Java Cours 4

**Les constructeurs et Variables de classe**

**Les constructeurs :**

Le constructeur par défaut n’a pas de paramètres, des valeurs standards sont attribuées :

* Int → 0
* Booléen → false
* Objet → NULL
* Caractères → NULL

Si on veut créer un constructeur, il faut lui donner le nom de la classe. Un constructeur ne possède pas de retour. Ce constructeur prend alors des paramètres et quand on l’appel pour créer l’objet, il faut donner les paramètres. Ex : Point p = new Point(10, 10) ; Impossible de faire : Point p = new Point() ;

Pour faire ça il faut créer plusieurs constructeurs, le défaut et plusieurs autres. Ils ont tous le même nom, mais ne possèdent pas le même nombre de paramètres.

C’est la surcharge. Il est possible de faire de la surcharge aussi sur les méthodes.

L’utilisation du this, il faut mettre le constructeur le plus utilisé en premier et les autres font références à celui-là par le biais du this.

**Les variables de classe :** *A partir de la diapo 11 (****classe objet 2eme partie****)*

**Diapo 13**

Mise en avant du problème de l’eps différent pour les deux objets que l’on compare. Une fonction égale il faut une symétrie, a=b et b=a. Ici avec des eps qui sont différents on perd la symétrie de l’égalité.

**Diapo 14**

En jaune c’est l’objet-class.

**Diapo 16**

Un membre de classe déclaré static, sera commun à tous les objets créés pour cette classe. Les attributs que l’on met habituellement sont spécifiques à chaque objet créé.

**Diapo 17**

Variable avec static sont des variables de classes, elle est la même pour tous les objets créés par cette classe, l’accès aux variables de classes se fait à l’aide du nom de la classe : « Classe.variableStatic ».

Les variables n’ayant pas le mot clé static sont des variables d’instances, elles sont spécifiques à l’objet qui vient d’être créé. Pour accéder aux variables d’instance, il faut utiliser le mot clé « this » : this.variableInstance ;

**Diapo 20**

Les méthodes get et set utilisent maintenant la classe objet et pas l’objet instancié donc il faut déclarer ces méthodes static pour pouvoir utiliser la variable eps qui est déclarée static. Pour appeler ces méthodes, il faut utiliser le nom de la classe, ex : Point.setEps(xxxx)

**Diapo 21**

Pour créer une constante de classe, on doit combiner les mots-clés static et final. Static fait que cette variable est une variable de classe. Final fait que cette variable ne peut plus être modifiée. Le nom d’une telle variable doit être tout en majuscule.

**Diapo 24**

La main est une methode static

**Diapo 25**

Les classes d’un programme peuvent contenir chacune un main différent. Le main utilisé par la JVM sera celui de la classe donnée à la JVM.

**Diapo 26**

Les variables d’une classe peuvent être initialisées, cela permet d’avoir des valeurs par défaut si le constructeur appelé ne prend pas de paramètre.

**Diapo 27**

On ne peut pas initialiser une variable avec une boucle for, il faut une expression. Pour pouvoir faire l’initialisation de la variable static, on crée un bloc static pour créer les initialisations de la variable.