

**Reglas del juego:**

Tal como se le dijo antes,

- Durante este examen usted no puede pedir ABSOLUTAMENTE nada prestado a sus compañeros, ni hablar con ellos.
- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- Puede consultar su cuaderno y su libro; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo.
- No se resolverá ninguna duda durante el examen, así que si algo no le queda claro y usted requiere hacer algún tipo de supuesto, ESCRÍBALO y de una breve explicación de la razón por la cual lo supone.

La forma de calificación de este parcial, para cada una de las situaciones planteadas en cada caso es:

Ítem	Valor
<b>Primer punto:</b>	<b>1.5</b>
Seleccionar la alternativa de salida correcta y justificar las no seleccionadas.	0.5
Identificar errores en el código y corrección de los mismos.	0.5
Identificar entradas correctas.	0.5
<b>Segundo punto:</b>	<b>3.5</b>
Identificar métodos	0.9
Identificar Salidas y Valores de Retorno	0.25
Identificar Entradas y Parámetros	0.25
Descripción del proceso de cada Método	2.1

**PRIMER PUNTO (Vale 1.5 puntos)**

La siguiente matriz de tipo double, almacena el valor en dólares de las ventas de cada uno de los tres productos que vende una empresa en cada una de las cuatro ciudades en las que presencia:

Productos	Ciudades			
	Bogotá	Cali	Medellín	Bucaramanga
Leche en polvo	2000.0	10000.0	7000.0	2000.0
Crema de leche	1000.0	5000.0	900.0	6000.0
Leche condensada	800.0	4000.0	3000.0	700.0

Figura 1 - Matriz con el valor de las ventas de cada producto en cada ciudad

Dado el siguiente código de una clase Ejecutable:

```

package segundo;
import java.util.*;
public class Primero {

    public static void main(String[] args)
    {
        Scanner elQueLee=new Scanner(System.in);
        final int CANT_PRODS=3;
        final int CANT_CIUUD=4;
        int continuar=0;
        double[] [] matriz=ventasProductos(CANT_PRODS,CANT_CIUUD);
        do
        {
            consultar(matriz, CANT_PRODS,CANT_CIUUD);
            System.out.println("1. Para continuar. 0. En caso contrario");
            continuar=elQueLee.nextInt();
        }while(continuar==1);
    }

    public static double[] [] ventasProductos(int cantProductos, int cantCiudades)
    {
        Scanner elQueLee=new Scanner(System.in);
        double[] [] ventasPorCiudad=new double[cantProductos][cantCiudades];
        for(int contColumnas=0;contColumnas<cantCiudades;contColumnas++)
        {
            System.out.println("Para la ciudad No. "+(contColumnas+1));
            for(int contFilas=0;contFilas<cantProductos;contFilas++)
            {
                System.out.println("Para el producto No. "+(contFilas+1));
                ventasPorCiudad[contFilas][contColumnas]=elQueLee.nextDouble();
            }
        }
        return ventasPorCiudad;
    }

    public static void consultar(double ventas[] [], int cantProductos, int cantCiudades)
    {
        Scanner elQueLee = new Scanner(System.in);
        System.out.println("Indique el tipo de consulta a realizar");
        System.out.println("1. Total vendido en una ciudad");
        System.out.println("2. Total de ventas");
        System.out.println("3. Totales de ventas por ciudad");
        int opción;
        do
        {
            opción=elQueLee.nextInt();
            switch(opción)
            {
                case 1: double ventasCiudad=totalEnUnaCiudad(ventas,cantProductos,cantCiudades);
                    System.out.println("Total de ventas en la ciudad $" +ventasCiudad);
                    break;

                case 2: totalVentas(ventas,cantProductos,cantCiudades);
                    break;

                case 3: consultarVentasPorCiudad(ventas,cantProductos,cantCiudades);
                    break;

                default:opción=-1;
                    System.out.println("Opción Incorrecta");
            }
        }while(opción!=-1);
    }

    public static double totalEnUnaCiudad(double ventas[] [], int cantProductos, int
    cantCiudades)
    {
        Scanner elQueLee = new Scanner(System.in);
        System.out.println("Digite el número de la ciudad a consultar");
        int ciudad; boolean error=false;double totalCiudad=0.0;
        do
        {
            ciudad=elQueLee.nextInt();
            if(ciudad<0||ciudad>cantCiudades)
    
```

```

        {
            error=true;
        }
        else
        {
            for(int cont=0;cont<cantProductos;cont++)
            {
                totalCiudad+=ventas[cont][ciudad-1];
            }
        }
    }while(error);
    return totalCiudad;
}

public static void totalVentas(double ventas[][ ], int cantProductos, int
cantCiudades)
{
    int contFilas=0,contColumnas=0;
    for(;contFilas<cantCiudades;contFilas++)
    {
        for(;contColumnas<cantProductos;contColumnas++)
        {
            total+=ventas[contFilas,contColumnas];
        }
    }
    System.out.println("El total de ventas es $" +total);
    return total;
}

public static void consultarVentasPorCiudad(double ventas[][ ], int cantProductos, int
cantCiudades)
{
    for(int contColumnas=0;contColumnas<cantCiudades;contColumnas++)
    {
        System.out.println("Para la ciudad No. " +(contColumnas+1));
        double totalCiudad=0.0;
        for(int contFilas=0;contFilas<cantProductos;contFilas++)
        {
            totalCiudad+=ventas[contFilas][contColumnas];
        }
        System.out.println("Ventas totales: " +totalCiudad);
    }
}
}

```

1. Teniendo en cuenta el método consultarVentasPorCiudad, seleccione cual de las siguientes alternativas representa la salida correcta por consola. Justifique las razones por las cuales descartó las alternativas no seleccionadas.

a.

```

Para la ciudad No. 1
Ventas totales: 21000.0
Para la ciudad No. 2
Ventas totales: 12900.0
Para la ciudad No. 3
Ventas totales: 8500.0
Para la ciudad No. 4
Ventas totales: 8700.0

```

b.

```

Para la ciudad No. 1
Ventas totales: 3800.0
Para la ciudad No. 2
Ventas totales: 19000.0
Para la ciudad No. 3
Ventas totales: 10900.0
Para la ciudad No. 4
Ventas totales: 8700.0

```

c.

```

Para el producto No. 1
Ventas totales: 3800.0
Para el producto No. 2
Ventas totales: 19000.0
Para el producto No. 3
Ventas totales: 10900.0

```

2. Teniendo en cuenta el método `totalVentas`, indique si tiene errores o no. En caso de tenerlos identifique los errores e indique como deben corregirse. Tenga en cuenta que los errores pueden ser de sintaxis, lógicos o de uso inadecuado de las estructuras.
3. Teniendo en cuenta los métodos `consultar` y `totalEnUnaCiudad`, indique cuales serían las entradas correctas para obtener la siguiente salida por consola:

```
Total de ventas en la ciudad $10900.0
```

### SEGUNDO PUNTO (Vale 3.5 puntos)

El casino "DONDE USTED SIEMPRE GANA", ha decidido realizar de manera electrónica un nuevo juego denominado "PICAS Y FIJAS". Este juego consiste en adivinar una cadena de cuatro dígitos la cual es generada aleatoriamente por el sistema.

Al comenzar el juego, el sistema generará de forma aleatoria el número que el jugador debe adivinar, el cual se debe almacenar en la primera fila de una matriz de 4 columnas por 11 filas (Figura 2).

El jugador tendrá como máximo 10 oportunidades para adivinar el número. Cada vez que el jugador ingrese su número, se debe verificar si este es igual al generado por el sistema, si es igual, debe desplegarse el mensaje "Ganaste!!" y el juego se dará por terminado. Si el número no es el correcto, el sistema debe desplegar el mensaje "Inténtalo de nuevo" (sólo hasta la oportunidad 9) y deberá almacenar en la matriz cada uno de los 4 dígitos del número ingresado por el jugador, a partir de la segunda fila.

Ejemplo:

Número generado por el sistema	4	7	8	0
Intento 1	4	6	7	1
Intento 2	6	7	5	2
.	1	2	3	4
.	5	6	7	8
.	9	0	2	1
.	4	5	2	3
.	5	8	6	3
.	3	2	4	5
.	0	1	8	7
Intento 10	7	6	3	2

Figura 2 - Matriz del juego

En caso de que después de usar sus 10 oportunidades, el jugador no haya acertado, el sistema deberá mostrarle en pantalla la estadística de la cantidad de PICAS y FIJAS por cada uno de los números de cuatro digitados ingresados, es decir, la cantidad de fijas y picas en cada uno de los intentos.

**FIJAS:** Cantidad de dígitos del número ingresado por el jugador que se encuentran en la misma posición del número generado por el sistema.

**PICAS:** Cantidad de dígitos del número ingresado por el jugador contenidos dentro del número generado por el sistema y que no son FIJAS.

Por ejemplo:

Si el número generado de forma aleatoria fue el 4780 y en el primer intento el jugador ingresó los siguientes cuatro dígitos quedando almacenados en la fila 1 de la matriz.

4	6	7	1
---	---	---	---

**PICAS: 1** (el número 7 digitado de tercero en el primer intento del usuario se encuentra entre el conjunto de dígitos a adivinar, pero en otra posición. El 4 no se tiene en cuenta porque es una FIJA.

**FIJAS: 1** (El número 4 se encuentra en la posición correcta)

Un empleado muy pijo de la organización ha empezado a examinar este juego y ha realizado una descripción general de los siguientes métodos que le serán de utilidad.

- Método generarMatrizConNúmeroAleatorio: Este método crea una matriz de tipo entero (int) de 11 filas por 4 columnas. De igual forma, se encarga de almacenar en la primera fila los 4 dígitos aleatorios que corresponden al número que se debe adivinar. Este método no recibe ningún parámetro y retorna la matriz de tipo entero con los valores generados en la primera fila (int).
- Método cantidadPicas: Este método se encarga de calcular la cantidad de Picas de un intento del jugador. Recibe como parámetro la matriz de tipo entero (int) que contiene todos los intentos y la posición de tipo entero (int) de la fila que contiene el intento a evaluar. Retorna un número entero (int) con la cantidad de picas encontradas.

Usted debe:

1. Identificar los otros métodos necesarios para completar la funcionalidad del juego.
2. Realizar el análisis (proceso detallado, parámetros, entradas, salidas, retorno) para cada uno de los métodos que identifique.