Licence Informatique
Université Bordeaux 1
2006-2007

UE INF203
Programmation par objets
Travaux Dirigés
feuille 1

Edition, compilation, exécution

Exercice 1.

1. À votre racine créer le répertoire `programmation-par-objets`
   Ce répertoire contiendra l'ensemble de votre travail en programmation par objets.
2. Se placer dans le répertoire `programmation-par-objets`
   Y créer le sous-répertoire `src`. Se placer dans le répertoire `src`.
3. Télécharger les fichiers qui accompagnent le td à l'aide de la commande
   `wget http://dept-info.labri.fr/ENSEIGNEMENT/poo/src/fichiers-1.tar.gz`
4. Décompacter cette archive au moyen de la commande `tar -xvzf fichiers-1.tar.gz`
   Le répertoire td01 est créé automatiquement.
5. Éditer avec emacs[1] le fichier `programmation-par-objets/src/td01/Prenom.java`
6. Compiler sous `emacs` le programme `Prenom.java` ; Pour cela taper
   `M-x compile` [2]
   puis remplacer `make -k` par
   `javac Prenom.java`
7. Analyser le compte-rendu de la compilation ;
8. Dans un shell, vérifier le contenu du répertoire **td01** et exécuter le programme à l'aide de la commande `java Prenom`.

Exercice 2.

1. Éditer avec emacs le fichier `Monnaie.java`
2. Corriger et indenter correctement ce code.
3. Lancer la compilation du programme depuis `emacs` ;
   Traiter les erreurs de syntaxe en utilisant la clé `C-x ‘` [3]
4. Exécuter ce programme et vérifier s'il donne tous les résultats attendus. Corriger eventuellement le code source.
5. Modifier le programme précédent afin qu'une éxécution sans paramètre génère l'affichage
   `Usage: java Monnaie <somme>`

[1] il est préférable de ne pas lancer plusieurs processus emacs en parallèle

[2] Pour otenir M-x (méta x), il faut appuyer successivement sur les touches "Esc" puis x.

[3] pour obtenir le symbole ‘ sur un clavier AZERTY, il faut taper "AltGr"-7.Cette clé permet de se positionner automatiquement dans le fichier contenant la première erreur et sur la ligne de l'erreur. Chaque nouvelle frappe de la clé **C-x ‘** provoque le positionnement sur l'erreur suivante, et ceci jusqu'à ce que toutes les erreurs aient été balayées.

Arguments d'un programme

Exercice 3. Écrire un programme **Combien** qui affiche le nombre d'arguments sur la ligne de commande.
Exemple :

```
> java Combien a xx 546
Le programme a 3 arguments
```

Exercice 4. Écrire un programme Somme qui affiche la somme des arguments (entiers) de la ligne de commande.
Exemple :

```
> java Somme 1 3 5 7
La somme est 16
> java Somme
Usage : Somme <entier> [<entier> ...]
```

Structures de contrôle

Exercice 5. Écrire un programme **PgcdPpcm** qui calcule puis affiche le pgcd et le ppcm de deux entiers en utilisant l'algorithme d'Euclide.
Rappels :

algorithme d'Euclide : a et b sont deux entiers. Tant que le reste de la division de a par b est non nul, a prend la valeur de b, b prend la valeur du reste. Le pgcd est le dernier reste non nul.

le produit de deux entiers est égal au produit de leur pgcd par leur ppcm.

Exercice 6. Écrire un programme `Premiers` comportant les fonctions suivantes

1. Une fonction `premier` qui retourne *true* si le nombre entier passé en paramètre est premier et *false* sinon.
2. Une fonction `premiersEntre` qui affiche tous les nombres premiers compris, au sens large, entre les 2 valeurs entières min et max passées en paramètres.
3. Une fonction `main` qui ait le comportement suivant :
   - dans le cas où l'argument est unique, affiche celui-ci s'il est premier.
   - dans le cas où il y a deux arguments, affiche tous les nombres premiers compris,au sens large, entre ceux-ci.
   - affiche le message d'erreur "`Usage : Premiers <nombre> [<nombre2>]`" dans tous les autres cas.

Tableaux

Exercice 7. Écrire un programme `Tableaux` qui comporte les fonctions suivantes :
– une fonction `minimum` qui retourne le minimum du tableau d'entiers passé en paramètre.
– une fonction `indiceMaxi` qui retourne l'indice du maximum du tableau d'entiers passé en paramètre.
– une fonction `moyenne` qui retourne la moyenne des éléments du tableau d'entier passé en paramètre.
Utiliser ces fonctions sur le tableau d'entiers constitué des arguments du programme ; un message d'erreur sera affiché dans le cas où ceux-ci seraient absents.

Exercice 8. Ajouter au programme précédent (et tester)
– une fonction `opposes` qui retourne un nouveau tableau contenant les opposés des éléments du tableau d'entiers passé en paramètre.
– une fonction `afficheTableau` qui affiche le tableau d'entiers passé en paramètre.