Pourquoi la **POO**?

plus de ﬁabilité et de maintenabilité dans les logiciels, notamment grâce à l’encapsulation des données au sein d’un objet et à l’automatisation de l’initialisation avec les constructeurs; + héritage

**Java** est un langage pur objet : un programme sera formé d’une classe ou de la réunion de plusieurs classes et il instanciera des objets.

**La portabilité :** le même bytecode s’exécute de la même façon avec la même précision quel que soit l’environnement. Java assure la gestion mémoire : aucune fuite mémoire possible. Java est sûr : il ne peut pas y avoir d’accès direct à la mémoire non-autorisé + détections des variables non-initialisées et utilisées. Le temps de débogage est relativement court.

**L’encapsulation des données** signiﬁe qu’il n’est pas possible d’agir directement sur les données d’un objet, il faut passer forcément pas les méthodes (≈ fonctions) accessibles de l’objet ; cela diminue les sources de bogue.

**Le code source d’une classe publique DOIT** TOUJOURS se trouver dans un ﬁchier portant le même nom que la classe et possédant l’extension java.

Java refuse les conversions implicites qui peuvent dégrader les données : double vers ﬂoat, ﬂoat vers long et long vers int. Solution : 1.5f

Les classes : tout en minuscule et la première lettre de chaque mot en majuscule.

Les variables, objets et fonctions : tout en minuscule et la première lettre de chaque mot sauf le 1er en majuscule.

Les constantes : tout en majuscule et séparation des mots par \_

Dans System.out.println, System est une classe, out un objet, println une méthode.

**Dans clavier.JAVA :** *public static String lireString () { ... } public static ﬂoat lireFloat () { ... }*

***String prenom = Clavier.lireString() ; int age = Clavier.lireInt() ; String phrase = Clavier.lireString() ;***

**Ou avec scanner :** *java.util.Scanner entree = new java.util.Scanner(System.in) ;*

*System.out.println("Donnez votre prénom et votre nom") ; String prenom = entree.next() ;*

**FINAL : variable qui ne changera pas dans la classe :** *final int NFOIS = 5 ;*

**Utilisation d’une fonction math :** *racx = Math.sqrt(x)*

**(Type.MIN\_VALUE et Type.MAX\_VALUE sont des constantes parfois utiles.)**

**12.f : le f ou F pour obtenir une constante de type ﬂoat. ATTENTION : Float différent de ﬂoat!**

**Il existe une version sans court-circuit du ou | et du et (à revoir)**

**Il n’existe pas de surcharge des opérateurs en Java (à revoir)**

**Déclaration de variables propres à une classe :**

private final int max;

private int val;

\*\*\*\*\*\*\*\*\*\*\*\*\*

public int get\_val() {

return val;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*

public void set\_val(int newVal) {

val = newVal;

}

**Création d’objet « test » de type EntierContraint :**

EntierContraint test = new EntierContraint (1,5,4);

**Utilisation d’une méthode sur l’objet :**

test.incr(1);

**Dans la class test :**

public class TestEntierContraint {

public static void main(String [] args){

et là on rentre les system.out.println/méthodes écécutées/objets créés …