

Travaux pratiques d'informatique N° 9

Le but de cette séance est de vous permettre d'appliquer vos connaissances concernant les chaînes de caractères et la classe **String**.

1. Ecrire une application autonome qui :

- vérifie si l'utilisateur a fourni un nombre compris entre 9 et 13 en tant que **seul** argument de la méthode **main()** et s'arrête prématurément au cas contraire (après avoir affiché un message d'erreur approprié) ;
- demande à l'utilisateur d'introduire une chaîne de caractères formée de maximum **n** caractères, où **n** est le nombre entier introduit "sur la ligne de commande" ;
- vérifie si la chaîne introduite respecte la consigne précisée et demande à l'utilisateur de réintroduire une nouvelle chaîne en cas d'erreur d'introduction des données ;
- affiche à l'écran :
 - la longueur de la chaîne ;
 - la chaîne écrite en ordre inverse ;
 - la chaîne écrite seulement avec des majuscules et, respectivement, des minuscules ;
- demande à l'utilisateur d'introduire un caractère (imprimable) à rechercher et un caractère (imprimable) de remplacement ;
- affiche le nombre d'occurrences du caractère à rechercher (si celui-ci se trouve dans la chaîne), ainsi qu'une nouvelle chaîne dans laquelle le caractère à rechercher est remplacé partout avec le caractère de remplacement.

Indications :

- dans l'environnement de programmation **Eclipse**, afin de fournir des arguments à la méthode **main()** de la classe principale d'un projet, on choisit dans le menu **Run** l'article (item) **Run Configurations...** et, dans l'onglet (x)=**Arguments** de la nouvelle fenêtre flottante **Run Configurations**, on introduit dans la zone de texte **Program arguments**: les paramètres appropriés séparés par des espaces ;
- pour transformer une chaîne de caractères en un entier de type **int**, on peut utiliser la méthode statique **parseInt()** de la classe enveloppe **Integer** ;
- afin d'imposer l'arrêt intempestif de l'exécution du programme par la JVM, on peut utiliser la méthode statique **exit()** de la classe **System**.

Exemple d'affichage après l'exécution du programme (pour $n=11$) :

```

Introduire une chaîne de maximum 11 caractères :
123456789abcd
Introduire une chaîne de maximum 11 caractères :
ABcBB 1Bab
La chaîne introduite a 10 caractères.
La chaîne écrite en ordre inverse :
baB1 BBcBA
La chaîne écrite seulement avec des majuscules est :
ABCBB 1BAB
La chaîne écrite seulement avec des minuscules est :
abcbb 1bab
Introduire un caractère (imprimable) à rechercher :
B
Introduire un caractère (imprimable) de remplacement :
*
Le caractère recherché apparaît 4 fois dans la chaîne !
La nouvelle chaîne après la modification est :
A*c** 1*ab

```

2. Ecrire une application autonome qui :

- demande à l'utilisateur d'introduire une chaîne de caractères formée de deux mots séparés par une et une seule espace (et qui ne commence ni ne finit par des espaces) ;
- vérifie si l'utilisateur a respecté les consignes précisées et lui demande de réintroduire une nouvelle chaîne en cas d'erreur d'introduction des données ;
- construit et affiche à l'écran une nouvelle chaîne formée, dans l'ordre, du deuxième mot, d'une espace et du premier mot.

Exemple d'affichage après l'exécution du programme :

```

Introduire une chaîne formée par deux mots séparés par une seule espace :
trois mots ici
La chaîne introduite ne respecte pas les consignes !

Introduire une chaîne formée par deux mots séparés par une seule espace :
espace début
La chaîne introduite ne respecte pas les consignes !

Introduire une chaîne formée par deux mots séparés par une seule espace :
123 abc
La nouvelle chaîne, après avoir inversé l'ordre des mots, est :
abc 123

```

Remarque : Pour les détails concernant les champs et les méthodes de la classe **String**, veuillez consulter les notes de cours ainsi que la documentation officielle d'ORACLE (l'API Java).