

Práctica 7: Programación Orientada a Objetos

Moisés García Villanueva

Noviembre de 2012

1. Clases y Herencias

1.1. Primer programa

Iniciamos con el famoso “Hola Mundo”:

```
class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.print ("Hola mundo, soy el primer programa en Java");
        System.out.println ("di "+args[0]+" "+args[1]);
    }
}
```

Recordar que el nombre del archivo debe ser igual al nombre de la clase con extensión `.java`

Ahora hacemos: `javac HolaMundo.java` y ejecutamos nuestro programa de la forma siguiente:

```
java HolaMundo
```

Veamos ahora un ejemplo de una clase que puede leer números y cadenas de caracteres:

```
class leer{
    public static void main(String[] argv){
        // leer datos desde consola
        int num1;
        System.out.println("Digite un número");
        String linea = System.console().readLine();
        num1 = Integer.parseInt( linea );

        String nombre;
        System.out.println("Digite su nombre");
        nombre = System.console().readLine();

        System.out.println("Tú nombre es: " + nombre + "\nEl número que escribistes es: "+num1);
    }
}
```

Actividades:

1. Hacer un programa en java que obtenga los números impares en el rango que proporcione el usuario.
2. Hacer un programa en java que obtenga la siguiente sumatoria:

$$\sum_{i=0,5}^n \frac{i^2}{n + \pi}$$

1.2. Clases

Al escribir un programa en un lenguaje orientado a objetos tratamos de modelar un problema del mundo real pensando en objetos que forman parte del problema y que se relacionan entre sí. Daremos ahora una primera definición de objeto y clase:

- **Objeto:** entidad existente en la memoria de la computadora que tiene unas propiedades (atributos o datos sobre sí mismo almacenados por el objeto) y unas operaciones disponibles específicas (métodos).
- **Clase:** abstracción que define un tipo de objeto especificando qué propiedades (atributos) y operaciones disponibles va a tener.

**Definición de una clase:
estructura de datos + funciones**

```
class CuentaBancaria {  
    long numero;  
    String titular;  
    long saldo;  
  
    void ingresar (long cantidad) {  
        saldo += cantidad;  
    }  
    void retirar (long cantidad) {  
        if (cantidad > saldo)  
            System.out.println ("Saldo insuficiente");  
        else saldo -= cantidad;  
    }  
}
```

Variables
Métodos

Figura 1: Ejemplo de definición de una clase en java.

Actividades

1. Escribir una nueva clase que haga uso de la clase definida en la figura 1.
2. Escribir una clase para almacenar una imagen en formato ppm
3. Crear una clase que permita dibujar línea, rectángulo, cuadrado y círculo en una imagen ppm.