PRACTICA 3 "DIVISIBLE"

Introducción

Se proporcionaron dos versiones de un codigo llamado "Divisible" cuyo objetivo es solicitar dos numeros enteros y determinar si el primero es divisible por el segundo. A continuación se analizara cada uno de ellos para determinar la diferencia y utilidad que tienen.

Desarrollo

```
public class Divisible {  --> Nombre de la Clase
    public void Dividir (){  --> Nombre del metodo
        int n, d;  --> Definición y declaración de variables
        Scanner entrada = new Scanner (System.in);  --> Creacion de objeto para la libreria Scanner
        System.out.println ("Introduzca dos enteros");  --> Impresión de instrucciones
        n = entrada.nextInt();
        d = entrada.nextInt();  } Asignación de valores a variables
        if (n %d == 0){  --> Condición de si n es divisible por d
            System.out.println (n + " es divisible por " + d);  --> Impresión de resultados
        }
    }
    public static void main (String [] args){  --> Metodo main
        Divisible div = new Divisible();  --> Objeto para la clase Divisible
        div.Dividir();  --> Llamada al metodo Dividir
    }
}
```

En esta version del codigo al ingresar los dos números vemos que solo obtenemos una salida si resulta ser que el numero n si es divisible por el número d de lo contrario no muestra nada y termina la ejecución sin mostrar ningun resultado.

La otra version del codigo seria:

```
pmport  juva. util. Scanner;  → Importación de la libreria Scanner

public class Divisible 2 {  → Nombre de la clase
  public void Dividir 2 {  → Nombre del metodo
    int n, d;
    Scanner entrada = new Scanner (System.in);
    System.out.print( Introduzca el primer valor  → Lectura del primer valor
    n= entrada. nextInt();
    System. out. print ("Introduzca segunda valor");  → Lectura del segundo valor
    d= entrada. nextInt();

    if (n%d ==0) {
      system.out.println (n+" es divisible entre " +d);  → Instruccion si la condicion es v.
    } else {
      System.out.println (n+" no es divisible entre " +d);  → Instrucción si la condición es F.
    }
  }

  public static void main (String [] a) {  → Metodo main.
    Divisible 2 div2 = new Divisible2();
    div2. Dividir 2();
  }
}
```

En este codigo se puede observar que se agrego un mensaje para leer cada variable haciendo mas facil al oscuria su utilizacion. Tambien se agrego una instruccion extra en el if en caso de que no se cumpla la condicion deseada dando una salida para cada resultado posible.

Conclusion

Como conclusion se puede apreciar que aunque ambos codigos cumplen con la misma funcion, el segundo resulta ser mas eficiente ya que al incluir mas elementos facilita su comprension a la hora de implementarlo.

PRACTICA: MAYOR NÚMERO

Introducción

Se proporciona el programa "MayorNumero" el cual recibia dos valores de entrada y determinaba cual de los dos valores era el número mayor.

Desarrollo

```
import java.util.Scanner;                    → Importación de la libreria Scanner

public class MayorNumero {                   → Nombre de la clase
    public void NumeroMayor(){               → Nombre del metodo
        int n1, n2;                          → Definición y declaración de variables
        Scanner entrada = new Scanner(System.in);   → Objeto para la libreria
        System.out.print("Introduzca primer entero");  } Lectura del primer valor
        n1 = entrada.nextInt();
        // solicitud del segundo valor
        if (n1 > n2){
            System.println(" El mayor es " + n1);   → Instrucción en caso de que la condición se verdadera
        } else {
            System.println(" El mayor es " + n2);   → Instrucción en caso de que la condición sea falsa.
            }
    }

    public void main (String [] args){
        MayorNumero NumMay = new MayorNumero();   }
        NumMay.NumeroMayor();
    }
}
```

Conclusión

Con el desarrollo de esta practica se pudo apreciar una de los usos que se le puede dar a la estructura if-else asi como su correcta implementación y los errores que puede llegar a presentar ya que en este caso aunque ofrecen dos posibles respuestas, no hay una en caso de que ambos números sean iguales.

PRACTICA5 "NOTA APROBADO"

Introducion

Se proporciono el programa "NotaAprobada" el cual solicita digitar una calificación y si esta es menor a 5 se considera como aprovada o no.

Desarrollo

```
import java.util.Scanner; → Importe de la libreria Scanner
public class NotaAprobada { → Nombre de la clase
    public void Aprobado(){ → Nombre del metodo
        int nota; → Definicion y declaracion de la variable
        Scanner entrada = new Scanner (System.in); → Objeto libreria Scanner
        System.out.println("Introduzca nota a analizar"); } Lectura del dato
        nota = entrada.nextInt();
        if (nota >5){ → Condicion
            System.out.println("Prueba Superada"); → Instrucion si la condicion es verdadera.
        }
}
public static void main (String[] args){
    NotaAprobada Nota = new NotaAprobada(); } Metodo main.
    Nota.Aprobado();
}
}
```

Conclusion

Al realizar esta practica puedo concluir que aunque el codigo cumple con su función, no es del todo eficaz ya que en caso de que la nota no sea aprobatoria no muestra ningun mensaje indicando esto, terminando así su ejecución lo que no lo hace lo suficientemente funcional para que el usuario lo comprenda.

PRACTICA6 POSITIVO-SIGNONUMERO

Introducción

Se proporcionaron dos programas de nombres "Positivo" y "Signonumero". El primero tenia como objetivo determinar si el número ingresado es mayor a 0 es decir si es positivo, El segundo solicitaba introducir un número real y determinaba si es mayor, menor o igual a 0.

Desarrollo

```
public class Numero Signo(){ → Nombre de la clase
  float numero; → variables
  Scanner entrada = new Scanner (System.in); → objeto Scanner
  System.out.println("Introduzca un numero real"); } Lectura de datos
  numero = entrada.nextFloat();
  if(numero >0) { → Condicion
    System.out.println(numero + " es mayor que cero"); → Instruccion si la condicion es verdadera
  }

  public static void main (String [] args){ } Metodo main.
    Positivo Pos = new Positivo();
    Pos.NumPositivo();
  }
}
```

Podemos observar que en este codigo se hace la comparacion del valor de la variable con 0 para determinar si este es mayor, sin embargo, en caso de ser menor no hay ninguna instruccion que lo indique lo que hace que el codigo no sea muy eficiente para que el usuario lo comprenda.

El siguiente codigo se representaria como:

```
public class SignoNumero{ --> Nombre de la clase
    public void NumeroSigno(){ --> Nombre del metodo
        float numero;
        Scanner entrada = new Scanner (System.in);
        numero = entrada.nextFloat();
        if (numero >0){ --> Si es positivo
            System.out.println(numero + " es mayor que cero");
        }
        if (numero< 0){ --> Si es negativo
            System.out.println(numero + " es menor que cero");
        }
        if (numero ==0){ --> Si es cero
            System.out.println (numero + "es igual que cero");
        }
    }
    public static void main(String args[]){
        SignoNumero SigNum= new SignoNumero();
        SigNum.NumeroSigno();
    }
}
```

Comparaciones con 0 para la variable.

En este codigo se estructuro de mejor forma ya que hace diferentes comparaciones para cada situación teniendo asi diferentes salidas y dando siempre un resultado

Conclusión

En ambos programas se mostro el uso de la estructura if y aunque el segundo es más eficiente que el primero, ambos aun se podrían mejorar utilizando la estructura if-else permitiendo optimizar aun más los programas y usando menos lineas de codigo.