix `(i++)` on avait utilisé l'opérateur d'incrémentation préfix `(++i)` ? Expliquez.
c) Pourquoi l'expression `25++` est-elle incorrecte ?

4 Opérateurs booléens

Exercice 13: Ecrire un programme `ExprBool` qui évalue et affiche les expressions booléennes suivantes :
a) $10 > 5$
b) $10 == 5$
c) $5 == 5$
d) $5 == (11 - 6)$
e) false || $(5 <> 4)$
f) false && $(5 <> 4)$
h) $!(30 \% 3 == 0)$
i) $((0.3 - 0.2) == (0.2 - 0.1))$

Exercice 14: Ecrire un programme `Bissextile` qui demande à l'utilisateur d'entrer une année et qui lui dit si l'année est bissextile.

indice : une année est bissextile si elle est multiplie de 4, à l'exception des années multiples de 100 mais pas de 400.

Exercice 15: Ecrire un programme `Logique` qui

1. demande à l'utilisateur d'entrer successivement trois variables entières a puis b et c
2. évalue l'expression booléenne correspondant aux conditions suivantes
   - a est supérieur à 3 et le quotient entier de la somme de a et de b par c est un multiple de 4
   - a divise b et a divise c
   - si a divise b alors a divise bc

Tester ces expressions avec les valeurs a=10, b=22, c=2.

5 Manipulation des chaînes de caractères

Exercice 16: Ecrire un programme `ExprString` qui évalue et affiche les expressions arithmétiques suivantes :

1. 'a'
2. `'\n'`
3. `'\u1234'`
4. `'\\'`
5. `'\"'`
6. `"aujourd'hui"`
7. `"dites \"Ahh!\" ."`
8. `"\tun\ttexte\ttrès\tespacé\t!"`
9. `"deux"+"deux"`
10. `'o'+'k'`
11. `'o'+"k"`

Exercice 17: A demi mots

Ecrire un programme `SousChaine` qui extrait et affiche la première moitié d'un mot donné par l'utilisateur.

Tester le programme avec les mots *ordinateur* et *programme*.

Exercice 18: Tête à queue

Ecrire un programme `TeteAQueue` qui inverse la première et la dernière lettre d'un mot donné par l'utilisateur.

Tester le programme avec les mots *informatique*, *laval*, et *ko*. Pourquoi le programme ne fonctionne-t-il pas avec le mot x?

Exercice 19: Prédiction

Ecrire un programme qui :

1. demande à l'utilisateur de donner un chiffre strictement compris entre 50 et 100.
2. ajoute ce nombre à 62 et affiche le résultat
3. supprime le premier chiffre de ce résultat et l'ajoute au chiffre restant (ex : 123 -> 1 + 23 = 24) puis affiche le résultat
4. enlève ce dernier nombre obtenu au premier nombre donné par l'utilisateur et affiche le résultat.

Vérifier que le résultat final vaut bien 37.

6 Blocs et Variables locales

Exercice 20: Le code suivant est-il correct ?

```
public static void main(String[] args){
   int n = 4;
   {
     int m = 5;
   }
   Deug.println(n);
   Deug.println(m);
}
```

Exercice 21: Le code suivant est-il correct ?

```
public static void main(String[] args){
   int n = 10;
   {
     int n = 100;
   }
   Deug.println(n);
}
```