

GUIA

Facultad de ciencias de la computación

Ing. En ciencias de la computación



Irvyn xicale cabrera

201963582

1. Las clases abstractas pueden instanciar y tener clases hijas: Falso o Verdadero, explique.

2. Una clase final puede heredar de una clase abstracta: Falso o Verdadero, explique

3. Suponga que se tiene la siguiente definición: int B]=new int(5) es correcta SI o NO, explique y corrija el error int [B] = new int [5]

4. Para que las clases hereden características de otra clase se usa la palabra \_\_\_\_\_extends\_\_\_

5. Para que las interfaces puedan utilizarse con las clases se usa la palabra\_\_\_**\_**implements\_\_\_\_\_

6. Las clases abstractas nos permiten realizar el polimorfismo: Falso o verdadero, explique

7. Las clases abstractas no necesariamente deben tener métodos abstractos: Falso o verdadero,

explique

8. El uso de la palabra super es solo entre clase padre e hija o entre diferentes niveles: Falso o verdadero, explique

9. Suponga que A es una clase abstracta y B hereda de ella. ¿Cuál es la forma correcta de crear un objeto que pueda utilizar los métodos públicos de la clase A?

a) A base=new B(); b) A base=new A(); c) B obj=new A(); d) Ninguna es correcta

10. Suponga la siguiente instrucción en Java: s=2; q=++s-2; ¿Cuál es el valor de q despues de ejecutar estas instrucciones?

b) 0 b) 4 c) 1 d) Ninguna es correcta

11. ¿Cuáles son los ocho tipos primitivos en Java? short bit long Float String double Object byte char true false int Integer float boolean string

12. Los tipos de accesos son:\_\_private, default, protected y public\_

13. Los atributos de las clases abstractas deben ser preferentemente\_\_private\_\_\_\_\_\_\_\_\_\_\_

14. Los atributos de las clases hijas deben ser preferentemente\_\_\_\_private, public\_\_\_\_\_\_\_\_\_\_\_

15. Para que se utilizan los constructores:

a) Crear variables y objetos b) inicializar variables c) a y b d) Para construir

16. La diferencia entre un método static y otro que no lo es:

a) Pertenece a la clase y no al objeto b) Maneja los atributos de la clase c) no hay diferencia

17. Un constructor es un método que puede devolver un tipo de dato mediante la palabra reservada return: :

Falso o verdadero, explique

No, el constructor no devuelve ningún valor.  
Al declarar un constructor, no tendrá nada como el tipo de retorno.  
En general, se llama implícitamente a Constructor en el momento de la instanciación.  
Y no es un método, su único propósito es inicializar las variables de instancia.

18. En una clase se puede tener varios constructores:

Falso o verdadero, explique

El número de constructores que puede contener una clase en Java es ilimitado, siempre y cuando su signatura sea distinta. Es decir, han de ser constructores sobrecargados, que acepten distintos parámetros.

19. Al uso de varios constructores se le llama:\_\_\_sobrecarga\_\_\_\_\_\_\_\_\_\_\_\_\_\_

20. ¿Qué significa que Java tenga recolector de basura?, Explique

Que java reasigna la memoria automáticamente evitando que se desperdicie memoria.

20. ¿Qué significa que Java tenga recolector de basura?, Explique

Que java reasigna la memoria automáticamente evitando que se desperdicie memoria.

21. ¿Qué es la máquina virtual de Java?, Explique

Es el encargado de ejecutar el código compilado.

22. Los archivos java tiene extensión\_\_\_java\_\_\_\_\_\_\_ y cuando se compila se genera la extensión\_\_class\_

23. Los archivos class son:\_\_\_archivos java compilados\_\_\_\_\_

24. Un arreglo de objetos es:\_\_\_\_un conjunto de un mismo objeto \_\_\_\_

25. Una matriz en Java se declara:\_\_\_int[ ][ ] matriz= new int[5][2] \_\_\_\_\_\_\_\_\_

26. ¿Qué es el bytecode en Java? Lo podemos encontrar en el archivo .class el cual será procesado por la maquina virtual para ser compilado en lenguaje maquina.

27. ¿Qué significa sobrecargar (overload) un método?

Es el tener varios métodos con el mismo nombre y diferenciarlos por los parámetros de estos

28. Cuales son la formas de leer datos en java, explique mediante un ejemplo

Se puede leer con scanner

Scanner leer= new Scanner(System.in);

PruebaString=leer.nextLine();

PruebaInt=leer.nextInt();

PruebaDouble=leer.nextDouble();

Y con bufer

InputStreamReader entrada=new InputStreamReader (System.in);

BufferedReader flujoEntrada=new BufferedReader(entrada);

try

    {

    datoEntrada = flujoEntrada.readLine();

    Marca=datoEntrada;

datoEntrada = flujoEntrada.readLine();

    amo=Integer.parseInt(datoEntrada);

}

    catch(IOException error)

    {

    System.err.println("Error " + error.getMessage());

    }

29. Diferencia entre las clases Vector() y ArrayList()

Varios subprocesos podrían operar en ArrayList al mismo tiempo en el Vector es un proceso a la vez. Aemas que el vector suele ser utilizado principalmente para números.

30. En las interfaces no existen herencia entre ellas, Falso o verdadero, explique

Verdadero, una interfaz si puede heredar de otra interfaz y esta también podría heredar a otra interfaz o clase.

Ejercicios 1.

Escribir un programa que visualice un cuadrado mágico de orden impar n comprendido entre 3 y 11; el usuario debe elegir el valor de n. Un cuadrado mágico se compone de números enteros comprendidos entre 1 y n; las sumas de los números que figuran en cada fila, columna y diagonal son iguales.

Ejemplo:

8 1 6

3 5 7

4 9 2

Un método de generación consiste en situar el número 1 en el centro de la primera fila, el número siguiente en la casilla situada por encima y a la derecha, y así sucesivamente; el cuadrado es cíclico: la línea encima de la primera es, de hecho, la última y la columna a la derecha de la última es la primera. En el caso de que el número generado caiga en una casilla ocupada, se elige la casilla sobre el número que acaba de ser situado. (En caso de dificultad, pueden realizar solo la comprobación de un cuadro mágico)

Diagrama

Descripción generada automáticamente

Ejecuta.java

public class Ejecuta {

    public static void main(String[] args){

        Menu menu = new Menu();

        menu.menu();

    }

}

Menu.java

import java.io.\*;

public class Menu {

    // lectura

    InputStreamReader entrada = new InputStreamReader(System.in);

    BufferedReader flujoEntrada = new BufferedReader(entrada);

    String datoEntrada;

    // atributos

    int op,op2;

    boolean serie = false;

    Numeros numeros;

    CuadroMagico cuadro;

    public void menu(){

        do{

            try{

                System.out.println("\tMENU\n1.-Generar tamaño y seria para el cuadro\n2.-Ver cuadro magico\n3.-Salir");

                datoEntrada = flujoEntrada.readLine();

                op = Integer.parseInt(datoEntrada);

                switch(op){

                    case 1:

                        do{

                            System.out.println("ingresa el tamaño del cuadro (recuerda que el tamaño debe ser impar y del 3 al 11");

                            datoEntrada = flujoEntrada.readLine();

                            op2 = Integer.parseInt(datoEntrada);

                        }while(op2!=3 && op2!=5 && op2!=7 && op2!=9 && op2!=11);

                        numeros = new Numeros(op2);

                        numeros.setNumAleatorios();

                        serie = true;

                    break;

                    case 2:

                        if (serie==true){

                            cuadro = new CuadroMagico(numeros.tamMatriz);

                            cuadro.setCuadro(numeros.getNumAleatorios());

                            cuadro.verCuadro();

                        }else System.out.println("crea una serie primero");

                    break;

                    case 3:

                        System.out.println("gracias por usar este programa");

                    break;

                    default: System.out.println("opcion invalida");

                }

            }catch(Exception e){

                System.out.println("Error " + e.getMessage());

            }

        }while(op!=3);

    }

}

Números.java

public class Numeros {

    // atributaos

    int[] numAleatorios;

    int tamMatriz;

    // metodos

    public Numeros(int tam){

        numAleatorios = new int[tam\*tam];

        tamMatriz = tam;

    }

    public void setNumAleatorios(){

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

        System.out.print("Se genero la serie: ");

        numAleatorios[0]=(int)(Math.random()\*100+1);

        System.out.print(numAleatorios[0]+ " ");

        for(int i=1;i<numAleatorios.length;i++){

            numAleatorios[i]=numAleatorios[i-1]+1;

            System.out.print(numAleatorios[i] + " ");

        }

        System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

    public int[] getNumAleatorios(){

        return numAleatorios;

    }

}

CuadroMagico.java

public class CuadroMagico {

    int[][] cuadro;

    public CuadroMagico(int tam){

        cuadro = new int[tam][tam];

        for(int i=0;i<cuadro.length;i++){

            for(int j=0;j<cuadro[i].length;j++){

                cuadro[i][j] = -1;

            }

        }

    }

    public void setCuadro(int[] nums){

        int x=0,y;

        int previoX,previoy;

        int filas = cuadro.length;

        int columnas = cuadro[0].length;

        y = ((columnas)/2);

        cuadro[x][y]=nums[0];

        for(int i=1; i<filas\*columnas;i++){

            previoX=x;

            previoy=y;

            if(x - 1<0){

                x=filas-1;

            }else x--;

            if(y+1>columnas-1){

                y=0;

            }else y++;

            if(cuadro[x][y]!=-1){

                x=previoX+1;

                y=previoy;

            }

            cuadro[x][y]=nums[i];

        }

    }

    public void verCuadro(){

        int suma,sumaDiagonal=0,sumaDiagonalInvertida=0;

        int[] sumaColumna = new int[cuadro[0].length];

        // funciona para la diagonal invertida

        int k=cuadro[0].length-1;

        System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

        for(int i=0;i<cuadro.length;i++){

            // inicializamos los contadores en 0

            sumaColumna[i]=0;

            suma=0;

            for(int j=0;j<cuadro[0].length;j++){

                // suma es suma de filas

                suma += cuadro[i][j];

                // se suma las columnas

                sumaColumna[i] += cuadro[j][i];

                // se suma la diagonal

                if(i==j){

                    sumaDiagonal += cuadro[i][j];

                }

                System.out.print(cuadro[i][j] + " ");

            }

            // imprime las sumas de las filas

            System.out.println("  la suma de esta fila es = " + suma);

        }

        System.out.println("");

        // imprime las sumas de las columnas

        for(int i=0;i<cuadro[0].length;i++){

            System.out.println("la columna " + (i+1) + " su suma es: " + sumaColumna[i]);

        }

        // realiza la suma de la diagonal invertida

        for(int i=0;i<cuadro[0].length;i++){

                sumaDiagonalInvertida += cuadro[i][k];

                k--;

        }

        System.out.println("la suma de la diagonal es: " + sumaDiagonal);

        System.out.println("la suma de la diagonal Invertida es: " + sumaDiagonalInvertida);

        System.out.println("es un cuadrado magico");

        System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

    }

}

Ejercicio2

Diagrama

Descripción generada automáticamente

Ejectuta.java

public class Ejecuta {

    public static void main(String[] args){

        Menu menu = new Menu();

        menu.menu();

    }

}

Menu.java

import java.io.\*;

public class Menu {

    // lectura

    InputStreamReader entrada = new InputStreamReader(System.in);

    BufferedReader flujoEntrada = new BufferedReader(entrada);

    String datoEntrada;

    // atributos

    int op;

    Distrito distrito = new Distrito();

    // metodos

    public void menu(){

        // espera la opcion

        do{

            do{

                try{

                    System.out.println("\tMENU\n0.-salir\n1.-ingresar candidatos\n2.-Imprimir la tabla\n3.-Calcular e imprimir el numero total de votos recibidos por cada candidato y el porcentaje del total de votos emitidos\n4.-imprimir el nombre de los dos candidatos más votados");

                    datoEntrada = flujoEntrada.readLine();

                    op = Integer.parseInt(datoEntrada);

                }catch(Exception e){

                    System.out.println("Error " + e.getMessage());

                }

            } while(op!=0 && op!=1 && op!=2 && op!=3 && op!=4);

            // empieza el switch segun la opcion

            switch(op) {

                // salir

                case 0:

                    System.out.println("Gracias por usar este programa");

                break;

                // ingresar candidatos

                case 1:

                    distrito.ingresarCandidatos();

                break;

                // Imprimir la tabla

                case 2:

                    distrito.imprimirTabla();

                break;

                // Calcular e imprimir el numero total de votos

                case 3:

                    distrito.votos();

                break;

                // imprimir el nombre de los dos candidatos más votados

                case 4:

                    distrito.muestraMejores();

                break;

                default: System.out.println("opcion invalida");

            }

        }while(op!=0);

    }

}

Distrito.java

import java.io.\*;

public class Distrito{

    // lectura

    InputStreamReader entrada = new InputStreamReader(System.in);

    BufferedReader flujoEntrada = new BufferedReader(entrada);

    String datoEntrada;

    // atributos

    // primera parte de la matriz controla el distrito y la segunda el candidato

    int distrito = 5;

    int candidato = 4;

    int[][] candidatos = new int[distrito][candidato];

    double[] porsentaje = new double[candidato];

    // metodos

    public void ingresarCandidatos(){

        try{

            for(int i=0;i<candidatos.length;i++){

                for(int j=0;j<candidatos[0].length;j++){

                    System.out.print("Ingresa el numero de votos del candidato " + (j+1) + " en el distrito " + (i+1) + ": ");

                    datoEntrada = flujoEntrada.readLine();

                    candidatos[i][j] = Integer.parseInt(datoEntrada);

                }

                System.out.println("\n");

            }

        }catch(Exception e){

            System.out.println("Error " + e.getMessage());

        }

    }

    public void imprimirTabla(){

        System.out.print("\t     ");

        for(int i=0;i<candidatos[0].length;i++){

            System.out.print("Candidato " + (i+1) + "\t");

        }

        System.out.println("");

        for (int i=0; i<candidatos.length;i++){

            System.out.print("distrito " + (i+1) + ":\t     ");

            for(int j=0;j<candidatos[0].length;j++){

                System.out.print(candidatos[i][j] + "\t        ");

            }

            System.out.println("");

        }

    }

    public void votos(){

        double votantes=0;

        // votos totales por candidato

        int[] votosTotales = new int[candidato];

        for (int i=0; i<candidatos[0].length;i++){

            votosTotales[i] = 0;

            for(int j=0;j<candidatos.length;j++){

                votosTotales[i] = votosTotales[i] + candidatos[j][i];

                votantes = votantes + candidatos[j][i];

            }

        }

        for(int i=0;i<candidatos[0].length;i++){

            System.out.print("\t     Candidato " + (i+1));

        }

        System.out.println("");

        System.out.print("Votos: \t       ");

            for(int i=0;i<candidatos[0].length;i++){

                porsentaje[i]=(votosTotales[i]/votantes)\*100;

                System.out.print(votosTotales[i] + " - " + String.format("%.2f", porsentaje[i]) + "%\t       ");

            }

            System.out.println("");

            for(int i=0;i<candidatos[0].length;i++)

                if(porsentaje[i]>50){

                    System.out.println("El candidato " + (i+1) + " acaba de ganar con un " + porsentaje[i] + " de la votacion");

                }

            System.out.println("\n");

    }

    public void muestraMejores(){

        System.out.println("hola");

        double[] ordenar = new double[porsentaje.length];

        double temp;

        for (int i = 0; i < porsentaje.length; i++) {

            ordenar[i] = porsentaje[i];

        }

        for (int i = 0; i < ordenar.length -1; i++) {

            for (int j = ordenar.length - 1; j > i; j--) {

                if (ordenar[j] > ordenar[j - 1]) {

                    temp = ordenar[j];

                    ordenar[j] = ordenar[j - 1];

                    ordenar[j - 1] = temp;

                }

            }

        }

        for (int i = 0; i < 2; i++) {

            for(int j=0;j<ordenar.length;j++){

                if (ordenar[i]==porsentaje[j]){

                    System.out.println("el candidato " + (j+1) + " esta en la posision " + (i+1));

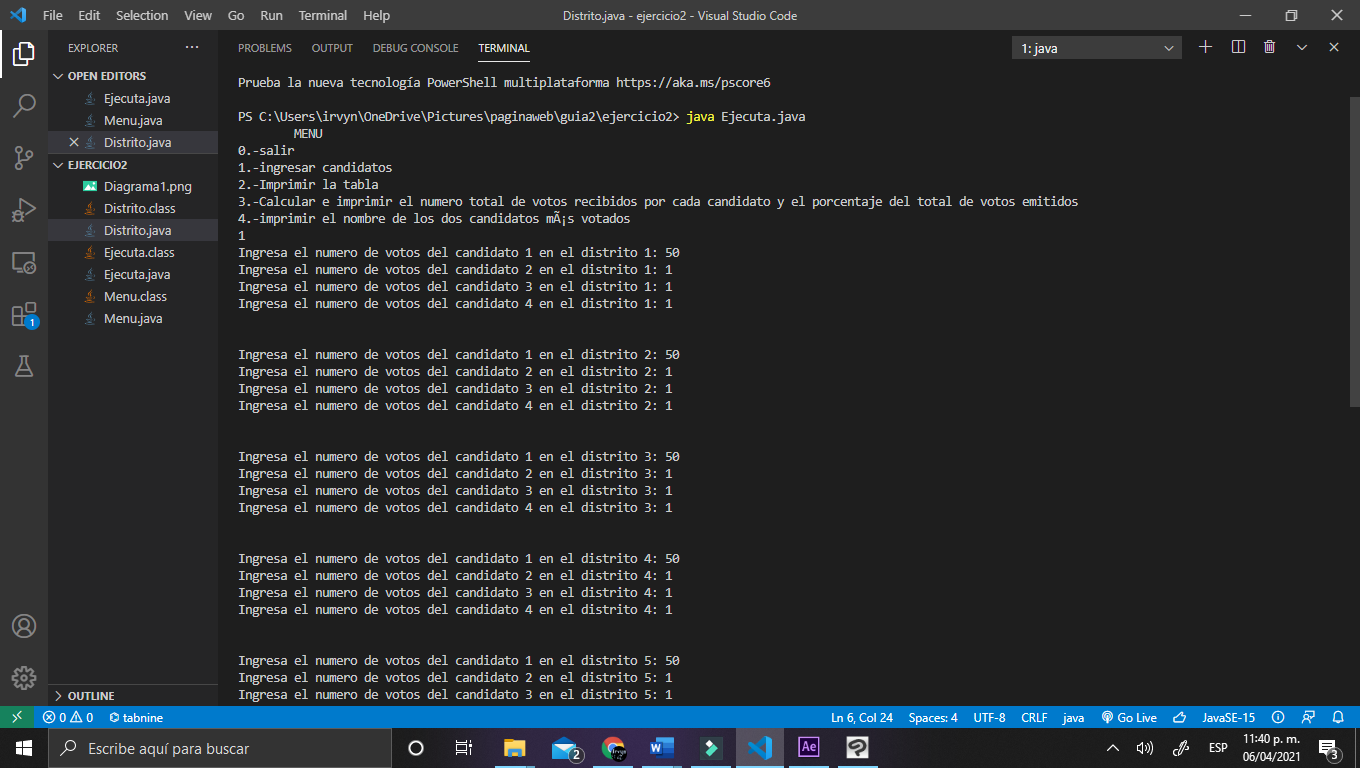
                }

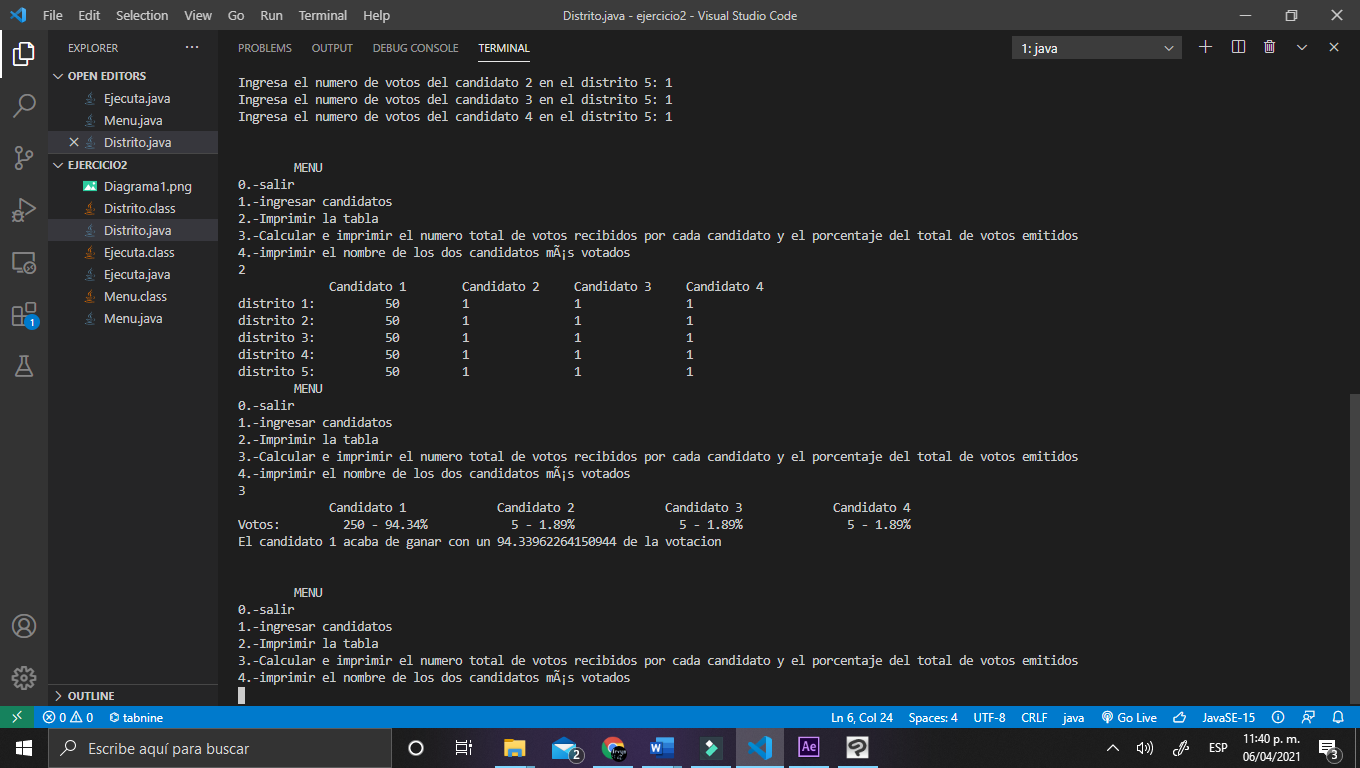
            }

        }

    }

}





Ejercicio 3

