

Práctica 4

Fundamentos de Programación Trim.16-I

Profra. Graciela Román Alonso

Manejo del Ambiente de Desarrollo Integrado Eclipse para la codificación de programas Java.

1. Abre Eclipse
2. Selecciona un espacio de trabajo (workspace), el Escritorio.
3. Construye un nuevo proyecto: Selecciona File -> New -> Java project, da el nombre de proyecto: Pr4_FP -> Finish
4. Crearemos una clase. Posiciona el mouse sobre el nombre del proyecto y con el botón derecho selecciona -> New -> Class. Luego en la ventana emergente escribe el nombre de la clase en el campo **name**: Prog1_java, marca el cuadrito que indica: **public static void main**. Luego haz click en Finish. Observa el código generado **automáticamente**:

```
public class Prog1_java {  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
    }  
}
```

Si no ves el código, abre el directorio del proyecto, luego abre el directorio src el cual contiene a la clase Prog1_java. Haz doble click sobre el nombre de la Clase y observa el código generado automáticamente.

Ese código es la estructura básica para empezar a hacer un programa Java.

Lo que se encuentra delimitado por `/*` y `*/` se considera un comentario para Java, el compilador ignora todo el texto que esté entre esos dos delimitadores, pueden ser varias líneas. Los símbolos `//` sirven para indicar que el texto que sigue en esa línea es también un comentario, sólo en esa línea.

5. Dentro de las llaves de la clase podemos ver el método especial llamado **main**. **main** es una palabra reservada de Java que se usa para definir un método que contiene al conjunto de instrucciones a ejecutar por la computadora, en orden secuencial. Modifica el programa para añadir una instrucción de escritura en pantalla (usando `System.out.println`), como se muestra a continuación:

```
public class Prog1_java {  
    public static void main(String[] args){  
        System.out.println("Hola, primer programa java");  
    }  
}
```

Para correr el programa en eclipse haz click en el icono superior de la ventana que contiene una flechita blanca dentro de un círculo verde. Observa el resultado.

Agrega dos nuevas instrucciones de escritura para obtener la siguiente salida:

```
"Este es el metodo main"  
"Hola, primer programa java"  
"Adios, terminamos programa java"  
corre el programa.
```

6. Modificaremos el método main para que después de enviar un saludo al usuario le pida que de su nombre, su edad y su promedio de prepa. En Java, las **variables** a utilizar **siempre deben ser declaradas** antes de ser usadas. La declaración de una variable se hace anteponiendo primero su tipo y luego el nombre de la variable, seguido de ";" . Los tipos de datos simples son: int(entero), float(real), double(real), char(caracter), boolean. Un tipo compuesto es String.

Primero vamos a declarar tres variables, **nombre**, **edad** y **calif**, dentro del main:

```
public class Prog1_java {
    public static void main(String[] args){
        String nombre; //cadena de caracteres
        int edad;       // un entero
        float calif;    //un real

        System.out.println("Hola, primer programa java");
    }
}
```

Corre el programa.

Adiciona en el main la declaración de la variable **lectura**, del tipo especial **Scanner**, de la siguiente manera:

```
public class Prog1_java {
    public static void main(String[] args){
        Scanner lectura=new Scanner(System.in); //Scanner para lectura de teclado
        String nombre; //cadena de caracteres
        int edad;       // un entero
        float calif;    //un real

        System.out.println("Hola, primer programa java");
    }
}
```

Esta variable **lectura** nos permitirá leer del teclado varios datos. Si Eclipse subraya en rojo la palabra Scanner, pon el mouse sobre la palabra Scanner sin dar click y aparecerá una ventana que sugiere importar la librería Scanner, da click en 'Import Scanner', y verás que en la parte superior del programa aparecerá lo siguiente: **import java.util.Scanner;**

El código siguiente permite pedir al usuario su nombre, leerlo del teclado y desplegarlo en pantalla:

```
System.out.println("Como te llamas?");
nombre = lectura.nextLine();
System.out.println("Hola "+ nombre);
```

Observa que la operación de asignación en Java se hace usando el símbolo "=", esto es equivalente a usar "<-" en pseudocódigo. En java usamos + para separar cadenas de variables en la escritura.

corre el programa.

Siguiendo el ejemplo anterior, ahora pide al usuario su edad y su promedio invocando a las operaciones para capturar un entero y un real, respectivamente: lectura.nextInt() y lectura.nextFloat(). Despliega la información que el usuario te dio.

Corre el programa.

7. Mostrar uno de los siguientes mensajes al usuario:

Obtuviste una beca!, si su promedio fue mayor o igual que 8.0,

Tienes un descuento!, si su promedio es menor que 8 pero mayor o igual que 6,

Pagas recargo!, si su promedio es menor que 6.

Si usamos "=" para la asignación, entonces ahora se usa "==" como operador relacional para checar si dos datos son iguales. La estructura de control condicional simple y doble en java, es la siguiente:

```
if (condición) //condicional simple
{
    instrucción1;    >>si hay más de una instrucción, si se usan llaves {},
    instrucción2;    si solo es una instrucción no son necesarias las llaves
    instrucciónN;
}
```

```
if (condición) //condicional doble
{
    instrucción1;    >>si hay más de una instrucción, Si se usan llaves {},
    instrucción2;    si solo es una instrucción no son necesarias las llaves
    instrucciónN;
}
else
{
    instrucciónk;    >>si hay más de una instrucción, si se usan llaves {},
    instrucciónr;    si solo es una instrucción no son necesarias las llaves
}
```

corre el programa.

8. Modifica el programa para validar que el número de calificación que de el usuario esté en el rango entre 0 y 10, si no lo está mandar un mensaje de error y pedirle nuevamente su calificación.

La estructura de control iterativa **mientras-haz** en Java, es la siguiente:

```
while(condición)
{
    instrucción1;    >>si hay más de una instrucción, si se usan llaves {},
    instrucción2;    si solo es una instrucción no son necesarias las llaves
    instrucciónN;
}
```

La estructura de control iterativa **haz-mientras** en Java, es la siguiente:

```
do{
    instrucción1;    >>si hay más de una instrucción, si se usan llaves {},
    instrucción2;    si solo es una instrucción no son necesarias las llaves
    instrucciónN;
} while(condición);
```

corre el programa.