

1 Boucles avec accumulateurs

- ▶ **Exercice 1** : Écrire une fonction `puissance` qui prend en argument un réel `a` et un entier `n` supposé positif et renvoie a^n sans se servir de la fonction `Math.pow`. Vérifier que $2,5^3$ vaut 15,625.
 Modifier cette fonction pour traiter le cas où `n` est un entier quelconque. Vérifier que $2,5^{-3}$ vaut 0,064.
- ▶ **Exercice 2** : Écrire une fonction `sommeCarres` qui prend un entier `n` en argument et renvoie la somme des carrés des entiers de 1 à `n`. Vérifier que $1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 = 91$.
- ▶ **Exercice 3** : Écrire une fonction qui calcule la factorielle d'un entier positif. On rappelle que la factorielle d'un entier `n` est le produit des entiers de 1 à `n`.

2 Boucles et entrées multiples

Dans cette partie, l'utilisation de tableaux est proscrite.

- ▶ **Exercice 4** : Écrire un programme qui lit un entier `n`, puis lit `n` nombres réels, et affiche la moyenne de ces nombres. Par exemple :

```
$ java Moyenne
Combien de nombres ? 4
Nombre 1 ? 3
Nombre 2 ? 5
Nombre 3 ? 1
Nombre 4 ? 54
La moyenne est 15.75.
```

- ▶ **Exercice 5** : Écrire un programme qui lit un entier `n`, puis lit `n` nombres et affiche leur maximum.
- ▶ **Exercice 6** : Écrire un programme qui lit un entier `n`, puis lit `n` lignes et affiche la ligne la plus longue. Par exemple :

```
$ java LigneLaPlusLongue
Combien de lignes ? 5
Ligne 1 ? Ceci
Ligne 2 ? est
Ligne 3 ? un
Ligne 4 ? vague essai
Ligne 5 ? ...
Ligne la plus longue : vague essai
```

Pour rappel, la longueur de la chaîne "toto" est obtenue par `"toto".length()`.

3 Boucles imbriquées

- **Exercice 7 :** Écrire une fonction qui prend en argument un entier n , puis affiche avec des étoiles un carré de côté n :

```
$ java Carre
Cote ? 4
****
****
****
****
```

- **Exercice 8 :** Écrire trois fonctions qui prennent chacune en argument un entier n et affichent un triangle de hauteur n , respectivement rectangle avec la pointe en haut à gauche, rectangle avec la pointe en haut à droite et isocèle. Voici des exemples d'exécution d'un programme qui utiliserait ces fonctions :

<pre>\$ java Triangle Hauteur ? 4 Pointe ? -1 * ** *** ****</pre>	<pre>\$ java Triangle Hauteur ? 4 Pointe ? 1 * ** *** ****</pre>	<pre>\$ java Triangle Hauteur ? 4 Pointe ? 0 * *** ***** *****</pre>
---	--	--

4 De vrais graphiques

Télécharger sur [didel](#) le fichier `Drawings.jar` et le mettre dans votre répertoire de travail (pour trouver le fichier, suivre le lien `Documents` et `liens`, puis `Enonces_TP`, puis `tp5`).

Cette archive définit une classe `Drawings` contenant des fonctions qui permettent de faire des graphiques très élémentaires, un lien vers la documentation de cette classe est disponible sur [didel](#).

Pour compiler une classe `MesDessins` qui utilise cette archive, il faut utiliser la ligne de commande :

```
javac -classpath Drawings.jar MesDessins.java
```

- **Exercice 9 :** Écrire une classe `MesDessins` et la compiler. Pour chacune des questions suivantes, il faudra naturellement compiler et tester le résultat.

1. Écrire une fonction `main` qui crée une fenêtre de dessin de largeur 400 et de hauteur 200.
2. Modifier ce `main` pour que l'utilisateur puisse donner les dimensions de la fenêtre en ligne de commande.
3. Ajouter une méthode qui dessine un segment de droite horizontal (trouver les paramètres pertinents) ;
4. Ajouter une méthode qui affiche un segment de droite vertical (trouver les paramètres pertinents) ;
5. Ajouter une méthode qui affiche un rectangle creux (trouver les paramètres pertinents) ;
6. Ajouter une méthode qui affiche un rectangle plein (trouver les paramètres pertinents) ;
7. Ajouter une méthode qui affiche un segment de droite quelconque (trouver les paramètres pertinents) ;
8. Ajouter une méthode qui affiche un triangle creux (trouver les paramètres pertinents) ;
9. Ajouter une méthode qui affiche un triangle plein (trouver les paramètres pertinents).