

<https://mensuel.framapad.org/p/javasetoulouse?lang=fr>

Bonjour tout le monde !

I - Les bases :

II- Les conditions et les loops

III- Les tableaux

Toute application informatique est composée de 3 couches:

-Couche présentation:

- * client lourd: Swing et Java FX
- * client léger: web - Servlet/JSP - JSF, Spring MVC.....

-Couche métier:

- *Classes Java, web services,

-Couche accès aux données: fichiers, bases de données autres systèmes.....

Environnement JAVA:

- Installer le JDK 1.8

- * Un compilateur
- * JVM
- * un ensemble d'outils contenus dans le dossier BIN
- - Variable d'environnement:
- * poste de travail - propriétés - paramètres système avancés - variables d'envi système -
indiquer le chemin du dossier BIN
- * IDE : Eclipse version Dev
- - pointer vers le dossier BIN: windows - préférences - java - in
- stalled JRE - add -JDK1.8
- - encodage - windows - préférences - enco - workspace - UTF8

-Variables:

zone mémoire permettant de stocker une valeur typée

déclaration : TYPE NOM_DE_La_VARIABLE

-

2 Types:

- * Type primitifs: int, char, float, boolean, long, double, byte, short
- * Type complexe(référence) : Objets qui contiennent des attributs et des méthodes

- Wrapper: pour chaque type primitif un wrapper lui est associé - Une classe offrant des méthodes utilisées de conversion - int - Integer, char - Character et les autres remplacer la première lettre par une majuscule

- Transtypage:

- * implicite: passer d'un type inférieur à un type supérieur ex: int vers float
- * explicite: passer d'un type sup vers un type inf - CAST -

- Constante:

- doivent être initialisées et leurs valeurs ne change pas

ex: final int i = 10;

-Commentaire sur une ligne: //

multiple: /*-----*/

- Méthode: un ensemble d'instructions réutilisable - éviter de répéter du code
visibilité mot-clé type de retour (void) nomMethode(arguments){ instruction; return; }

visibilités:

- public : accessible partout
- private: uniquement à l'intérieur de la classe
- protected : accessible dans la classe et les sous-classes
- default: accès aux classes qui se trouve dans le même package

-package:

représentation logique et physique de l'arborescence du projet qui contient des classes(packages)
dans le d'éviter des conflits de nommage

- Boucles:

-IF-ELSE conditions:

si (conditions) alors instructions sinon instructions

- opérateur ternaire:

int y = (w > 10) ? 20 : 40;

-While - DO-While

-Do-While: s'exécute au moins fois contrairement à la boucle While

- For (int i = 0; i < longueur; i++) { instructions; }

for(int val : tableau){instructions;}

-Switch: remplace les conditions du type if else if else

* default: optionnel et peut se mettre au début, milieu et à la fin de la boucle. Break à la fin de chaque case si on veut sortir de la boucle.

-Tableau 1 et 2 dimensions:

possèdent une taille fixe. int[] tab = new int[3]

int[] tab = { 1, 2, 3};

int[][] tab2D = new int[3][]; la taille du premier tableau est obligatoire.

index commence toujours par ZERO (0)

-Méthode main:

point d'entrée de toute application JAVA. Sa signature ne change pas

public static void main(String[] args){ }

Programmation Objet:

Concepts Objets:

- Encapsulation
- Héritage
- Polymorphisme
- Agrégation
- Abstraction - Interface

