

Estructuras de Programación en Java

Objetivo

Comprender y aplicar las estructuras de programación en Java mediante el uso de variables, operadores, estructuras de control condicionales y repetitivas, y bloques de código. Al finalizar esta práctica, los estudiantes serán capaces de desarrollar programas funcionales que utilicen estas técnicas, incluyendo un menú interactivo.

Actividades Prácticas

Actividad 1: Declaración y Uso de Variables

Objetivo: Familiarizarse con la declaración, inicialización y uso de variables.

Instrucciones:

1. Crea una clase llamada `VariablesEjemplo`.
2. Dentro del método `main`, declara y asigna valores a las siguientes variables:
 - Entero (`int`): edad
 - Decimal (`double`): salario
 - Carácter (`char`): inicial
 - Lógico (`boolean`): aprobado
 - Cadena de texto (`String`): nombre
3. Imprime los valores en consola con mensajes descriptivos.

Código sugerido:

```
public class VariablesEjemplo {  
    public static void main(String[] args) {  
        int edad = 25;  
        double salario = 3500.75;  
        char inicial = 'A';  
        boolean aprobado = true;  
        String nombre = "Juan";  
  
        System.out.println("Edad: " + edad);  
        System.out.println("Salario: " + salario);  
        System.out.println("Inicial: " + inicial);  
        System.out.println("Aprobado: " + aprobado);  
        System.out.println("Nombre: " + nombre);  
    }  
}
```

Resultado Esperado: Los valores asignados se imprimen con sus descripciones.

Actividad 2: Operadores Aritméticos, Relacionales y Lógicos

Objetivo: Practicar el uso de operadores para realizar cálculos y evaluaciones.

Instrucciones:

1. Crea una clase llamada `OperadoresEjemplo`.
2. Implementa el siguiente código para realizar:
 - Operaciones aritméticas: suma, resta, multiplicación y división.
 - Comparaciones usando operadores relacionales (`>`, `<`, `==`, `!=`).
 - Condiciones usando operadores lógicos (`&&`, `||`, `!`).

Código sugerido:

```
public class OperadoresEjemplo {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 5;  
  
        // Operadores aritméticos  
        System.out.println("Suma: " + (a + b));  
        System.out.println("Resta: " + (a - b));  
        System.out.println("Multiplicación: " + (a * b));  
        System.out.println("División: " + (a / b));  
  
        // Operadores relacionales  
        System.out.println("a es mayor que b: " + (a > b));  
        System.out.println("a es igual a b: " + (a == b));  
  
        // Operadores lógicos  
        boolean esPositivo = (a > 0 && b > 0);  
        System.out.println("Ambos números son positivos: " + esPositivo);  
    }  
}
```

Resultado Esperado: Los resultados de las operaciones se imprimen en consola con mensajes explicativos.

Actividad 3: Estructuras de Control Condicionales

Objetivo: Usar estructuras de control como `if`, `else if` y `else` para tomar decisiones.

Instrucciones:

1. Crea una clase llamada `ControlCondicional`.
2. Escribe un programa que lea un número ingresado por el usuario y determine si es positivo, negativo o cero.

Código sugerido:

```
import java.util.Scanner;

public class ControlCondicional {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingresa un número: ");
        int numero = scanner.nextInt();

        if (numero > 0) {
            System.out.println("El número es positivo.");
        } else if (numero < 0) {
            System.out.println("El número es negativo.");
        } else {
            System.out.println("El número es cero.");
        }

        scanner.close();
    }
}
```

Resultado Esperado: El programa evalúa el número ingresado y muestra un mensaje correspondiente.

Actividad 4: Estructuras de Control Repetitivas

Objetivo: Practicar bucles como `for`, `while` y `do-while`.

Instrucciones:

1. Crea una clase llamada `BuclesEjemplo`.
2. Escribe el siguiente código para:
 - Imprimir los números del 1 al 10 usando un bucle `for`.
 - Sumar números hasta alcanzar un límite usando un bucle `while`.
 - Mostrar un menú hasta que el usuario elija salir usando un bucle `do-while`.

Código sugerido:

```
import java.util.Scanner;

public class BuclesEjemplo {
    public static void main(String[] args) {
        // Bucle for
        System.out.println("Números del 1 al 10:");
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }

        // Bucle while
        int suma = 0;
```

```

int limite = 50;
int contador = 1;

while (suma < limite) {
    suma += contador;
    contador++;
}
System.out.println("Suma acumulada: " + suma);

// Bucle do-while
Scanner scanner = new Scanner(System.in);
int opcion;

do {
    System.out.println("Menú:");
    System.out.println("1. Opcion 1");
    System.out.println("2. Opcion 2");
    System.out.println("3. Salir");
    System.out.print("Elige una opción: ");
    opcion = scanner.nextInt();
} while (opcion != 3);

scanner.close();
}
}

```

Resultado Esperado: Los bucles funcionan correctamente y cumplen con las instrucciones.

Actividad 5: Creación de un Menú Interactivo con `switch`

Objetivo: Implementar un menú interactivo usando la estructura de control `switch`.

Instrucciones:

1. Crea una clase llamada `MenuInteractivo`.
2. Implementa un menú con opciones:
 - Sumar dos números.
 - Determinar si un número es par o impar.
 - Salir del programa.

Código sugerido:

```

import java.util.Scanner;

public class MenuInteractivo {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int opcion;

        do {

```

```

System.out.println("Menú Principal:");
System.out.println("1. Sumar dos números");
System.out.println("2. Verificar si un número es par o impar");
System.out.println("3. Salir");
System.out.print("Elige una opción: ");
opcion = scanner.nextInt();

switch (opcion) {
    case 1:
        System.out.print("Ingresa el primer número: ");
        int num1 = scanner.nextInt();
        System.out.print("Ingresa el segundo número: ");
        int num2 = scanner.nextInt();
        System.out.println("Resultado: " + (num1 + num2));
        break;
    case 2:
        System.out.print("Ingresa un número: ");
        int numero = scanner.nextInt();
        if (numero % 2 == 0) {
            System.out.println("El número es par.");
        } else {
            System.out.println("El número es impar.");
        }
        break;
    case 3:
        System.out.println("Saliendo del programa...");
        break;
    default:
        System.out.println("Opción no válida. Intenta de nuevo.");
}
} while (opcion != 3);

scanner.close();
}
}

```

Resultado Esperado: El menú permite realizar las operaciones seleccionadas hasta que el usuario elija salir.