université PARIS DIDEROT PARIS 7

INTRODUCTION À L'INFORMATIQUE ET À LA PROGRAMMATION (IF1)

TP1 : ENTRÉES/SORTIES, VARIABLES ET EXPRESSIONS

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

1 Premiers pas en Java

► **Exercice 1** :

1. Créez un fichier `Menthe.java` (dans le répertoire `~/L1/IF1/TP1`) et écrivez-y le programme :

```
class Menthe {
    public static void main(String[] args) {
        System.out.println("Java, c'est pas de la menthe a l'eau.");
    }
}
```

   Sauvegardez-le, puis vérifiez à l'aide de la commande `ls` que le fichier a bien été créé [1].

2. Tapez la commande suivante : `javac Menthe.java` Quels fichiers ont été créés ?
3. La commande `javac` *compile* le *fichier source* `.java` en un fichier `.class` (voire plusieurs [2]). Ici, le fichier nommé `Menthe.class`, dit *fichier bytecode*, contient un code dit *bytecode*, qui peut être exécuté par la commande `java`. Exécutez-le : `java Menthe`
4. Enlevez le point-virgule de la ligne 3 du fichier `Menthe.java` , sauvegardez et compilez-le à nouveau. Que se passe-t-il ?
5. Remettez le point-virgule et remplacez l'identifiant `main` par `supermain`, sauvegardez et compilez-le à nouveau. Que se passe-t-il ? Que donne l'exécution ?
6. Toujours dans le répertoire `~/L1/IF1/TP1`, créez le fichier `Division.java` contenant :

```
import java.util.Scanner;

class Division {
    public static void main(String[] args) {
        int n, r;
        Scanner sc = new Scanner(System.in);
        System.out.print("Entrez un entier : ");
        n = sc.nextInt();
        r = 42 / n;
        System.out.println("Le resultat est : " + r);
    }
}
```

   Ce programme demande à l'utilisateur d'entrer un entier n et affiche le quotient de la division de `42` par `n`.

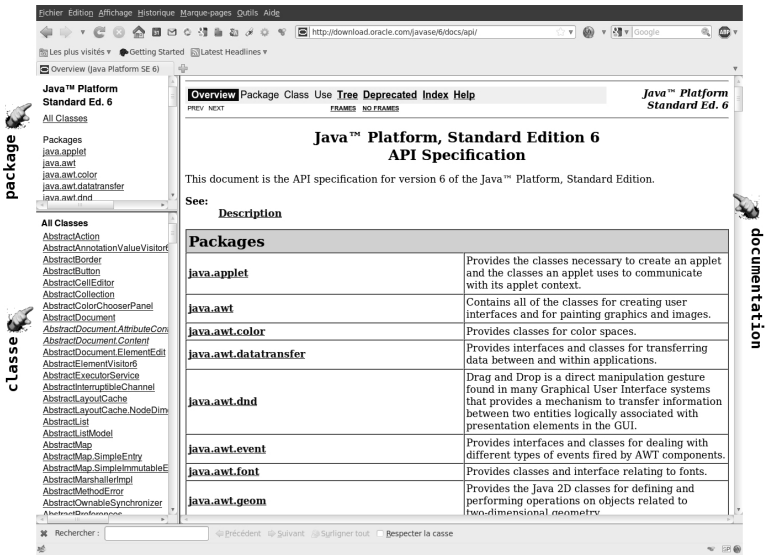
   Compilez ce programme, vérifiez que le fichier *bytecode* a bien été créé, puis testez le programme. Que se passe-t-il si vous entrez `0` ? Et si par vice ou par mégarde vous entrez `3.14` ?

1. Vous pouvez également utiliser les commandes `cat`, `more` et `less` pour visualiser directement son contenu sans passer par un éditeur de texte.
2. Toujours dans le fichier `Menthe.java`, vous pouvez copier-coller deux copies de `class Menthe{...}` à la suite, renommer celles-ci en `Menthe2` et `Menthe3`, modifier les messages en `"Java, c'est pas non plus du diabolo menthe."` ou `"Java, c'est vraiment pas de la tisane verveine-menthe."`, sauvegarder, compiler, examiner les fichiers ainsi créés et les exécuter.

2 Affichage et saisie

Les entrées/sorties peuvent se faire grâce aux classes `System` et `Scanner`.

► **Exercice 2** : Dans un navigateur, allez sur `http://docs.oracle.com/javase/6/docs/api/` C'est la documentation actuelle de `java` [3]. Vous remarquez 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* :



1. Affichez la documentation de la classe `System` en cliquant sur `System` dans la frame des classes (en bas à gauche). Dans le tableau des champs (`Field Summary`), suivez le lien vers le champs `out` : vous voyez que l'utilisation de `System.out.print(...)` ou `System.out.println(...)` est explicitée.
   Écrivez un programme `Nom` qui affiche votre prénom. Compilez-le et testez-le.
2. Affichez la documentation de la classe `Scanner`. Examinez le premier exemple qui y est donné.
   Écrivez 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.
   Modifiez le programme pour qu'il ait le comportement suivant :

```
> java Age
annee actuelle ? 2013
annee de naissance ? 1995
Si votre anniversaire est deja passe, vous avez 18 ans,
sinon vous en avez 17.
```

3. 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.

► **Exercice 3** : Écrivez un générateur de lettres de félicitation à vos chargé.e.s de TP. Le programme `Lettre` demande leurs prénoms et le vôtre et affiche un message de la forme suivante :

```
Chers chargés_de_TP,
J'adore vos TPs, ils sont tellement bien !
Signé étudiant.
```

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.

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

3 Expressions arithmétiques

► **Exercice 4** : Quel est le type et la valeur de chacune des expressions suivantes ?

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

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

► **Exercice 5** : Écrivez un programme `Moyenne` qui lit quatre notes et affiche leur somme et leur moyenne. Compilez et testez-le.

► **Exercice 6** : 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 π. Compilez et testez votre programme.

► **Exercice 7** : Écrivez un programme `Temperature` qui demande une température en degré Celsius et affiche 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. Compilez et testez votre programme.

► **Exercice 8** : En ces temps de crise, il peut être utile de rappeler que le *carat* est une mesure de pureté de métaux précieux. Un carat représente un vingt-quatrième de la masse totale d'un alliage. Par exemple, de l'or à 15 carats signifie que 24 g d'alliage contient 15 g d'or pur.

Écrivez un programme `Carat` qui demande deux entiers `all` et `met` et affiche la pureté (en carat) de l'alliage de masse `all` (en tonne) contenant une masse `met` (en tonne) de métal précieux. Compilez et testez-le.

► **Exercice 9** : Écrivez un programme `Euclide` qui demande deux entiers `a` et `b` et affiche la division euclidienne `a = b*q + r` (avec `0≤r<b`) et sa description « en `a` combien de fois `b` ? euh... `q` fois et il reste `r` ». Compilez et testez-le.

► **Exercice 10** : Écrivez un programme `Seconde` qui demande une durée en secondes et affiche cette durée exprimée en heures, minutes et secondes. Compilez et testez-le.

Une expression de type `int` peut être vue comme étant de type `double` (conversion implicite). Dans le sens contraire, il faut expliciter la conversion via l'opérateur `(int)`, qui peut être considéré comme un opérateur de troncature : l'évaluation de `(int)Math.PI` donne `3` et celle de `(int)-Math.PI` donne `-3`.

► **Exercice 11** : Quel est le type et la valeur de chacune des expressions suivantes ?

```
7.2 / 4        7.2 / 4.5        (int) (7.2 / 4)        (int) 7.2 / 4
7 / 4.5        7 / 4            (double) (7 / 4)       (double) 7 / 4
```

Écrivez un programme `Transtypage` qui affiche la valeur de chacune d'elles pour vérifier vos réponses.

► **Exercice 12** : L'évaluation de `Math.random()` renvoie un réel pseudo-aléatoire de l'intervalle $[0,1[$. Donnez une expression dont l'évaluation renvoie un entier pseudo-aléatoire dans $\mathscr{D} = \{1,2,3,4,5,6\}$. Écrivez un programme pour la tester, puis faites varier $\mathscr{D}$.

► **Exercice 13** : Modifiez les programmes des exercices 5, 6, 7 et 8 de sorte que chacun demande le degré de précision attendu, c'est-à-dire le nombre de chiffre(s) après la virgule dans la réponse. Compilez et testez-les.

4 Expressions booléennes

► **Exercice 14** : Quelles sont les valeurs des expressions booléennes suivantes ?

```
10 > 5       5 == 5           false || (5 != 4)      !(30 % 3 == 0)
10 == 5      5 == 11 - 6      false && (5 != 4)      0.3 - 0.2 == 0.2 - 0.1
```

Vérifiez vos réponses en écrivant un programme `Bool` qui les évalue.

► **Exercice 15** : Écrivez une expression booléenne à partir de trois variables entières qui vaut `true` si leurs valeurs sont ordonnées de façon croissante, `false` sinon. Écrivez un programme pour la tester.

► **Exercice 16** : On rappelle qu'une année `a` est bissextile si `a` est multiple de 4, sauf si `a` est aussi multiple de 100 mais pas de 400. Donnez une expression booléenne qui vaut `true` si `a` est bissextile et `false` sinon. Écrivez un programme pour la tester.

5 Compilation sous Emacs

Nous avons vu comment compiler un programme `java` dans une fenêtre shell. Cela consistait à se placer dans le répertoire du fichier à compiler, puis à taper la commande `javac MonProgramme.java`

Il est également possible de compiler un programme directement sous Emacs.

► **Exercice 17** : 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␣est␣" + n_cube);
    }
    public static void main(String[] args){
        int x;
        Scanner sc = new Scanner(System.in);
        System.out.println("Entrez␣un␣entier");
        x = sc.nextInt();
        affichageCarreEtCube(x);
    }
}
```

1. Compilez-le dans Emacs, à l'aide de la commande `compile`. Pour cela, tapez d'abord `M-x` (tenez appuyée Alt pour `M-`, appuyez sur x, puis relâchez Alt). Le curseur passe alors dans le *mini buffer*, situé en bas de la fenêtre d'Emacs, tapez alors `compile` et Entrée. Remplacez `make -k` par la commande de compilation appropriée `javac ProgrammeBogue.java` puis appuyez sur Entrée.
2. Pointez sur une erreur[4], 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, 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.

4. 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 ou plusieurs.