

TP 1 : entrées/sorties, variables et fonctions

Introduction à l'informatique et à la programmation (IF1)

Semaine du 26 septembre 2011

Commencez par créer à l'aide du shell un répertoire nommé TP1 dans lequel vous travaillerez durant ce TP (ce répertoire devrait logiquement se trouver dans le répertoire ~/L1/IF1).

1 Premiers pas en Java

► Exercice 1 :

1. Dans le répertoire ~/L1/IF1/TP1, créez un fichier `Menthe.j v`, et écrivez-y le programme suivant :

```
class Menthe {  
    public static void main(String[] args) {  
        System.out.println("Java ,_c'est_pas_de_la_menthe_a_l'eau.");  
    }  
}
```

Sauvegardez-le, et vérifiez à l'aide de la commande `ls` que le fichier a bien été créé (vous pouvez également utiliser les commandes `cat`, `more` et `less` pour visualiser directement son contenu sans passer par un éditeur de texte).

2. Tapez la commande suivante : `j v c Menthe.j v`
Quels fichiers ont été créés ?
3. La commande `j v c` compile le *fichier source* `.j v` en un fichier `.class`. Le fichier nommé `Menthe.class`, dit *fichier bytecode*, contient un code dit *bytecode*, qui peut être exécuté par la commande `j v`. Exécutez-le : `j v Menthe`
4. Enlevez le point-virgule de la ligne 3 du fichier `Menthe.j v`, enregistrez le fichier et compilez-le à nouveau. Que se passe-t-il ?
5. Toujours dans le répertoire ~/L1/IF1/TP1, créez le fichier `Division.j v` suivant :

```
import java.util.Scanner;  
  
class Division {  
    public static void main(String[] args) {  
        int n, r;  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Entrez_un_entier");  
        n = sc.nextInt();
```

```

    r = 42 / n;
    System.out.println("Le_résultat_est_:" + r);
}
}

```

Ce programme demande à l'utilisateur d'entrer un entier n et affiche la partie entière de $42/n$. Compilez ce programme, vérifiez que le fichier *bytecode* a bien été créé, puis testez le programme. Que se passe-t-il si vous entrez 0 ?

2 Affichage et saisie

Comme vous l'avez constaté dans la section précédente, l'affichage d'un texte peut se faire par la méthode `System.out.println(...)`, à laquelle on fournit en argument ce qu'elle doit afficher.

- **Exercice 2 :** Dans un navigateur web, allez à l'url <http://download.oracle.com/javase/6/docs/api/>
C'est la documentation actuelle de `java 6`¹. Vous remarquerez que cette url contient le numéro de version (ici 6) et l'acronyme `api` (Application Programming Interface).
La page affichée est séparée en trois parties appelées *frames* (voir figure 1).

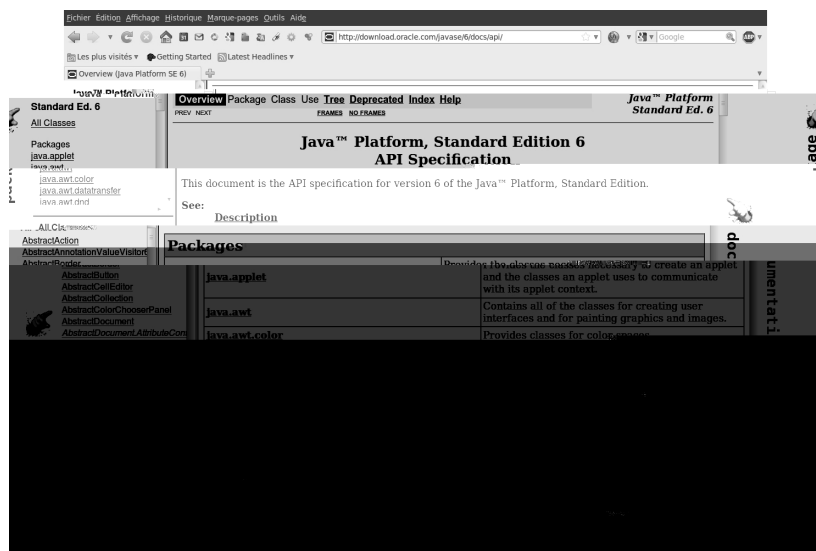


FIGURE 1 – Page d'accueil de la documentation `java`

1. Affichez la documentation de la classe `System` en cliquant sur `System` dans la frame des classes (en bas à gauche).
2. Dans le tableau des champs (`Field Summary`), suivez le lien vers le champs `out` : vous voyez que l'utilisation de `System.out.println(...)` est explicitée.

Prenez l'habitude quand vous utilisez une classe de l'API `java` de jeter un œil à sa documentation en cas de doute. Ajoutez le lien de l'API `java` dans vos favoris.

1. En fait `java 7` est sorti fin juillet 2011, mais n'est pas encore disponible pour la plateforme utilisée au `script`. Pour voir sa documentation, il suffit de remplacer 6 par 7 dans l'url.

- **Exercice 3 :** Ecrivez un programme `Nom` qui affiche votre prénom. Compilez-le et testez-le.
- **Exercice 4 :** Pour la saisie de données, on utilise la classe `Scanner`.
 1. Regardez la documentation de la classe `Scanner`. Pour utiliser la classe `Scanner` dans un programme, il faut indiquer au compilateur dans quel package elle se trouve, grâce à la directive d'importation


```
import java.util.Scanner;
```

 Repérez dans la documentation où est donné le nom complet de cette classe.
 2. Ecrivez un programme `Age` qui vous demande l'année actuelle et votre année de naissance et affiche votre âge, en supposant que votre anniversaire est déjà passé.
 3. Modifiez votre programme pour qu'il ait le comportement suivant :


```
> java Age
annee actuelle ? 2011
annee de naissance ? 1993
Si votre anniversaire est deja passe , vous avez 18 ans , sinon
vous en avez 17.
```

3 Expressions arithmétiques

- **Exercice 5 :** Écrivez un programme `Moyenne` qui lit quatre notes et affiche leur somme et leur moyenne. Compilez et testez-le.
- **Exercice 6 :** Écrivez une classe `Cercle` qui contient une fonction renvoyant le périmètre d'un cercle dont le rayon a été fourni en paramètre et une autre renvoyant sa surface. Pour tester ces fonctions, dans le `main`, demandez le rayon d'un cercle et affichez son périmètre et sa surface. Vous pourrez vous servir de la constante `Math.PI` dont la valeur est une approximation de π . Compilez et testez votre programme.
- **Exercice 7 :** 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 8 :** Écrivez une classe `Conversion` qui contient une fonction `celsius2Fahrenheit` prenant en argument une température en degré Celsius et renvoyant la température en degré Fahrenheit correspondante. On rappelle la formule

$$f = \frac{9c}{5} + 32$$
 où f est la température en degré Fahrenheit et c en degré Celsius. (Pensez au type des variables — ce ne sont pas forcément des entiers.) Dans le `main`, demandez une température en degré Celsius et convertissez-la en degré Fahrenheit. Compilez et testez votre programme.

- **Exercice 9 :** Dans la classe `Conversion` introduite à l'exercice 8, ajoutez une fonction `ht2Ttc` qui à partir d'un prix hors taxe et d'un taux de TVA (fournis en arguments) calcule le prix toutes taxes comprises correspondant.
Complétez le `main` pour testez cette fonction. Compilez et testez votre programme.

4 Opérateurs booléens

- **Exercice 10 :** Quelles sont les valeurs des expressions booléennes suivantes ?

<code>10 > 5</code>	<code>f lse (5 != 4)</code>
<code>10 == 5</code>	<code>f lse && (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 11 :** Dans une classe `Annee`, écrivez une fonction `estBissextile` qui détermine si une année (fournie en argument) est bissextile. (On rappelle qu'une année est bissextile si son numéro est multiple de 4, sauf si son numéro est aussi multiple de 100 mais pas de 400.)

5 Compilation sous XEmacs

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

- **Exercice 12 :** Tapez le programme suivant :

```
import java.util.Scanner;
```

```
public class ProgrammeBogue{
    public static void affichageCarreEtCube(int n){
        int n_carre, n_cube;

        n_carre = n * n;
        System.out.println("la_valeur_de_son_carre_est_" + n_carre);
        n_cube = n_carre * n;
        System.out.println("la_valeur_de_son_cube_
```

```

        affichageCarreEtCube(x);
    }
}

```

1. Compilez-le dans XEmacs, à l'aide de la commande `compile`. Pour cela, tapez 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` et `[Entrée]`. Remplacez `m ke -k` par la commande de compilation appropriée

```
j v c Progr mmeBogue.j v
```

puis appuyez sur la touche `[Entrée]`.

2. Pointez sur une erreur², et cliquez sur le bouton du milieu de la souris, que se passe-t-il ?
3. Corrigez les erreurs et recompilez jusqu'à ce que 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'XEmacs.

6 Chaînes de caractères

À partir d'ici, il n'est pas systématiquement précisé qu'il faut faire des fonctions séparées du `main`, mais pensez à le faire pour que la lecture du `main` soit allégée.

- **Exercice 13** : Écrivez un générateur de lettres de félicitation 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 TP, ils sont tellement bien !
Signé étudiant.

```

Comme d'habitude, compilez et testez votre programme.

Modifiez ensuite votre programme pour insister sur la qualité des TP 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.

7 S'il vous reste du temps

- **Exercice 14** : Écrivez un programme qui convertit un nombre de secondes en une durée exprimée en heures, minutes et secondes.
- **Exercice 15** : Écrivez un programme qui demande à l'utilisateur trois entiers consécutifs, affiche `true` s'ils sont donnés en ordre croissant, `false` sinon.

2. Le programme ne contient pas d'erreur, mais il est vraisemblable que vous en ayez ajouté en le copiant. Si ce n'est pas le cas, ajoutez-en une.