

Primera Actividad estructura de datos

Joel moreno
Juan González

Fundación universitaria compensar
Ingeniería de software
Estructura de datos

Resumen

En este documento encontraremos la actividad realizada punto por punto con su respectiva solución.

1. Realizar el mapa mental con los conceptos vistos de la unidad de reconocimiento donde se identifique los usos , las diferencias y la importancia que tienen las sentencias de control del flujo de datos.

Mapa mental en pdf aparte

2. . Aplicando las sentencias de decisión y de interacción vistas anteriormente, desarrolle un proyecto en JAVA para: A) Elaborar un programa que calcule el área y el perímetro de 4 figuras geométricas, analice cuales son las entradas, procesos y salidas. B) Elabore un programa que lea un número n y determine si el número es par o impar, escribir la tabla de multiplicación del número leído y adicionalmente se debe preguntar si, se desea analizar otro número o si desea terminar. C) Elabore un menú con 6 opciones (4 figuras geométricas, Tabla del número n, e incluir la opción para terminar)

Solución en el siguiente código

```
package com.mycompany.trabajo1;

import java.util.Scanner;
/**
 *
 *
 */
public class Trabajo1 {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int n = 1;
        do {
            System.out.println("1. calcula area y perimetro del Rectangulo");
            System.out.println("2. calcula area y perimetro del Cuadrado");
            System.out.println("3. calcula area y perimetro del Triangulo");
            System.out.println("4. calcula area y perimetro del Trapecio");
            System.out.println("5. calcula la tabla de multiplicar de un numero");
            System.out.println("6. terminar el proceso");
```



```

        double area = (Math.sqrt(3) / 4)*(lado*lado);
        double p = lado * 3;
        System.out.println("area del Triangulo Equilatero: " + area);
        System.out.println("perimetro del Triangulo Equilatero: " + p);
    }
    case 3 -> {
        System.out.print("Ingresa la longitud del lado 1: ");
        double lado1 = teclado.nextDouble();
        System.out.print("Ingresa la longitud del lado 2: ");
        double lado2 = teclado.nextDouble();
        System.out.print("Ingresa la longitud del lado 3: ");
        double lado3 = teclado.nextDouble();
        double perimetro = lado1 + lado2 + lado3;
        double semip = perimetro/ 2;
        double area = Math.sqrt(semip * (semip - lado1) * (semip - lado2) * (semip -
lado3));

        System.out.println("area de un Isosceles o Escaleno: " + area);
        System.out.println("perimetro de un Isosceles o Escaleno: " + perimetro);
    }
    default -> {
        System.out.println("la opcion que ingresaste es invalida\n");
    }
}
t = t - 1;
} while (t != 0);
}
case 4 -> {
    System.out.print("Ingresa la longitud de la base mayor: ");
    double baseMayor = teclado.nextDouble();
    System.out.print("Ingresa la longitud de la base menor: ");
    double baseMenor = teclado.nextDouble();
    System.out.print("Ingresa la altura: ");
    double altura = teclado.nextDouble();
    System.out.print("Ingresa la longitud de uno de los lados: ");
    double lado1 = teclado.nextDouble();
    System.out.print("Ingresa la longitud del otro lado: ");
    double lado2 = teclado.nextDouble();
    double area = ((baseMayor + baseMenor) / 2) * altura;
    double perimetro = baseMayor + baseMenor + lado1 + lado2;
    System.out.println("El area del trapecio es: " + area);
    System.out.println("El perimetro del trapecio es: " + perimetro);
}
case 5 -> {
    System.out.print("Ingresa el numero del cual deseas saber la tabla: ");
    double num = teclado.nextDouble();

```

```

        if (num % 2 == 0) {
            System.out.println("El numero es par.");
        } else {
            System.out.println("El numero es impar.");
        }
        for (int i = 1; i <= 10; i++) {
            int resultado = (int) (num * i);
            System.out.println(num + " x " + i + " = " + resultado);
        }
    }
    case 6 -> {
        System.out.println("terminaste el proceso");
        n = n - 1;
    }
    default -> {
        System.out.println("la opcion que ingresaste es invalida\n");
    }
}
} while (n != 0);
System.out.println("Fin del programa.");
}
}

```


ELSE

Sentencia else: La sentencia else se asocia directamente con la sentencia if. Cuando la condición de la sentencia if resulta ser falsa, el bloque de código dentro de la sentencia else se ejecutará en su lugar. Esto proporciona una alternativa o un "plan B" en caso de que la condición original no se cumpla

SWITCH

ayuda a seleccionar un camino de ejecución entre varias opciones. Su importancia radica en simplificar el manejo de múltiples condiciones y mejorar la legibilidad del código.

SENTENCIAS DE CONTROL DE FLUJO DE DATOS EN ESTRUCTURAS DE DATOS

SENTENCIA IF

La sentencia if se utiliza para evaluar una expresión o una condición lógica. Si esta condición se evalúa como verdadera (es decir, su valor es true), el bloque de código dentro del if se ejecutará. Si la condición es falsa (valor false), el bloque de código se omitirá y no se ejecutará.

CICLO WHILE Y DO-WHILE:

El ciclo **while** evalúa una condición antes de ejecutar el bloque de código. Si la condición es verdadera, se ejecuta el bloque.

-El ciclo **do-while** ejecuta el bloque de código al menos una vez antes de evaluar la condición. Luego, si la condición es verdadera, se repite.

Importancia: Estos ciclos son útiles para iterar sobre una estructura de datos mientras se cumpla una condición específica. El do-while asegura al menos una ejecución

CICLO "FOR"

-El ciclo for combina la inicialización, la condición y la actualización en una sola línea

Importancia: Perfecto para iterar una cantidad específica de veces. Control preciso sobre la variable de control y los límites de iteración.



IMPORTANCIA DE LAS SENTENCIAS DE CONTROL DE FLUJO

La sentencia if se utiliza para evaluar una expresión o una condición lógica. Si esta condición se evalúa como verdadera (es decir, su valor es true), el bloque de código dentro del if se ejecutará. Si la condición es falsa (valor false), el bloque de código se omitirá y no se ejecutará. Permite recorrer estructuras como listas, arreglos y árboles para realizar operaciones en cada elemento.

CREATED FOR

BARUC MORENO
JUAN GONZALEZ

ESTUCTURA DE DATOS 2023