



**Facultad de
Ciencias**
UNAM

Trabajo:

Reporte de Práctica 2 de Laboratorio

Asignatura:

Introducción a Ciencias de la Computación

Profesor:

Salvador López Mendoza

Ayudante de Laboratorio:

Ricardo Enrique Pérez Villanueva

Semestre:

2025-1

Fecha de Entrega:

22 de Agosto del 2024

Estudiante:

Valeria Camacho Hernández

Reporte de Práctica 2 de Laboratorio

Palabras Reservadas e Identificadores

1 Asignación

Hacer una lista con los identificadores encontrados en el siguiente programa y expresar que es lo que están representando. No es necesario escribir más de una vez cada identificador

```
public class Identificadores {  
    public static void main(String pps) {  
        float pi = 3.1416;  
        int radio = 15;  
        double area = pi * (radio * radio);  
        String mensaje = "Resultado = ";  
        System.out.println(Mensaje+area);  
    }  
}
```

- Identificadores: Representa el nombre de la clase.
- pps: Es el nombre del parámetro que se está utilizando en el método main.
- pi: Representa el nombre de la variable 'float'.
- radio: Es el nombre de la variable 'int'.
- area: Es el nombre de la variable 'double'.
- Mensaje: Es el nombre de la cadena String.

2 Asignación

Dadas las siguientes declaraciones:

```
int x, y;  
float z = 3.1313f;  
double w = 3.1212;  
boolean verdad = true;
```

Evaluar las siguientes expresiones siempre y cuando sea posible. En caso de no ser posible la evaluación justificar la respuesta.

- | | |
|-------------------|------------------------|
| • $x = 6;$ | • $\text{verdad} = 1;$ |
| • $y = 1000;$ | • $z = 3.1416;$ |
| • $y = 2.33333;$ | • $(x + y) ++;$ |
| • $25 ++;$ | • $y = 1;$ |
| • $w = 175, 000;$ | |

- $x = 6;$ Es correcta, porque es un número entero como se definió como 'int'.

- `y = 1000;` Es correcta, porque es un número entero como se definió como `'int'`.
- `y = 2.33333;` Es incorrecta, porque es un número decimal y no un entero como se definió la variable `'int'`.
- `25++;` Es incorrecta, porque el operador de incremento que se utiliza para aumentar el valor de una variable en 1, solo se puede aplicar a variables, no a valores literales como 25.
- `w = 175, 000;` Es incorrecta, porque no es un tipo de dato decimal.
- `verdad = 1;` Es incorrecta, porque las variables booleanas solo pueden tener el valor `true` o `false`. No se pueden asignar valores numéricos a variables booleanas.
- `z = 3.1416;` Es incorrecta, porque, a pesar de que se declaró la variable como `'float'`, al final no se puso una `f` para indicarlo.
- `(x+y)++;` Es incorrecta, porque la expresión no almacena un valor de manera que pueda ser incrementado. Es el mismo error que el de `25++`.
- `y = 1;` Es correcta, porque es un entero, como se declaró.

3 Asignación

Hacer un programa que imprima en terminal un ejemplo de cada tipo de dato primitivo.

```
public class EjemploPrimitivos {
    public static void main(String[] args) {

        //Tipo numérico:
        int unInt = 771;
        float unFloat = 567.89f;
        double unDouble = 3.4523421;

        //Tipo carácter:
        char unChar = 'M';
        String unString = "¡Hola!";

        //Tipo booleano:
        boolean unBool = true;

        //En terminal
        System.out.println("El valor de un tipo de variable int es: "+unInt);
        System.out.println("El valor de un tipo de variable float es: "+unFloat);
        System.out.println("El valor de un tipo de variable double es: "+unDouble);
        System.out.println("El valor de un tipo de variable char es: "+unChar);
        System.out.println("El valor de una String es: "+unString);
        System.out.println("El valor de un tipo de variable boolean es: "+unBool);

    }
}
```

En terminal se ve lo siguiente:

```
El valor de un tipo de variable int es: 771
El valor de un tipo de variable float es: 567.89
El valor de un tipo de variable double es: 3.4523421
El valor de un tipo de variable char es: M
El valor de una String es: ¡Hola!
El valor de un tipo de variable boolean es: true
```

4 Asignación

Investigar, ¿porque las String (Cadenas) no son un tipo primitivo en Java?

En Java, las cadenas (Strings) no son un tipo primitivo, sino una clase. Esto significa que cada vez que utilizamos un 'String', estamos creando un objeto. A diferencia de los tipos de datos primitivos, un 'String' no tiene una longitud fija, lo que resulta ventajoso, ya que permite añadir funcionalidades sin complicaciones. Simplemente se puede extender la clase correspondiente. Además, los 'String' en Java son inmutables, lo que significa que, una vez creados, su valor no puede ser modificado.

Fuentes:

1. "Cadenas · GitBook". (s. f.). <https://catedu.github.io/poo-java/cadenas.html>
2. "Tutorial Java". (s.f.). http://dis.um.es/~lopezquesada/documentos/IES_1314/IAW/curso/UT3/java/java3/tutorial10.html