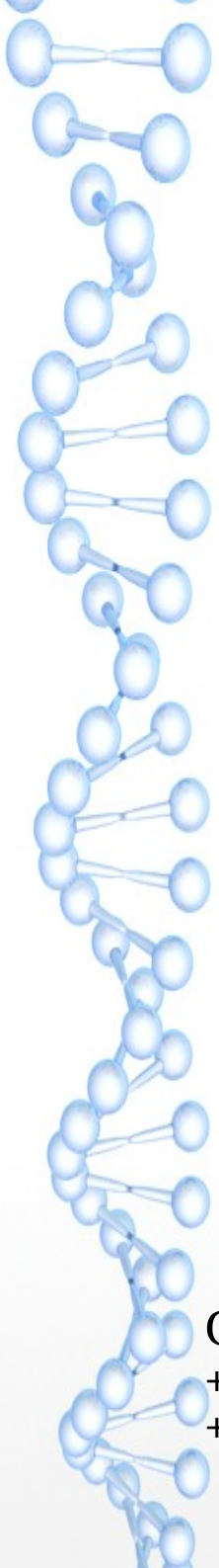


Kabola

Formation Développeur Java SE

Manuel de formation

Ce manuel n'est pas un cours complet

Contact

+ 242 06 602 22 22

+ 242 05 383 45 45

Module 1

Section 1

Kabola

Module 1 : CONCEPTS DE BASES

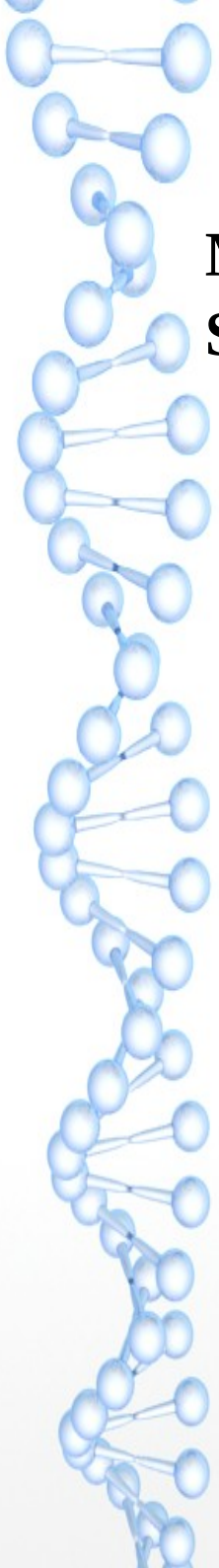
Section 1 : Initiation

Plate-forme Java SE

Java Standard Edition (Java SE) environnement d'exécution et ensemble complet d'API pour des applications de type desktop. Cette plate-forme sert de base en tout ou partie aux autres plateformes

- JDK: Ensemble de librairies fournissant des services de base de la plateforme tels que
  - Le compilateur java
  - La machine virtuelle java
  - Le garbage collector
  - Les librairies de base pour la gestion des entrées-sorties, des threads, interfaces graphiques, mémoire, mécanisme des exceptions, etc.
- JVM: Responsable de l'exécution du bytecode java
- Garbage Collector ou Ramasse miette: Responsable de l'optimisation de la gestion de la mémoire.

Les classes en java (.java) sont compilées par le compilateur java pour produire le bytecode java (.class), qui est ensuite interprété par la JVM qui peut être installée sur n'importe quelle plateforme d'où la portabilité du langage java.



Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Installation JDK

Points pratique…

TP 1 : Installation JDK.

Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Premier programme java

- Ecrire une classe nommé HelloWord.java ayant pour méthode main qui affiche « bonjour à tous ».
- À l'aide de la console
- Taper : **javac HelloWorld.java ,** permet de compiler le code source en bytecode. Cette commande va generée un fichier HelloWorld.class.
- Taper : **java HelloWorld** , permet d'exécuter le byteCode.

La méthode main(), c'est la méthode invoqué par la JVM.

TP 2 : Exécution d'un programme à l'aide de la console.

Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Les règles syntaxiques de base : Java est sensible à la casse. Chaque instruction se termine par un caractère ';' (point virgule). L'indentation est ignorée du compilateur mais elle permet une meilleure compréhension du code par le programmeur.

Les instructions sont contenues dans des blocs de code {}

L'opérateur point . permet de manipuler les membres d'une classe ou d'une instance

Les commentaires : Ils ne sont pas pris en compte par le compilateur donc ils ne sont pas inclus dans le pseudo code. Ils ne se terminent pas par un caractère ";".

Type de commentaire	Exemple
Commentaire abrégé	// Commentaire sur une seule ligne
Commentaire multi ligne	/* */
Commentaire de documentation automatique	/** * */

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Déclaration de variable : Une variable possède un nom, un type et une valeur. La déclaration d'une variable permet de réserver la mémoire pour en stocker la valeur.

Le type d'une variable peut être :

- soit un type élémentaire dit aussi type primitif déclaré sous la forme : type_élémentaire variable;
- soit une classe déclarée sous la forme : classe variable.

Rappel : les noms de variables en Java peuvent commencer par une lettre, par le caractère de soulignement ou par le signe dollar. Le reste du nom peut comporter des lettres ou des nombres mais jamais d'espaces.

Java est un langage à typage rigoureux qui ne possède pas de transtypage automatique lorsque ce transtypage risque de conduire à une perte d'information.

Pour les objets, il est nécessaire en plus de la déclaration de la variable de créer un objet avant de pouvoir l'utiliser. Il faut réserver de la mémoire pour la création d'un objet (remarque : un tableau est un objet en Java) avec l'instruction new. La libération de la mémoire se fait automatiquement grâce au garbage collector.

Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Type élémentaires (Primitifs) : Les types élémentaires commencent tous par une minuscule.

Type	Désignation	Longueur	Valeur
boolean	valeur logique : true ou false	1 bit	True ou false
byte	Octet signé	8 bits	-128 à 127
short	Entier court signé	16 bits	-32768 à 32767
char	Caractère Unicode	16 bits	\u0000 à \uFFFF, soit 0 à 65535
int	Entier signé	32 bits	-2147483648 à 2147483647
float	Virgule flottante simple précision (IEEE754)	32 bits	1.401e-045 à 3.40282e+038
double	virgule flottante double précision (IEEE754)	64 bits	2.22507e-308 à 1.79769e+30

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Type élémentaires (Primitifs) : Les types élémentaires commencent tous par une minuscule.

Type	Désignation	Longueur	Valeur
long	Entier long	64 bits	-9223372036854775808 à 9223372036854775807

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Bloc de code : Le code est contenu dans un bloc de code **{ }**

Un bloc peut contenir d'autres bloc de code

Un Bloc de code :

- Regroupe des instructions
- Limite la portée des variables

Opérateurs arithmétiques : Les opérateurs arithmétiques se notent **+ (addition), - (soustraction), * (multiplication), / (division, attention aux divisions entre entier) et % (reste de la division entière)**

Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Priorités des opérations : Les parenthèses ayant une forte priorité, l'ordre d'interprétation des opérateurs peut être modifié par des parenthèses. (plus prioritaire au moins prioritaire)

Les parenthèses	( )	
les opérateurs d'incrémentation	++ ; - -	Incrément ; décrément
les opérateurs arithmétiques	* ; / ; % ; - ; +	
les opérateurs de décalage	<< ; >>	décalage à g auche, décalage à droit e
es opérateurs de comparaison	< ; > ; <= ; >=	Inférieur ; Supérieur ; inférieur ou égal ; supérieur ou égal
les opérateurs d'égalité	== ; !=	Égalité ; Inégalités
l'opérateur OU exclusif	^	OU EX CL USIF (X OR)
l'opérateur ET	&	et
l'opérateur OU	|	ou
l'opérateur ET logique	&&	ET logique
l'opérateur OU logique	||	OU logique
les opérateurs d'assignement	= ; += ; -=	Affectation ;
Les opérateurs de négations	!	NON

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Autres opérateurs

Opérateurs	Description	
.	point, opérateur d'accès aux membres d'une instance	voiture.rouler();
[ ]	crochet, opérateur de gestion des tableaux	tab[2] = 3;
instanceof	vérifie si l'instance est d'un type particulier	if(d instanceof De)
new	instancie une classe	De d = new De()
-	opérateur unaire, signe opposé	i = -i;
? :	opérateur ternaire	int i = j>0 ? j : -j;

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Structure de contrôles

Les boucles :

```
while ( boolean )
{
        ... // code à exécuter dans la boucle
}
```

Le code est exécuté tant que le booléen est vrai. Si avant l'instruction while, le booléen est faux, alors le code de la boucle ne sera jamais exécuté

```
do {
...
} while ( boolean );
```

Cette boucle est au moins exécutée une fois quelque soit la valeur du booléen;

```
for ( initialisation; condition; modification) {
 ...
}
```

Dans l'initialisation, on peut déclarer une variable qui servira d'index et qui sera dans ce cas locale à la boucle.
Il est possible d'inclure plusieurs traitements dans l'initialisation et la modification de la boucle : chacun des traitements doit être séparé par une virgule.

TP 3 : Remplissage d'un tableau.

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Structure de contrôles

Les branchements conditionnels :

```
if (boolean) {
        ...
} else if (boolean) {
        ...
} else {
        ...
}
```

TP 4 : Calcul de la valeur absolue d'un nombre réel

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Structure de contrôles

Les branchements conditionnels :

```
switch (expression) {
        case constante1 :
        instr11;
        instr12;
        Break;

    case constante2 :
    ...
            default :
                                                    ...
}
```

On ne peut utiliser switch qu'avec des types primitifs d'une taille maximum de 32 bits (byte, short, int, char).

Si une instruction case ne contient pas de break alors les traitements associés au case suivant sont exécutés.

Il est possible d'imbriquer des switch

TP 5 : Affichage des jours de la semaine.

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Structure de contrôles

Les branchements conditionnels :

```
switch (expression) {
        case constante1 :
        instr11;
        instr12;
        Break;

    case constante2 :
    ...
            default :
                                                    ...
}
```

On ne peut utiliser switch qu'avec des types primitifs d'une taille maximum de 32 bits (byte, short, int, char).

Si une instruction case ne contient pas de break alors les traitements associés au case suivant sont exécutés.

Il est possible d'imbriquer des switch

TP 5 : Affiche en fonction des nombre allont de 1 à 7 les jours de la semaine.

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Structure de contrôles

Les débranchements :

break ; : permet de quitter immédiatement une boucle ou un branchement. Utilisable dans tous les contrôles de flot

continue ; : s'utilise dans une boucle pour passer directement à l'itération suivante

return ; : permet de sortir de la boucle courante et de la méthode courante.

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Les tableaux

Ce sont des collections de taille fixe

- le premier élément est indicé à 0

Les éléments peuvent être type primitif ou classe

Les tableaux sont eux-mêmes des objets

- créés avec l'opérateur new
- ont une propriété length

Déclaration d'un tableau de primitifs

- Int[ ] entiers =new int[4];

Déclaration d'un tableau de références sur des objets

- Voiture[ ] voitures = new Voitur[4];

Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Syntaxe java

Les tableaux

Après la création des tableaux les éléments ne sont pas initialisés

- contiennent 0 ou null

Initialisation d'un tableau

- **Int[ ] entiers = new int[3] { 1,2,3 };**
- **Voiture[ ] voitures = new Voiture[2] { new Voiture(),new Voiture() }**

l'instruction new peut être implicite

- **Int[ ] entiers = { 1,2,3 };**
- **Voiture[ ] voitures = { new Voiture(),new Voiture() };**

Tableaux anonymes

- return new String[ ]{"hello","bonjour};

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Prise en main d'un IDE (eclipse)

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate−forme modulaire pour permettre de réaliser des développements informatiques.

Eclipse utilise énormément le concept de modules nommés "plug−ins" dans son architecture. D'ailleurs, hormis le noyau de la plate−forme nommé "Runtime", tout le reste de la plate−forme est développé sous la forme de plug−ins

Points forts d'éclipse

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Une plate−forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug−ins
- Plusieurs versions d'un même plug−in peuvent cohabiter sur une même plate−forme.
- Propose le nécessaire pour développer de nouveaux plug−ins.

Kabola

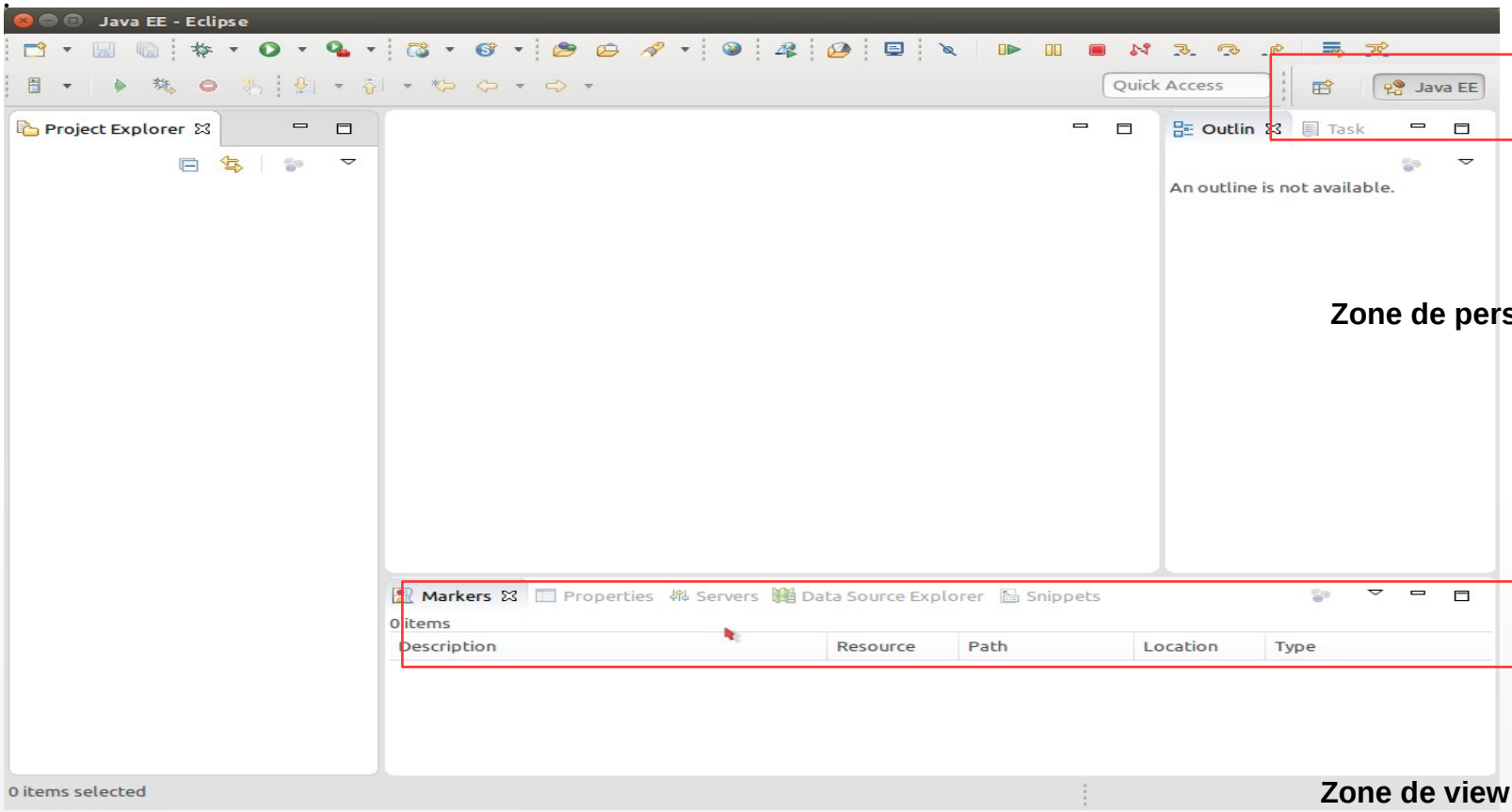
Module 1 : CONCEPTS DE BASES

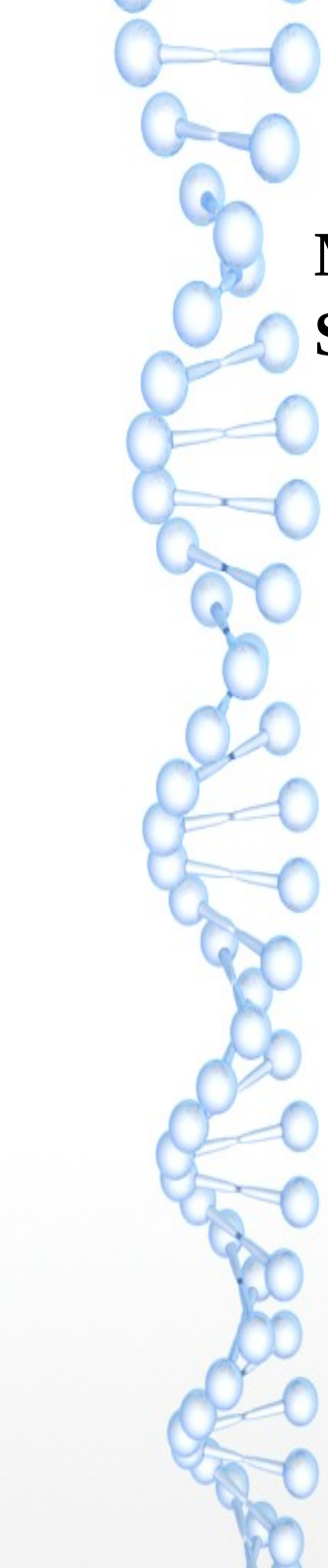
Section 1 : Initiation

Prise en main d'un IDE (eclipse)

Workbench : c'est la seule fenêtre qui s'ouvre au lancement d'éclipse, c'est Le plan de travail

Il est composé de perspectives dont plusieurs peuvent être ouvertes mais une seule est affichée en même temps.





Kabola

Module 1 : CONCEPTS DE BASES

Section 1 : Initiation

Prise en main d'un IDE (eclipse)

Workspace : L'espace de travail est l'entité qui permet de conserver les projets et leur contenu. L'espace de travail contient tous les éléments développés pour le projet : il est possible de créer, de dupliquer, de renommer ou de supprimer des éléments.