Notions d'objets en programmation (en Java)

Un objet est un conteneur ou un moule. Nous avons déjà utilisé des objets en JAVA notamment avec String, Scanner, ArrayList/List mais également en HTML avec les balises.

<nom attributs>

Contenu .innerHTML

</nom> .style

Il contient des attributs et des méthodes, les attributs étant des variables et les méthodes des traitements. Donne un peu plus de souplesse et surtout rend le code plus lisible.

Pour définir un objet, on doit créer une classe. Mais par défaut, une classe "n'existe pas" c'est-à-dire que l'on ne peut pas l'utiliser.

Instances de l'objet

Structure de l'objet

Variable du type de ma classe

En Java, on peut écrire:

NomClass.Java

Class NomClasse{

Public type nomvariable;

Public type nomfonction(){

}

Public void procedure(){

}

}

Les publics définissent la portée. On peut aussi écrire privée (private) ou protégée (protected). Lorsque l'on déclare une variable avec une portée publique, on peut à partir d'une instance de l'objet, on peut accéder à l'objet en dehors. Alors qu'en privée, on ne peut pas.

Il faut également écrire les constructeurs qui vont permettre d'initialiser les attributs de l'instance -> IMPORTANT!!

Personne

nom

prenom

dateDeNaissance

telephone

email

Personne.java

public class Personne {

String nom, prenom, telephone, email;

Date dateDeNaissance;

public Personne(){

}

public String toString(){

return this.nom + " " + this.prenom;

}

public static void main (String[] args) {

Personne unePersonne; Création d'une variable de type Personne

unePersonne = new Personne(); Affectation

unePersonne.nom = "PAIRAULT";

unePersonne.prenom = "Marie";

…

Sysout ("Je suis" + unePersonne.nom + " " + unePersonne.prenom);

Sysout ("Je suis " + unePersonne);

}

}

Le **this** permet de définir l'objet de l'instance en cours.

##### Exercice:

Créer un tableau de 7 personnes de la classe et trier le tableau (tri bulle) puis les afficher les unes ou dessus des autres.

**/!\** Lorsque l'on veut afficher une classe (par exemple Personne), le programme affiche la signature de la classe (une sorte d'adresse mémoire) car elle prend en compte l'objet toString de la classe. Si l'on veut éviter cela, il faut créer soi-même sa fonction toString.