Série exercices en programmation procédurale

OBJECTIFS

- Connaître la logique de base de la programmation
- Utilisation:

  - de variables primitives (int, float, char, boolean)
  - des types String & Tableau
  - de la classe String
  - des structures conditionnelles et itératives
  - des sous-programmes (avec méthodes de classe, pas d'instanciation d'objet)

Correction : \Java\Dev\JSE\BaseDuLangage\exoManipVar

1.1 PREMIÈRES MANIPULATIONS DE VARIABLES

Ecrire l' application qui somme 2 valeurs entières, saisies par l'utilisateur ,
choisir le type de variable selon les cas suivants:
1)la valeur mini est -128, la valeur max est + 127 (256 possibilitées)
2) la valeur mini est -32768, la valeur max est +32767(2^16)

A noter le résultat de la somme peut dépasser les tailles limites,
il sera alors retypé implicitement
mettre en évidence le type choisi en utilisant le champ 'SIZE' pour chacun des types

```
taille en bit pour le type byte :8
taille en bit pour le type short :16

ajoute 3 à valMaxTypeByte :-126

ajoute 3 à valMaxTypeShort:-32766
```

(voire cours basedulangage, diapo type primitif/choix d'un type entier)

1.2 Calcul d'un prix

Ecrire un programme invitant à saisir un prix unitaire HT, un taux de TVA et un nombre d'articles, puis qui calcule le montant TTC de l'achat.

```
Calcul d'un prix
saisissez un prix unitaire HT?
100
saisissez un taux de tva?
0,2
montant TTC par article:120.0
saisissez nombre article?
3
montant TTC de l'achat:360.0
```

1.3 Echange de 2 variables

Ecrire le programme réalisant l'échange des valeurs de 2 variables numériques saisies par l'utilisateur

1. En utilisant une 3 eme variable (variable temporaire)
2. En utilisant que 2 variables :

aidez vous des opérateurs + et –

```
sasissez un entier?
4
sasissez un entier?
8
AVANT ECHANGE
val de var1:  4
val de var2: 8
APRES ECHANGE
val de var1: 8
val de var2: 4
```

1.4 Echange de trois variables

Ecrire un programme réalisant la permutation circulaire de 3 variables x, y et z dont les valeurs sont saisies par l'utilisateur.

Cela signifie qu'à la fin du programme:
x contient la valeur initiale de y,
y contient la valeur initiale de z
z contient la valeur initiale de x.

1.5 Conversion de durées (1)

Ecrire un programme demandant une durée en heures, minutes, secondes à l'utilisateur et qui la convertit en secondes. Par exemple 3 heures, 25 minutes et 34 secondes deviendra 12334 secondes.

1.6 Conversion de durées (2)

Ecrire un programme demandant une durée en secondes à l'utilisateur, et qui la convertit en heures, minutes, secondes.

```
saisissez une duree en secondes?65
conversion  heures: 0 minutes: 1 secondes: 5
```

1.7 MA toute première calculatrice

Le programme demande à l'utilisateur de saisir 2 nombres
Les 4 opérations (addition, soustraction, multplication, division) s'effectuent sur les 2 nombres et affiche le résultat correspondant.

```
Veuillez saisir un nombre :
4
Veuillez saisir un autre nombre :
3
Resultat addition: 7
Resultat soustractionn: 1
Resultat multiplication : 12
Resultat division : 1.3333333333333333
```

1.8 Operateur ternaire

Ecrire plus simplement l'instruction suivante :

```
z = (a>b ? a : b) + (a <= b ? a : b) ;
```

1.9 test le signe de la valeur saisie

Le programme demande à l'utilisateur de saisir un nombre n pour afficher:

- -1 si n est négatif,
- 0 si n est nul,
- 1 si n est positif.

1.10 code ASCII

afficher code A.S.C.I.I et code hexa de la lettre a et A
vérifier le résultat avec la table a.s.c.i.i
conseil:utiliser la foncton *toHexString pour le code hexa*

```
caractère = A code A.S.C.I.I: 65 code hexa 41
caractère = a code A.S.C.I.I: 97 code hexa 61
```

2 STRUCTURES CONDITIONNELLES

2.1 déterminer si un nombre saisi est pair ou impair (opérateur %)

```
Saisir un nombre
8
le nbre saisi est pair
```

```
Saisir un nombre
7
le nbre saisi est impair
```

2.2 écrire le programme qui calcule et affiche les solutions d'une équation du premier degré

```
RESOLUTION DE L'EQUATION : ax+b=0


 'a' ?
4
 'b' ?
1
La solution de l'equation est : -0.25
```

2.3 Calcul du montant d'une remise

Un commerçant accorde une remise de 5 % pour tout achat d'un montant compris entre 100 et 500 € et 8 % au-delà. Écrire un programme de calcul du montant de la remise sur un achat donné.

2.4 Savoir si trois entiers sont triés

Ecrire un programme faisant saisir trois entiers **x, y, z** à l'utilisateur, et lui indiquant si ces nombres sont dans l'ordre croissant (x <= y <= z).

2.5 Tri de trois réels

Ecrire un programme faisant saisir trois nombres réels **x, y, z** à l'utilisateur et qui les trie par ordre croissant (à la fin du déroulement du programme **x ≤ y ≤ z**).

2.6 Calculatrice choix de l'opération

Écrire le programme qui demande à l'utilisateur de choisir une opération (addition, soustraction, multiplication, division) puis de fournir 2 nombres entiers le programme calcule et affiche le résultat correspondant.

```
operation (+ - * / ) ?
+
donnez 2 nombres entiers :
3
4
leur somme est :7
```

```
operation (+ - * / ) ?
*
donnez 2 nombres entiers :
3
4
leur produit est :12
```

3 STRUCTURES CONDITIONNELLES ET ITÉRATIVES

3.1 somme d'une suite d'entiers saisie par l'utilisateur

calculer la somme d'une suite d'entiers saisie par l'utilisateur,
la saisie se termine par 0

```
saisissez une val
4
saisissez une val
5
saisissez une val
0
la somme des entiers saisis vaut: 9
```

3.2 Somme des carrés des entiers saisis par l'utilisateur

Ecrire un programme calculant la somme de 1 à n des carrés d'entiers,
n est un entier naturel saisi par l'utilisateur

```
saisissez une val?
3
la somme de 1 à  n des carrés d'entiers saisis vaut: 14
```

3.3 en utilisant une seule variable i et une strucure itérative afficher les valeurs impaires comprises entre 0 et 10

```
1, 3, 5, 7, 9,
```

3.4 determiner la parité des entiers entre 0 et 20 non compris

```
1 est impair
2 est pair
3 est impair
4 est pair
5 est impair
6 est pair
7 est impair
8 est pair
9 est impair
10 est pair
11 est impair
12 est pair
13 est impair
14 est pair
15 est impair
16 est pair
17 est impair
18 est pair
19 est impair
```

3 STRUCTURES CONDITIONNELLES ET ITÉRATIVES

3.5 Suite de nombres pairs

Ecrire un programme affichant une suite de nombres pairs dans une plage déterminée par l'utilisateur. Ce dernier entre au clavier une borne basse, ainsi qu'une borne haute, et le programme affiche dans un ordre croissant les nombres pairs compris dans ce domaine, sans inclure les bornes.

3.6 calculs d'intérêts

Avant de coder, si ce n'est pas fait dessiner l'algorithme.
En combien d'années une montant soumis à un taux annuel de 20% **double**

3.7 Saisir une valeur entière pour afficher la table de multiplication correspondante

```
Saisir une valeur entiere pour afficher la table multiplication correspondante

3
3*0=0
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
```

suite:
Permettre à l'utilisateur de visualiser plusieurs tables de multiplication
l'utilisateur peut sortir du programme à tout moment
conseil: utilisez la méthode 'compareTo'

```
Voulez vous continuez?:(o/n)
o
Saisir une valeur entiere pour afficher la table multiplication correspondante


Voulez vous continuez?:(o/n)
n
merci pour votre participation
```

3.8 affichage de tous les nombres de 10 à 20 sauf le nombre 13.

en utilisant une boucle et l'instruction `continue`;

3 STRUCTURES CONDITIONNELLES ET ITÉRATIVES

3.9 De jolies étoiles !!!

Ecrire un programme demandant à l'utilisateur de saisir un entier strictement positif et réalisant l'affichage ci-dessous :

proposition de progression pour cet exercice :

- utilisez une suite de boucles for non imbriquée
- repérez la répétition du code pour optimiser le code en une boucle for imbriquée dans une autre boucle for

```
saisissez un nb de lignes
 9
*********
********
*******
******
*****
****
***
**
*
```

```
saisissez un nb de lignes
5
*****
****
***
**
*
```

3.10 numerotepage.c

calculer le nombre de fois ou les caracteres 0 à 9 sont utilisés pour numéroter les pages d'un livre de moins de 10 000 pages

```
le nombre de fois ou les caracteres sont utilises est: 38889
```

3 STRUCTURES CONDITIONNELLES ET ITÉRATIVES

3.11 Palindrome Recherche de symétrie dans un mot

Un palindrome est un mot que l'on peut lire
– de droite à gauche
– de gauche à droite.
Par exemple :
A
AA.
38783.
LAVAL.

Ecrire un programme qui détermine si un mot est un palindrome.

```
sasissez un mot?
laval
le mot saisi  est un Palindrome
```

Permettre a l'utilisateur de tester plusieurs mots et de sortir du programme à tout moment

```
saisissez un mot?
38783
le mot saisi  est un Palindrome
Voulez vous continuez?:(o/n)
o
saisissez un mot?
laval
le mot saisi  est un Palindrome
Voulez vous continuez?:(o/n)
o
saisissez un mot?
police
le mot saisi n'est pas un Palindrome
Voulez vous continuez?:(o/n)
n
Au plaisir de vous revoir
```

```
if(strReverse.compareTo(strSaisie) == 0 )
    System.out.println("Palindrome");
```

Permettre à votre programme de tester des phrases:
LAVAL A ETE A LAVAL

3.12 Palindrome sur un chiffre

déterminer combien de nombres positifs de 0 à 1000 ont un carré qui est un palindrome.

Conseil : testez votre programme de 0 à 11 (5 palindromes) avant de généraliser.

```
le nombre de palindrome :15
```

3 STRUCTURES CONDITIONNELLES ET ITÉRATIVES

3.13 Afficher les initiales de votre nom

Write a program that computes your initals from your full name and displays them.

```
String myName = "Fred F. Flintstone";
```

extrait de code: `Character.isUpperCase(myName.charAt(i))`

```
My initials are: FFF
```

4 STRUCTURE DE DONNEE

4.1 nombre de lettres e (minuscules) présentes dans un texte

Ecrire un programme déterminant le nombre de lettres e (minuscules) présentes dans un texte saisi au clavier.

```
saisir du texte?
je saisis du texte
votre texte comporte: 3 fois le caractere: e
```

4.2 nombre de lettres e (minuscules) présentes dans un tableau

Ecrire un programme déterminant si une lettre saisie(minuscules) est présente
dans le tableau suivant:`char tableauCaractere[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};`

```
Saisir une lettre en minuscule, SVP
e
 La lettre e se trouve bien dans le tableau !
```

4.3 Ecrire un programme qui supprime toutes les letres e (minuscules) d'un texte saisi au clavier.

```
saisir du texte?
le texte est la
le texte sans les e:l txt st la
```

4.4 Ecrire un programme qui affiche les code ASCII des lettres et des chiffres sous la forme suivante :

```
caractère = A     code = 65    code hexa = 41
caractère = B     code = 66    code hexa = 42
...
caractère = 1     code = 49    code hexa = 31
...
caractère = 9     code = 57    code hexa = 39
```

4 STRUCTURE DE DONNEE

4.5 Exercice convertir un mot en morse

```
Please enter a word to convert :
PoliCe
Voici le texte saisi convertit en morse:
.--.   ---   .-..   ..   -.-.   .
```

proposition: voire cours 03_NumberString.ptt (diapo 32 → 37)

4.6 Récupération de la valeur maximum dans un tableau

Ecrire le programme pour saisir 5 valeurs de type int qui seront indentifiées par une seule variable puis afficher la plus petite valeur contenu dans ce tableau

```
entrez un entier:
3
entrez un entier:
4
entrez un entier:
5
entrez un entier:
6
entrez un entier:
7
voici les élements contenues dans le tableau:

3
4
5
6
7
La valeur max est :7
```

4.7 Récupération de la valeur maximum dans un tableau dont la dimension du tableau se détermine au moment de l'exécution du programme.

L'utilisateur du programme saisit un nombre variable de valeurs de type int dont la dimension du tableau se détermine au moment de l'exécution du programme.

Les valeurs seront rangées dans une seule variable.

Ecrire le programme pour récupérer ces valeurs et afficher la plus petite valeur contenu dans ce tableau

```
taille du tableau?2
entrez un entier: 3
entrez un entier: 2
voici les élements contenues dans le tableau:
3;2;
La valeur max est :3
```

4 STRUCTURE DE DONNEE

4.8 Ranger les nombres entier saisi par l'utilisateur

dans un tableau pair pour les chiffres pair
dans un tableau impair pour les chiffres impairs (utilisation du modulo 2)
les tailles du tableau sont définies une seule fois par l'auteur de l'application taille 10

```
entrez un entier: 1
entrez un entier: 2
entrez un entier: 3
entrez un entier: 4
entrez un entier: 5
entrez un entier: 6
entrez un entier: 7
entrez un entier: 8
entrez un entier: 9
entrez un entier: 10
tableau d'entier pair:
2;4;6;8;10;
tableau d'entier impair:
1;3;5;7;9;
```

4.9 Gestion de la taille du tableau à l' éxécution du programme

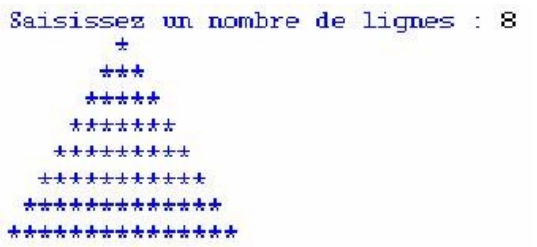
L'utilisateur définit les tailles des tableaux pairs et impairs
le nombre de saisi des val paires et impaires doit correspondre aux tailles de leur tableau respectif.
Les valeurs paires seront rangées dans le tableau pair ,
Les valeurs impaires seront rangées dans le tableau impair.

```
taille de votre tableau pair?
3
taille de votre tableau impair?
4
entrez un entier: 1
entrez un entier: 2
entrez un entier: 3
entrez un entier: 4
entrez un entier: 5
entrez un entier: 6
entrez un entier: 7
le tableau pair:
2;4;6;
le tableau impair:
1;3;5;7;
```

4 STRUCTURE DE DONNEE

4.10 Encore de jolies étoiles !!!

Ecrire un programme demandant à l'utilisateur de saisir un entier strictement positif pour réaliser l'affichage ci-dessous : (**correcton exoPyramideEtoile**)



conseil : utilisez un tableau de string pour manipuler le caractere '*'
un tableau de string dimension 1 suffit ;

4 STRUCTURE DE DONNEE

Les tableaux en 2 D (matrice)

4.11 Ecrire un programme pour à partir du tableau pairimpair afficher sur une ligne les nombres pairs sur l'autre ligne les nombres impairs

```
int pairImpair[][] = { {0,2,4,6,8},{1,3,5,7,9} };
```

```
02468
13579
```

4.12 Somme de matrices (tableau 2D avec des dimensions communes) :

Ecrire le programme calculant la somme de 2 matrices dont les éléments sont de type int et leur dimension est commune)

```
02468
13579
1591317
```

Soit le tableau T de type entier: `int T[] = { 9,4,2,12,42 };`

5.1 Ecrire la fonction Min.

On lui fournit en paramètre le tableau T et sa taille.
Elle retourne la plus petite valeur contenu dans ce tableau

```
la val min est :2
```

5.2 Ecrire une fonction de tri

Ecrire une fonction permettant de trier par ordre croissant les valeurs entières d'un tableau de taille quelconque (taille du tableau et le tableau sont transmis sont fournis lors de l'appel de la fct de tri).
Pour le tri il existe différents algorithmes: **(voire cours AlgorithmesDeTri.ppt)**

proposition

→ **tri a bulle**

→ **tri par sélection**

Différent appel de triTab dans une boucle
-Premier appel de triTab en fournissant à la fonction **triTab** le tableau et sa taille
Dans triTab recherche dans le tableau la plus grande valeur et on la permute avec le dernier élément du tableau
-Second appel de triTab en fournissant à la fonction **triTab** le tableau et sa taille -1
recherche dans le tableau la plus grande valeur et on la permute avec le dernier élément du tableau
-Troisième appel de triTab …....
Et ainsi de suite jusqu'à que la taille du tableau soit 1

5.3 écrire une fonction prenant en paramètre une durée en secondes et qui affiche la conversion de cette durée en heures, minutes, secondes.

```
int main()
{
    conversion(3661);
    return 0;
}
```

```
1 heures  1 minutes 1 secondes
```

5.4 écrire une fonction prenant en paramètre une durée en heures, minutes, secondes et qui retourne le nombre total de secondes correspondant.

5.5 écrire les fonctions cube et max pour déterminer la plus grande valeur entre une valeur et le cube d'une autre valeur fonction imbriquée

```
a = 123; b = 5;
System.out.println("Maximum: "+ Max(a, cube(b)));
//appel de cube le retour de la fct cube est le second argument transmis à Max
```

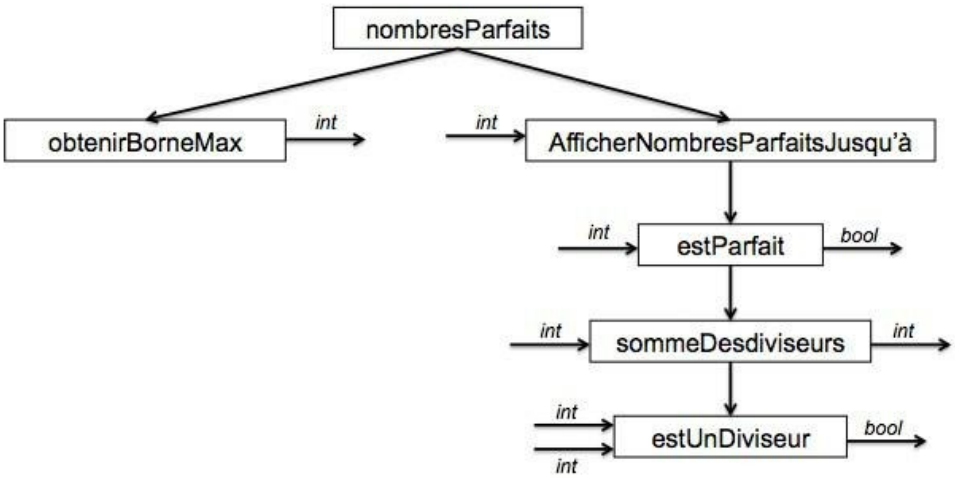
```
125
Maximum:125
```

5 Sous-programmes

5.6 Nombres parfaits **(correction exoNbreParfait)**

Ecrire un programme affichant tous les nombres parfaits jusqu'à un enter **n** saisi par l'utilisateur. Pour rappel, un nombre est parfait s'il est égal à la somme de ses diviseurs stricts (c'est à dire excepté lui même). par exemple, l'entier 6 est parfait car 6 = 1 + 2 + 3

proposition :calculer la somme des diviseurs propres de l'entier n
On découpera le problème en construisant des fonctions selon le plan suivant :



A noter:
Les nombres parfaits sont rares, il n'en existe que trois inférieurs à 1000 qui sont 6, 28 et 496.

```
borne max?
1000
liste des nombres parfait:6;28;496;
```

```
borne max?
10
liste des nombres parfait:6;
```

5.7 Surcharge de méthodes

écrire les fonctions **conversion** prenant en paramètre une durée :

1) en seconde et qui affiche la conversion de cette durée en heures, minutes, secondes.

2) en seconde , minutes et qui affiche la conversion de cette durée en heures, minutes, secondes.

3) en seconde , minutes , heures et qui affiche la conversion de cette durée en secondes.

5.8 Mise en évidence passage de paramètres (d'un type primitif) par valeur

En Java, on ne peut passer les paramètres d'un type primitf (par exemple int ou double) que par valeur, pas par adresse ou référence.
Mise en evidence avec la méthode permut à laquelle le passage des paramètres se fait par valeur.

```
dans le main avant appel fct permut:   A = 10 B = 20
A dans la  fct permut avant permutation : 10
B dans la  fct permut avant permutation :20
A dans la  fct permut apres permutation : 20
B dans la  fct permut apres permutation :10
retour dans le main apres appel fct permut:   A = 10 B = 20
```

Pour un passage par adresse/référence les variables doivent être des objets

```
dans le main avant appel fct permut:   A = 10 B = 20
dans le main apres appel fct permut:   A = 20 B = 10
```

Pour ce dernier exemple nous verrons comment dans le premier exercice sur la P.O.O