Programmation Orientée Objet
L2S4 2007–2008
hoenen@icps.u-strasbg.fr

Université Louis Pasteur
UFR de math-info
5 février 2008

TD1 : premiers pas

Exercice 1 : *conventions*

1. Quelles sont les différentes étapes à effectuer pour exécuter un programme donné sous forme de code source ?
2. Précisez les extensions des fichiers utilisés/générés et les commandes nécessaires en mode console.
3. Parmis les étapes ci-dessus, lesquelles sont dépendantes de l'architecture ?
4. Donnez les différentes manières de faire un commentaire dans un code source et expliquez leur différences.
5. Quelles sont les conventions de nommage en java : pour les classes, les méthodes, les variables, les constantes ? Et sur les fichiers sources ?

Exercice 2 : *premier programme*

On considère le fichier HelloWorld.java contenant le code source suivant :

```
1. import java.lang.System;
2. public class HelloWorld {
3.     public static void main(String[] args) {
4.         System.out.println("Hello world!");
5.     }
6. }
```

1. Que fait ce programme ? Que renvoit-il ?
2. À quoi sert la première ligne ? Peut-on s'en passer ?
3. Que trouve-t-on dans le tableau `args` ?
4. Modifiez le code pour qu'il affiche en plus la liste des arguments passés à la ligne de commande.

Exercice 3 : *types primitifs*

1. Quels sont les types primitifs existants ?
2. Quel est la différence entre `int` et `Integer` ?
3. Écrire un petit programme qui calcul la moyenne des nombres entiers passés en argument de la ligne de commande.

Exercice 4 : *lancé de dés*

On veut écrire un petit programme qui réalise le lancé d'un certain nombre de dés d'un type donné.

1. Écrivez la classe `De` qui modélise un dé possédant un certain nombre de faces `nFaces` (numérotées de 1 à `nFaces`). Ajoutez un constructeur permettant d'initialiser l'objet.
2. Ajoutez une méthode qui effectue un lancé aléatoire pour un dé et en renvoie le résultat. `Math.random()` renvoie un `double` positif compris entre 0.0 (inclus) et 1.0 (exclus).
3. Écrivez une classe `LanceDe` qui contient une méthode `main` qui affiche le résultat de 2 lancés d'un dés à 6 faces ainsi que la somme de ces résultats.
4. Faites de même mais cette fois le nombre de lancés et le type du dés sont deux arguments passés en ligne de commande.
5. Modifiez légèrement la classe `De` pour que celle-ci permette de différencier les dés normaux et les dés *pipés* (qui font toujours des scores au dessus de la moyenne : par exemple, un dé à 6 faces pipé ne fera que des résultats compris entre 4 et 6). On ne veut pas modifier le constructeur, rajoutez une méthode *pipeDe* qui transforme un dé classique en dé pipé.
6. Enfin, prennez en compte un troisième argument en ligne de commande qui dit si oui ou non le dé dont on veut faire un certain nombre de lancés est pipé ou classique.