université PARIS DIDEROT PARIS 7

INTRODUCTION À L'INFORMATIQUE ET À LA PROGRAMMATION (IF1)

TP4 : PREMIÈRES BOUCLES

► **Exercice 1** : Écrire une fonction `moutons` qui prend en argument un nombre de moutons et qui compte les moutons à haute voix :

```
$ java Moutons
Combien de moutons ? 3
1 moutons
2 moutons
3 moutons
$
```

La fonction fonctionne-t-elle si l'utilisateur spécifie un seul mouton ou zéro mouton ? Que se passe-t-il si l'utilisateur spécifie un nombre négatif de moutons ?
Modifier cette fonction pour qu'elle affiche un seul mouton au singulier : « o on » plutôt que « o ons ».

► **Exercice 2** : Écrire une fonction `carres` qui renvoie la chaîne de caractères donnant les carrés des dix premiers entiers naturels non nuls, c'est-à-dire donnant la suite d'entiers ... .
Dans la même classe, écrire une fonction `multiples` qui prend un argument entier `n` et renvoie la chaîne de caractères donnant les dix premiers multiples non nuls de `n` : par exemple, la chaîne ... pour `n` .

► **Exercice 3** : Écrire une fonction `horizontal` qui prend un argument entier `n` et renvoie une règle faite avec `n` symboles « » : par exemple, `or on` renvoie la chaîne .
Dans la même classe, écrire une autre fonction `horizontal` qui prend deux arguments entiers et renvoie une règle graduée : par exemple, `or on` renvoie la chaîne .
Toujours dans la même classe, écrire les deux fonctions `vertical` correspondantes.

► **Exercice 4** : Écrire une fonction `estPremier` qui prend un entier en argument et teste si cet entier est un nombre premier. Écrire alors une fonction `premiersNombresPremiers` qui prend un argument entier `n` et renvoie la chaîne de caractères donnant les `n` premiers nombres premiers.

► **Exercice 5** : Dans la lignée de la question 3 de l'exercice **6** du TP2, écrire une nouvelle fonction `entierLu` qui prend en arguments au moins deux messages, demande un entier en insistant (via ces messages) tant que l'utilisateur fournit autre chose qu'un entier, puis finalement récupére l'entier en question.

► **Exercice 6** (Examen Juin 2013) : On considère un jeu où deux adversaires (ici la machine et l'utilisateur) choisissent puis présentent simultanément une figure parmi cinq possibles (codées `'C'`, `'F'`, `'L'`, `'P'`, `'S'`), les règles de supériorité entre ces figures étant : Ciseau coupe Feuille, qui recouvre Pierre, qui casse Ciseau, qui décapite Lézard, qui empoisonne Spock, qui vaporise Pierre, qui écrase Lézard, qui mange Feuille, qui réfute Spock, qui tord Ciseau.

1. Écrire une fonction `alea` qui renvoie le caractère codant une des cinq figures choisie aléatoirement.
2. Écrire une fonction `zuper` qui prend en arguments deux caractères codant des figures et renvoie `+1` si la première figure est supérieure à la seconde, `-1` si elle lui est inférieure et `0` en cas d'égalité.
3. Écrire une fonction `tour` qui choisit une figure `c1` aléatoirement, demande à l'utilisateur de choisir une figure `c2`, affiche la figure `c1` et renvoie `+1` si `c1` est supérieure à `c2`, `-1` si elle lui est inférieure et `0` sinon.
4. Le vainqueur d'un match est le premier qui gagne cinq tours. Écrire une fonction `match`.
   En donner un exemple d'exécution.
5. Proposer une manière de truquer discrètement ce jeu.