

## TP N° 8

### L’instruction `for` – le type `char`

**Ça va mieux en le révisant ...** L’instruction composée `for` permet d’itérer un bloc d’instructions. Elle est notamment utilisée quand le nombre de fois où il faut répéter le bloc est connu du programme avant de commencer à boucler. Le nombre de répétitions  $y$  est contrôlé par un *indice de boucle*. C’est une variable dont le type de donnée doit être à la fois primitif (ce n’est donc pas la référence d’un objet) et de préférence *discret*<sup>1</sup> (entiers, caractères, booléens).

L’entête de l’instruction `for` est composée de trois parties : la première consiste en l’initialisation de l’indice de boucle, éventuellement accompagnée de sa déclaration. La seconde contient la *condition de continuation* qui porte sur cet indice. Enfin, c’est dans la troisième que l’on précise le *pas de progression* de l’indice.

En `JAVA`, une variable de type `char` est destinée à stocker un caractère issu du jeu de caractères `UNICODE`. Un caractère est noté entre apostrophes afin de le distinguer d’un identificateur de variable. Les caractères ou symboles `UNICODE` sont ordonnés. La méthode `charAt(...)` s’applique à une chaîne de caractères de type `String` afin d’en extraire une lettre, sous forme de `char`.

## 1 Itérations simples

**Exercice 1)** Ecrivez une classe `TP8Ex1` permettant à l’utilisateur d’introduire une valeur entière strictement positive  $n$  et qui calcule puis affiche la valeur de :

$$S_n = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2}$$

◇

**Exercice 2)** On se propose d’écrire une classe `TP8Ex2` qui demande à l’utilisateur de saisir une chaîne de caractères puis qui fabrique et affiche la chaîne de caractères obtenue à partir de la première en inversant l’ordre de ses lettres.

1. Commencez par écrire une méthode dont l’en-tête est :

```
static String reverse ( String s )
```

La chaîne qu’elle renvoie est une copie de la chaîne passée en paramètre, où l’ordre des lettres a été inversé.

2. Ajoutez enfin à la classe `TP8Ex2` une méthode `main(...)` pour qu’elle effectue le travail demandé. Elle contiendra bien sûr un appel à la méthode `reverse(...)` précédente. Voici un exemple d’exécution :

Entrez une chaîne de caractères, svp :

**Il faut encore attendre le 31 décembre 2000 pour voir la fin du siècle.**

En renversant l’ordre des lettres, cela donne :

.elcèis ud nif al riov ruop 0002 erbmeccéd 13 el erdnetta erocne tuaf lI

◇

---

1. Les types primitifs *discrets* (entiers, caractères, booléens) sont à opposer aux types primitifs continus (nombres approchés).

## 2 Application au chiffrement

**Exercice 3) Chiffre de César** Nous vous proposons d'implanter une classe servant à chiffrer (ou déchiffrer) un texte, selon une méthode de décalage. Quoique rudimentaire, cette méthode était celle utilisée par Jules César afin de limiter aux seuls destinataires d'un message la compréhension de son contenu.

Cette méthode consiste à opérer un décalage de  $n$  lettres dans l'alphabet... latin, le z ayant comme successeur le a de sorte à rendre l'alphabet circulaire (idem dans le cas des majuscules).

Par exemple si  $n$  vaut 4, a devient e, b devient f... et z devient d.

1. Ecrivez une méthode `chiffrement(...)` comportant deux paramètres : le premier est la chaîne de caractères à chiffrer, le second est l'entier  $n$  utilisé pour le décalage.
2. Munissez votre classe d'une méthode `main(...)` afin de tester la méthode précédente. Vous pourrez choisir à votre convenance de demander l'entier  $n$  à l'utilisateur ou bien de le faire engendrer aléatoirement par l'ordinateur.
3. A présent, placez vous du point de vue du destinataire et construisez une méthode `déchiffre(...)`, à l'entête suivant :

```
static String déchiffre (String messageChiffré, int n)
```

Cette méthode consiste en un seul appel à la méthode précédente, avec l'entier  $n$  changé en  $26-n$ .

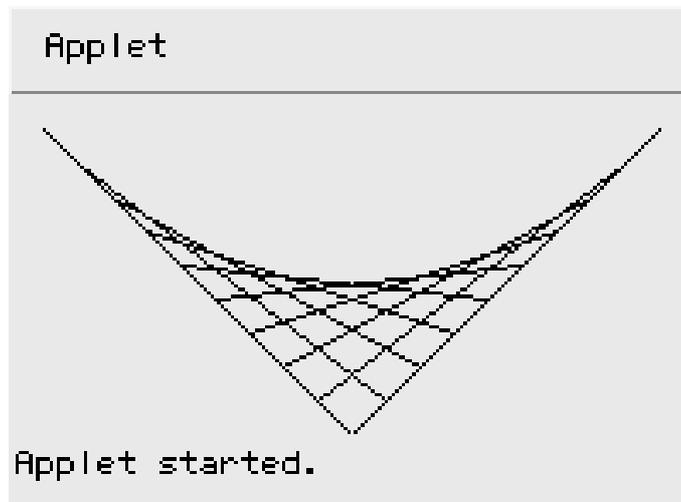
◇

## 3 Approfondissement

**Exercice 4) Fils tendus** On vous propose d'écrire une applet pour faire afficher le dessin itératif ci-dessous.

1. Ecrivez, dans une applet, une méthode pour tracer des lignes entre deux objets de la classe `Point`. Son en-tête sera le suivant :
- ```
static void traceLigne(Graphics g, Point p, Point q)
```
2. Faites tracer la figure à l'aide de l'algorithme suivant :

- p est le point  $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- q est le point  $\begin{pmatrix} 100 \\ 100 \end{pmatrix}$
- faire 10 fois :
  - tracer la ligne entre p et q
  - translater p de (10,10)
  - translater q de (10,-10)



◇