Java: Exercices de base

Code: act-java-intro

Originaux

url: http://tecfa.unige.ch/guides/tie/html/act-java-intro/act-java-intro.html

url: http://tecfa.unige.ch/guides/tie/pdf/files/act-java-intro.pdf

Auteurs et version

- Daniel K. Schneider - Vivian Synteta

- Version: 1.0 (modifié le 12/4/00 par VS)

Prérequis

- notions de base en programmation

Modules couverts

Module technique: java-intro

Module technique: java-util

Objectifs

- Savoir écrire des simples programmes Java

Exercice 1: Hello World

Le but de cet exercice est d'écrire un simple programme Java qui affiche un message sur une fenêtre "terminal".

La difficulté principale consiste à faire fonctionner votre environnement de développement.

url: départ = http://tecfa.unige.ch/guides/java/staf2x/ex/basics/Hello.java

Etape 1.a: Définissez un nom de classe (par exemple "Salut")

Votre fichier *.java doit avoir le même nom que celui de la classe !

Etape 1.b: Affichez un message de votre choix

Etape 1.c: Mettez des commentaires en français

Exercice 2: Chercher de la documentation dans l'API

Java est un petit langage, mais il contient énorment de librairies. Ici, il s'agit d'avoir un premier contact avec la documentation de l'API

Etape 2.a: Cherchez la classe Frame dans le API

1. Voir http://tecfa.unige.ch/guides/java/jdk/docs/api/

Etape 2.b: Cherchez la méthode paint par défaut pour la classe Frame

Hint: il existe un indexe dans la doc on-line

Etape 2.c: Listez toutes les méthodes de la classe Frame

Pourquoi "paint" ne s'y trouve pas ?

Etape 2.d: Listez les méthodes pour la classe Graphics

Cherchez les méthodes de la méthode paint de l'exemple "Rings"

Exercice 3: Graphisme simple (plus de rings)

Dessinez 3 bagues et changez le texte dans l'exemple Ring

url: http://tecfa.unige.ch/guides/java/staf2x/ex/basics/Ring.java

Etape 3.a: Il faut modifier la méthode paint

```
public void paint (Graphics g) {
   // Draw a yellow ring

   g.setColor (Color.yellow);
   g.drawOval (100,50,50,50);
   .....
   .....
```

Exercice 4: Afficher un titre dans la barre d'un frame

Affichez un titre dans la barre du "Frame" de l'Exercice 3: "Graphisme simple (plus de rings)" [4].

- Une solution est d'utiliser un autre constructeur que Ring()

```
Frame monFrame = new Ring (????);
```

- Une autre solution consiste à améliorer la méthode paint

```
public void paint (Graphics g) {
   // Draw a yellow ring
   g.afficherTitre ("....")
```

- Hint: voir la classe java.awt.Frame et ses méthodes pour trouver ce qu'il faut mettre à la place de "afficher_titre???? (....) ".

Exercice 5: Simple input / output

- Faites un "hello world" interactif qui demande 2 noms: nomX et nomY

Etape 5.a: Lire 2 noms (un après l'autre)

```
// 1. Définir un input buffer
BufferedReader in = new BufferedReader
       (new InputStreamReader(System.in));
// 2. Utiliser:
String xxxx = in.readLine()
String yyyy = .....
```

Etape 5.b: Imprimez Hello nomX + nomY

```
System.out.println("Bonjour " + .... + ......);
```

Exercice 6: Simples opérations arithmétiques

Affichez une somme de quelques nombres flottantes.

Etape 6.a: Définissez et inialisez 3-4 variables flottantes

```
float cash = (float) 1234.50; // par défaut un nombre est double
double cash = 2334.34; // alternative
```

- Ne tentez donc pas de lire ces nombres depuis le terminal (voir l'exercice 9 "Lires des nombres et vérifier l'input" [9])

Etape 6.b: Calculez la somme et la moyenne et affichez le résultat

Exercice 7: Lire 5 plats

1. Demandez à l'utilisateur d'indiquer 5 plats (un après l'autre)
2. A la fin, imprimez le message
   `"Merci, le dernier plat enregistré était: xxxxx"`
   - voici un bout de code:

```
public static void main(String[] args) {
 BufferedReader in = new BufferedReader
         (new InputStreamReader(System.in));
 for (....) {
      String soupe = in.readLine();
 }
 System.out.println ( .....);
```

- Remplacez le nom de variable "soupe" par quelque chose qui est plus approprié !

Exercice 8: Quiz simple

1. Dites à l'utilisateur de deviner cinq mots magiques dans l'ordre
   - Par exemple:
     Capitale de la Suisse ?
     Capitale de la France ?
     .....
2. Vous incrémentez un compteur "réussite" de 1 à chaque fois où la réponse est juste.
3. A la fin, vous lui communiquez le résultat (score).
4. Le score doit être présenté ainsi:

```
************************
*  Score du test = xxx *
************************
```

Exercice 9: Lires des nombres et vérifier l'input

Faites un programme qui lit une suite indéterminée de nombres entiers et qui en calcule la somme et la moyenne

Le programme doit vérifier que les input soient des nombres entiers.

Vous pouvez reprendre l'exercice 6 "Simples opérations arithmétiques" [6]

Etape 9.a: Faites d'abord une boucle de lecture simple

Etape 9.b: Calculs:

- Variables à définir et initialiser avant la boucle:
  - une pour la somme et une autre qui compte les nombres
- Calculez la somme à chaque pas dans la boucle
- Calculez la moyenne à la fin (après la boucle)

Etape 9.c: Vérification de l'input

- Rajoutez un test en utilisant une exception Java

Exercice 10: Lire des nombres et manipuler une simple table

- Variante plus sophistiquée de l'exemple 9 "Lires des nombres et vérifier l'input" [9]

Etape 10.a: Lire des nombres dans un array

- Ecrivez une méthode qui lit des nombres depuis le terminal et qui vérifie l'input
- Ces nombres doivent être stockés dans un array, par exemple
  ```
  float monArray [] = new float[100];
  ```
- Attention: Si vous voulez lire des nombres flottantes il faut utiliser qc comme:
  ```
  monArray[i] = (new Float (CurrentNumber)).floatValue();
  ```

Etape 10.b: Faites vos calculs et affichez un résultat

- Faites une deuxième méthode qui calcule la moyenne et la somme
- Affichez le résultat

Notes:

- Déclarez l'array au niveau de la classe
- Les 2 méthodes sont très simples, du type:
  ```
  public void lecture () { .... }
  ```
- Il faut les appeler depuis la méthode main