

Tarea 7

Problema 1.

En Java existen variables *primitivas* donde el nombre de la variable evalúa a el valor guardado en la variable. Existen también variables *por referencia* en las cuales el nombre de la variable evalúa a la dirección en memoria donde el objeto al que se hace referencia está guardado. En Java las variables son *primitivas* o *por referencia* pero no pueden ser ambas.

Los programas en C++ tienen cierta flexibilidad al utilizar variables por referencia, sin embargo pueden existir errores que no sean detectables al momento de compilar pero se manifiestan más tarde al momento de ejecutar. Java no soporta apuntadores, a pesar de que una variable por *por referencia* contiene una dirección, o una referencia a una dirección (como lo hace una variable en C++), Java no permite modificar arbitrariamente esa dirección como en el caso de los apuntadores en C++.

En cuanto a tipos de variables existen las siguientes:

Fuente: <http://java.sun.com/>

* Variables de instancia (Campos no estáticos). Técnicamente hablando, los objetos almacenan sus estados individuales en campos no estáticos, esto es, campos declarados sin la palabra `static`. Los campos no estáticos también son conocidos como instance variables por que sus valores son únicos en cada instancia de la clase (en cada objeto en otras palabras).

* Variables de clase (Campos estáticos). Una variable de clase es cualquier campo con el modificador `static`; esto le dice al compilador que hay exactamente una copia de esta variable en existencia , no importa cuantas veces la clase haya sido “instanciada”.

* Variables locales. Este tipo de variables son visibles únicamente por los métodos en los cuales son declarados, no son accesibles por el resto de la clase.

* Parámetros. Lo más importante a recordar es que los parámetros son siempre clasificados como “variables”, no como “campos”.

El siguiente código en Java es para ilustrar las diferencias en cuanto al manejo de los dos lenguajes.

Tarea 7

```
/*File member1.java Copyright 1997, R.G.Baldwin
Illustrates class variables, instance
variables, local variables, and method parameters.
```

Output from this program is:

Class variable is 5

Instance variable is 6

Method parameter is 7

Local variable is 8

```
*****/
class member1 { //define the controlling class
    //declare and initialize class variable
    static int classVariable = 5;
    //declare and initialize instance variable
    int instanceVariable = 6;

    public static void main(String[] args){ //main method
        System.out.println("Class variable is "
                           + classVariable);

        //Instantiate an object of the class to allow for
        // access to instance variable and method.
        member1 obj = new member1();
        System.out.println("Instance variable is "
                           + obj.instanceVariable);
        obj.myMethod(7); //invoke the method

        //declare and initialize a local variable
        int localVariable = 8;
        System.out.println("Local variable is "
                           + localVariable);
    } //end main
    void myMethod(int methodParameter){
        System.out.println("Method parameter is "
                           + methodParameter);
    } //end myMethod
} //End member1 class.
```