Travaux Dirigés de Java n°1

Cours de Java

—IG 2000 - IR 2—

Septembre 2003

Les premiers pas

Dans cette scéance nous faisons nos premiers pas en Java. Les notions de classes prédéfinies, d'exceptions, de tableaux, d'objets, de références sont introduites. Pour prendre de bonnes habitudes, vous écrirez toutes vos classes dans des paquetages dont le nom commence par `fr.umlv.nomdelogin.cequevousvoulez`. Par exemple, vous pouvez mettre les classes que vous écrirez dans le premier td dans le paquetage `fr.umlv.nomdelogin.td1`. On rappelle qu'il est préférable en Java que chaque classe soit définie dans un fichier qui lui est propre. Le nom du fichier doit être le nom de la classe qu'il contient, auquel on ajoute le suffixe ".java". Les noms de classe doivent être constitués de mots acollés dont la première lettre est une majuscule.

▸ **Exercice 1.** *Hello Groland ! Écrire un programme Java qui affiche* `Hello Groland`.

▸ **Exercice 2.** *Écrire un programme qui affiche la longueur de la chaîne de caractères "Bonjour monde". Si votre méthode* `main` *contient plus d'une instruction, débrouillez-vous pour qu'elle n'en contienne plus qu'une. Écrire un programme qui affiche la longueur de chaque paramètre de sa ligne de commande, ains que le nombre de paramètres sur sa ligne de commande. Écrire un programme qui affiche en majuscules les chaînes de caractères qui lui sont passées sur sa ligne de commande.*

▸ **Exercice 3.** *[types primitifs, enveloppes, classes des API, utilisation de méthodes] En Java les chaînes de caractères sont représentées par la classe* `String`*, dont les instances sont non mutables. On souhaite pouvoir convertir des chaînes de caractères, contenant des nombres entiers, en* `int`*. Le faire en utilisant la méthode* `static int Integer.parseInt(String s)`*. Définir une chaîne de caractères contenant un entier, et obtenir sa valeur sous forme d'*`int`*. Déclarer une chaîne de caractères contenant un nombre réel, et obtenir sa valeur sous forme de* `double`.

▸ **Exercice 4.** *Ecrire un programme prenant sur sa ligne de commande une expression mathématique en notation préfixée, de la forme "opérateur nombre nombre", et qui l'évalue.*

▸ **Exercice 5.** *[Tableaux]*

- *Déclarer deux tableaux* `t1` *et* `t2` *d'entiers.*
- *Créer pour* `t1` *et* `t2` *des tableaux de 5 entiers.*
- *Initialiser* `t2` *avec les valeurs* `1,567,23,78,4`.
- *Comment faire la déclaration, la création et l'initialisation en un seule instruction pour un tableau* `t3`.
- *Ecrire une méthode statique pour trouver dans un tableau le plus petit élément.*
- *Le trier par un tri à bulles.*

▸ **Exercice 6.** *Ecrire un programme qui prend une liste d'entiers sur la ligne de commande, les recopie dans un tableau d'entiers, trie le tableau et l'affiche.*

▸ **Exercice 7.** *[Objet, référence, égalité, mutabilité]*

- *Créer un tableau* `t` *de 4 références sur chaînes de caractères. Créer une chaîne de caractères dont le contenu est "Bon". Affecter les 2 premiers éléments du tableau avec cette chaîne et le troisième avec "Bonjour". Créer une nouvelle chaîne dont le contenu est "Bon", affecter le dernier élément du tableau avec cette chaîne. Comparez 2 à 2 les éléments du tableau en utilisant les méthodes* `String.equals()`, `String.compareTo()` *et l'opérateur* `==`*. Que constatez-vous ?*
- *Concaténez* `t[0]` *avec "jour", en utilisant* `String.concat()` *ou l'opérateur* `+`*. Comparez 2 à 2 les éléments du tableau en utilisant les méthodes* `String.equals()`, `String.compareTo()` *et l'opérateur* `==`*. Que constatez-vous ?*

▸ **Exercice 8.** *[Objet, référence, égalité, mutabilité]*

- *Écrire une classe* `Point` *représentant un point dans le plan, et possédant un constructeur* `Point(int x, int y)`*, une méthode opérant une translation, et une méthode de comparaison de signature* `boolean same(Point)`.
- *Créer un point, et le désigner par deux références* `p1` *et* `p2`*. Comparez* `p1` *et* `p2` *avec* `==` *et* `same()`*. Effectuez une translation sur* `p1` *et refaites les comparaisons. Commentez.*
- *Soit le code suivant:* `Point p1 = new Point(1,1); Point p2 = new Point(1,1);`. *Comparez* `p1` *et* `p2` *avec* `==` *et* `same()`*. Que constatez-vous ? On exécute maintenant l'instruction* `p1=p2`*. Que devient l'objet anciennement référencé par* `p1` *?*

▸ **Exercice 9.** *Déclarez un tableau* `t1` *contenant* $2,7$, $5,21$ *et* -7.3*. Recopier* `t1` *dans un autre tableau* `t2`*. Comparer les tableaux, en utilisant* `==`*, et* `equals()`.