

## Módulo 6 – Ejercicios resueltos + Java

### Problema 1

Haz una calculadora. Pide al usuario los 2 números con los que quiere operar y, como tercer argumento, la operación a realizar. En función de si nos pasan un +, -, \*, / decidiremos qué hacer. La lógica necesaria para realizar las operaciones debe hacerse en funciones separadas (una función sumar, otra restar... etc), que deben recibir 2 parámetros que serán los números con los que van a operar.

#### Algoritmo calculadora

Var

valor1: ENTERO, valor2: ENTERO, oper: ENTERO

Inicio

Escribe ("Introduce el primer operando:")

Leer(valor1)

Escribe ("Introduce el segundo operando:")

Leer(valor2)

Escribe ("Elija operacion: 1- SUMA \n 2 - RESTA \n 3 - MULTIPLICA \n 4 - DIVISIÓN")

Leer(oper)

Según sea(oper)

+ : Escribe ("Resultado de la suma:" + suma (valor1, valor2))

- : Escribe ("Resultado de la resta:" + resta (valor1, valor2))

\* : Escribe ("Resultado de la multiplicación:" + mult (valor1, valor2))

/ : Escribe ("Resultado de la división:" + div (valor1, valor2))

Fin

Función suma (val1: ENTERO, val2: ENTERO) : ENTERO

return (val1 + val2)

Fin funcion

Función resta (val1: ENTERO, val2: ENTERO) : ENTERO

return (val1 - val2)

Fin funcion

Función mult (val1: ENTERO, val2: ENTERO) : ENTERO

return (val1 \* val2)

Fin funcion

Función div (val1: ENTERO, val2: ENTERO) : ENTERO

return (val1 / val2)

Fin función

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    int valor1, valor2, oper;
    System.out.print("Introduce el primer operando: ");
    valor1 = reader.nextInt();
    System.out.print("Introduce el segundo operando: ");
    valor2 = reader.nextInt();
    System.out.print("Elija operacion: \n1 - SUMA \n2 - RESTA \n3 - MULTIPLICA \n4 - DIVISIÓN\n");
    oper = reader.nextInt();
    switch(oper){
        case 1:
            System.out.println("Resultado de la suma:" + suma(valor1, valor2));
            break;
        case 2:
            System.out.println("Resultado de la resta:" + resta(valor1, valor2));
            break;
        case 3:
            System.out.println("Resultado de la multiplicación:" + mult(valor1, valor2));
            break;
        case 4:
            System.out.println("Resultado de la división:" + div(valor1, valor2));
            break;
        default:
            System.out.println("Operacion Incorrecta");
    }
}

public static int suma (int val1, int val2){
    return (val1 + val2);
}

public static int resta (int val1, int val2){
    return (val1 - val2);
}

public static int mult (int val1, int val2){
    return (val1 * val2);
}

public static int div (int val1, int val2){
    return (val1/val2);
}
```

## Problema 2

Implementar un subprograma (función) llamada potencia y que realice precisamente esta función, a partir de dos parámetros de entrada: base y exponente. La función debe retornar el valor de dicha potencia.

### Algoritmo elevar

Var

base: REAL

exponente: ENTERO

Inicio

Escribir "Introduce la base: "

Leer (base)

Escribir "Introduce el exponente: "

Leer (exponente)

Escribir (base+"^"+exponente+" = "+potencia(base, exponente))

Fin

Function potencia (base: real; exponente: entero): real

Inicio

pot: REAL =1

i: ENTERO = 0

Mientras (i< exponente)

pot=pot \*base

i = i + 1

Fin mientras

Retorno pot

Fin potencia

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    double base;
    int exponente;
    System.out.print("Introduce la base: ");
    base = reader.nextDouble();
    System.out.print("Introduce el exponente: ");
    exponente = reader.nextInt();
    System.out.println(base + "^" + exponente + " = " + potencia(base, exponente));
}

public static double potencia (double base, int exponente){
    double pot=1;
    int i=0;
    while (i<exponente){
        pot = pot *base;
        i = i + 1;
    }
    return pot;
}
```

### Problema 3

Implementar una función que calcule el factorial de un número. El factorial de un número se calcula así:

Factorial (5) =  $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

#### Algoritmo elevar

Var

numero: ENTERO

Inicio

Escribir "Introduce el numero: "

Leer (numero)

Escribir ("El factorial de "+ numero+ " es "+factorial(numero))

Fin

Function factorial (numero: entero): entero

Inicio

fac: REAL =1

i: ENTERO = 1

Mientras (i <= numero)

    fact=fact \* i

    i = i + 1

Fin mientras

Retorno fac

Fin factorial

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    int numero;

    System.out.print("Introduce el numero: ");
    numero=reader.nextInt();
    System.out.println("El factorial de "+ numero+ " es " + factorial(numero));
}

public static int factorial (int numero){

    int fac = 1;
    int i = 1;
    while (i<= numero){
        fac = fac * i;
        i = i + 1;
    }
    return fac;
}
```

#### Problema 4

Implementar un subprograma que realice la serie de Fibonacci, que es:

Fibonacci (1) = 1

Fibonacci (2) = 1

Fibonacci (n) = Fibonacci (n-1) + Fibonacci (n-2)  $n > 2$

Algoritmo serie\_fibonacci

Var

i, n: entero

Inicio

Escribir ("Introduce un número: ")

Leer (n)

Escribir "La serie de fibonacci es: ")

Para i=1 mientras i<= n paso i=i+1

Escribir ("fibonacci (i)+ " ")

Fin para

Fin

Funcion fibonacci (num: entero): entero

Inicio

a1:entero=1, a2:entero=1, i:entero=3, an:entero

Si num=1 or num=2 entonces retorno 1

Mientras (i<=num)

an = a1 + a2

a1 = a2

a2 = an

i = i + 1

Fin Mientras

Retorno an

Fin fibonacci

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    int n, i;
    System.out.print("Introduce un número: ");
    n = reader.nextInt();
    System.out.print("La serie de fibonacci es: ");
    for (i=1; i<=n; i=i+1){
        System.out.print(fibonacci(i)+" ");
    }
}

public static int fibonacci(int num){
    int a1=1, a2=1, i=3, an=0;
    if (num==1 || num==2)
        return 1;
    while (i<=num){
        an = a1 + a2;
        a1 = a2;
        a2 = an;
        i++;
    }
    return an;
}
```

### Problema 5

Implementar un subprograma que realice la serie de Fibonacci, pero ahora de forma recursiva:

Fibonacci (1) = 1

Fibonacci (2) = 1

Fibonacci (n)= Fibonacci (n-1) + Fibonacci (n-2)  $n > 2$

#### Algoritmo serie\_fibonacci

Var

i, n: entero

Inicio

    Escribir ("Introduce un número: ")

    Leer (n)

    Escribir "La serie de fibonacci es: ")

    Para i=1 mientras i<= n paso i=i+1

        Escribir ("fibonacci (i)+ " ")

    Fin para

Fin

Funcion fibonacci (num: entero): entero

Inicio

    Si (num=1) o (num=2) entonces

        retorno 1

    Sino

        retorno (fibonacci (num-1) + fibonacci (num-2))

    Fin si

Fin fibonacci

```
public static void main(String[] args) {
    int i, n;
    Scanner reader = new Scanner (System.in);
    System.out.print("Introduce un número: ");
    n = reader.nextInt();
    System.out.print("La serie de fibonacci es: ");
    for (i=1; i<=n; i=i+1){
        System.out.print(fibonacci(i)+" ");
    }
}

public static int fibonacci (int num){
    if (num==1 || num==2)
        return 1;
    else
        return (fibonacci(num-1) + fibonacci(num-2));
}
```

## Problema 6

Implementar un subprograma que calcule en cuanto se convierte un capital C al final de N años y a un interés I, con la fórmula del interés compuesto.

$$C_F = C_I(1 + r)^n$$

### Algoritmo elevar

Var

capital: REAL, interes: REAL  
anos: ENTERO

Inicio

Escribir ("Introduce el capital: ")  
Leer (capital)  
Escribir ("Introduce el interes: ")  
Leer (interes)  
Escribir ("Introduce los anos: ")  
Leer (anos)  
Escribir ("El capital compuesto será " + calculo (capital, interés, anos))

Fin

Funcion calculo (c: real; i: real; n: entero): real

Inicio

Para var j:entero=0 mientras j<m paso j=j+1  
    c1 = c1 \* (1 + i/100)  
Fin para  
    retorno (c1)

Fin calculo

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    int capital, interes, anos;
    System.out.print("Introduce el capital: ");
    capital=reader.nextInt();
    System.out.print("Introduce el interes: ");
    interes=reader.nextInt();
    System.out.print("Introduce los años: ");
    anos=reader.nextInt();
    System.out.print("El capital compuesto será " + calculo((double)capital, (double)interes, anos));
}

public static double calculo(double c, double i, int m){
    double c1 = c;
    for (int j=0; j<m; j++){
        c1 = c1 * (1 + i/100);
    }
    return (c1);
}
```

### Problema 7

Implementar un subprograma al que pasándole como parámetros 2 valores enteros M y N, me calcule el combinatorio

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

#### Algoritmo combinatorio

Var

m,n: entero

Inicio

Escribir "Introduce el valor de m: "

Leer m

Escribir "Introduce el valor de n (n<m): "

Leer n

Escribir "El combinatorio es " + factorial (m) / (factorial(n)\*factorial(m-n))

Fin

Funcion factorial (num: entero): entero

Inicio

Si num=0 entonces

retorno 1

Sino

retorno (num \* factorial (num-1))

Fin si

Fin factorial

```
public static void main(String[] args) {
    Scanner reader = new Scanner (System.in);
    int m, n;
    System.out.print("Introduce el valor de m: ");
    m = reader.nextInt();
    System.out.print("Introduce el valor de n (n<m): ");
    n = reader.nextInt();
    System.out.print("El combinatorio es " + factorial(m) / (factorial(n) * factorial(m-n)));
}

public static int factorial (int num){
    if (num==0)
        return 1;
    else
        return (num * factorial(num-1));
}
```



### Problema 8

¿Qué se escribe en pantalla tras la siguiente ejecución?

```
Algoritmo EJ
Var
    A,B,C: entero
Inicio
    A = 1
    B = 2
    C = A+3
    A = F1 (A, B-C, C)
    B = C - F2(A, B)
    C = F3 (C)
    Escribir ("A:" + A + " B:" + B + " C:" + C)
Fin

Funcion F1 ( x: entero; y: entero; z: entero):entero
Inicio
    retorno (x + y + z)
Fin F1

Funcion P2 (x: entero; y: entero):entero
Inicio
    retorno (x - y)
Fin F2

Funcion F (x: entero): entero
Inicio
    retorno (x + 3)
Fin F3
```

```
public static void main(String[] args) {
    int A, B, C;
    A = 1;
    B = 2;
    C = A + 3;
    A = F1(A, B-C, C);
    B = C - F2(A, B);
    C = F3(C);
    System.out.println("A:" + A + " B:" + B + " C:" + C);
}

public static int F1 (int x, int y, int z){
    return (x + y + z);
}

public static int F2 (int x, int y){
    return (x-y);
}

public static int F3 (int x){
    return (x+3);
}
```

La solución es A=3; B=3; C=7

### Problema 9

¿Qué escribe este programa?

Algoritmo Principal

Var

Datos [10]: entero

i: entero

inicio

Para i=0 mientras i< 10 paso i=i+1

datos [i] = i+1

Fin Para

Datos[3] = F1 (datos)

Para i=0 mientras i< 10 paso i=i+1

Escribir (datos [i] + " ")

Fin Para

Fin

Funcion P1 (a[10]: entero): entero

Inicio

x: entero = 0

Para i=0 mientras i<10 paso i=i+1

x = x + a[i]

Fin Para

retorno x

Fin P1

```
public static void main(String[] args) {
    int datos [] = new int [10];
    int i;
    for (i=0; i<10;i=i+1){
        datos [i] = (i+1);
    }
    datos[3] = F1(datos);
    for(i=0; i<10; i++)
        System.out.print(datos[i]+" ");
}

public static int F1 (int a[]){
    int x = 0;
    for (int i=0; i<10; i++)
        x = x + a[i];
    return x;
}
```

Solución: 1, 2, 55, 4, 5, 6, 7, 8, 9, 10.

### Problema 10

Dada una matriz A de M\*N elementos, actualizarla tal que la matriz resultante tenga divididos a los elementos de la diagonal principal por la suma de los elementos que no forman parte de ella.

```
Algoritmo div_matriz
Const
    M: entero = 5
    N: entero = 3
Var
    A[M][N]: real
    Suma: real
Inicio
    pedir_datos(a)
    Suma=sumar (a)
    Escribir(a, suma)
Fin

Procedimiento pedir_datos (matriz [M][N]: real)
Var
    i, j: entero
Inicio
    Para i=1 mientras i<= M paso i=i+1
        Para j=1 mientras j<=N paso j=j+1
            Escribir ("Introduce el elemento [" + i + "][" + j + "]: ")
            Leer (a[i][j])
        Fin Para
    Fin Para
Fin pedir_datos

Funcion sumar (matriz[M][N]: real): double
Var
    i, j: entero
    s: real
Inicio
    s = 0
    Para i=1 mientras i<=M paso i=i+1
        Para j=1 mientras j<=N paso j=j+1
            Si i<> j entonces
                s = s + matriz [i][j]
            Fin si
        Fin Para
    Fin Para
    retorno s
Fin sumar
```

Procedimiento escribir (matriz [M][N] : real; s: real)

Var

i, j: entero

Inicio

Para i=1 hasta M paso j=j+1

Para j=1 mientras j<=N paso j=j+1

Si i=j entonces

escribir (a[i][j]/s + " ")

Sino

escribir (a[i][j]+ " ")

Fin si

Fin Para

escribir (Salto de línea)

Fin Para

Fin escribir

```
static int M=3;
static int N=3;
public static void main(String[] args) {
    double A [][] = new double [M][N];
    double suma;
    pedir_datos(A);
    suma = sumar(A);
    escribir(A, suma);
}

static void pedir_datos (double matriz [][]) {
    int i, j;
    Scanner reader = new Scanner(System.in);
    for (i=0; i<M; i++){
        for (j=0; j<N; j++){
            System.out.print("Introduce el elemento [" + i + "][" + j + "]: ");
            matriz[i][j] = reader.nextDouble();
        }
    }
}

static double sumar (double matriz[][]){
    int i,j;
    double s = 0;
    for (i=0; i<M; i++){
        for (j=0; j<N; j++){
            if (i!=j)
                s = s + matriz[i][j];
        }
    }
    return s;
}

static void escribir (double matriz [],[], double s){
    int i, j;
    for (i=0; i<M; i++){
        for (j=0; j<N; j++){
            if (i==j)
                System.out.print(matriz[i][j]/s+" ");
            else
                System.out.print(matriz[i][j]+" ");
        }
        System.out.println(" ");
    }
}
```

### Problema 11

Tengo guardado en una estructura los alumnos de nuestra escuela, sabiendo que hay 3 cursos, M alumnos por curso y N asignaturas por alumno, determinar mediante subprogramas:

1. Cuál es la nota media de un determinado curso.
2. Cuantos aprobados y suspensos hay en una determinada asignatura.
3. Cuál es el alumno de la escuela con mejor nota media.

#### Algoritmo escuela

##### Const

```
    cursos=3
    alumnos=M
    asignaturas=N
```

##### Var

```
    Nota [cursos][alumnos][asignaturas]: real
```

##### Inicio

```
    pedir_datos (nota)
    media_curso (nota)
    ap_susp (nota)
    Media_alum (nota)
```

##### Fin

#### Procedimiento pedir\_datos (n[][][]: entero)

##### Var

```
    i, j, k: entero
```

##### Inicio

```
    Para i=0 mientras i<cursos paso i=i+1
        Para j=0 mientras j<alumnos paso j=j+1
            Para k=0 mientras k<asignaturas paso k=k+1
                Escribir ("Nota del alumno [" + j + "] asignatura [" + k + "] curso[" + i + "]: ")
                Leer (n[i][j][k])
```

```
            Fin Para
```

```
        Fin Para
```

```
    Fin Para
```

##### Fin pedir\_datos

#### Procedimiento ap\_susp (n[][][]:entero)

##### Var

```
    susp, ap, i, j: entero
```

##### Inicio

```
    susp=0
    ap= 0
    Escribir ("¿De qué asignatura quieres ver aprobados y suspensos?")
    Leer (asig)
    Para i=1 mientras i<=cursos paso i=i+1
        Para j=1 mientras j<=alumnos paso j=j+1
            Si n[i][j][asig] >=5 entonces
                ap=ap + 1
            Sino
                susp=susp + 1
```

```

        Fin si
    Fin Para
Fin Para
Fin Para
Escribir ("En la asignatura " + asig + " hay " + ap + " aprobados")
Escribir ("En la asignatura " + asig + " hay " + susp + " suspensos")
Fin ap_susp

Procedimiento media_curso (n[][][]: real)
Var
    j, k, resp: entero
    media, suma: real
Inicio
    Suma = 0.0
    Escribir "¿De qué curso quieres hacer la media?"
    Leer (resp)
    Para j=1 mientras j<= alumnos paso j=j+1
        Para k=1 mientras k<= asignaturas paso j=j+1
            Suma:=suma + n[resp][j][k]
        Fin Para
    Fin Para
    Media = suma /(alumnos*asignatura)
    Escribir ("La nota media del curso " + resp+ " es " + media)
Fin media_curso

Procedimiento media_alum (n[][][]: real)
Var
    i, j, alum, curs: entero
    suma, media, mayor: real
Inicio
    Mayor = 0.0
    Para i=0 mientras i<curso paso i=i+1
        Para j=0 mientras j<alumnos paso j=j+1
            Suma = 0.0
            Para k=1 mientras k<asignaturas paso k=k+1
                suma = suma + n[i][j][k]
            Fin Para
            Media = suma / asignaturas
            Si media > mayor entonces
                mayor = media
                Curs= i
                Alum = j
            Fin si
        Fin Para
    Fin Para
    Escribir ("El alumno con mayor media es el " + alum + " del curso " + curs)
    Escribir ("Su nota media es de " + mayor)
Fin media_alum

```

```
static int cursos = 2;
static int alumnos = 2;
static int asignaturas = 2;

public static void main(String[] args) {
    double nota [][][] = new double [cursos][alumnos][asignaturas];
    pedir_datos(nota);
    media_curso(nota);
    ap_susp(nota);
    media_alum(nota);
}

static void pedir_datos (double nota[][][]) {
    int i,j,k;
    Scanner reader = new Scanner(System.in);
    for (i=0; i<cursos; i++){
        for (j=0; j<alumnos; j++) {
            for (k=0; k<asignaturas;k++){
                System.out.print("Nota del alumno [" + j + "] asignatura [" + k + "] curso [" + i + "]: ");
                nota[i][j][k] = reader.nextDouble();
            }
        }
    }
}

static void ap_susp (double nota [][][] ){
    Scanner reader = new Scanner(System.in);
    int susp, ap, i, j;
    susp =0;
    ap = 0;
    System.out.print("¿De qué asignatura quieres ver aprobados y suspensos? ");
    int asig = reader.nextInt();
    for (i=0; i<cursos; i++){
        for (j=0; j<alumnos; j++) {
            if (nota[i][j][asig]>=5)
                ap = ap + 1;
            else
                susp = susp + 1;
        }
    }
    System.out.println("En la asignatura " + asig + " hay " + ap + " aprobados");
    System.out.println("En la asignatura " + asig + " hay " + susp + " suspensos");
}
```

```
static void media_curso (double nota[][][]){
    Scanner reader = new Scanner(System.in);
    int j, k, resp;
    double media, suma;
    suma = 0.0;
    System.out.print("¿De qué curso quieres hacer la media? ");
    resp=reader.nextInt();
    for (j=0; j<alumnos; j++){
        for (k=0; k<asignaturas; k++){
            suma = suma + nota[resp][j][k];
        }
    }
    media = suma/(alumnos*asignaturas);
    System.out.println("La nota media del curso " + resp + " es " + media);
}
```

```
static void media_alum (double nota[][][]){
    int i, j, k, alum=0, curs=0;
    double suma, media, mayor;
    mayor = 0.0;
    for (i=0; i<cursos; i++){
        for(j=0; j<alumnos; j++){
            suma = 0.0;
            for (k=0; k<asignaturas; k++){
                suma = suma + nota[i][j][k];
            }
            media = suma / asignaturas;
            if (media > mayor){
                mayor = media;
                curs = i;
                alum = j;
            }
        }
    }
    System.out.println("El alumno con mayor media es el " + alum + " del curso " + curs);
    System.out.println("Su nota media es de " + mayor);
}
```



## Problema 12

Multiplicar 2 matrices de dimensiones  $M \times N$  y  $P \times Q$ .

Algoritmo multiplicar\_matrices

Const

M: entero=2

N: entero = 2

P: entero = 2

Q: entero=2

Var

mat1[M][N]: entero

mat2[P][Q]: entero

res[M][Q]: entero

Inicio

pedir\_datos (mat1, mat2)

multiplicar (mat1, mat2, res)

visualizar (res)

Fin

Procedimiento pedir\_datos (m1[][: entero; m2[][: entero)

Var

i, j: entero

Inicio

Para i=0 mientras i<M paso i=i+1

Para j=0 mientras j< N paso j=j+1

Escribir ("Introduce el elemento [" + i + "]" + j + "] de la matriz 1:")

Leer(m1[i][j])

Fin Para

Fin Para

Desde i=1 hasta P

Para j=1 mientras j<Q paso j=j+1

Escribir ("Introduce el elemento [" + i + "]" + j + "] de la matriz 2: ")

Leer (m2[i][j])

Fin Para

Fin Para

Fin pedir\_datos

Procedimiento multiplicar (m1[][: entero; m2[][: entero; resul[][:entero)

Var

Fila, i, j: entero

Inicio

Para fila=0 mientras fila<N paso fila =fila+1

Para i=0 mientras i<P paso i=i+1

resul[fila][i] = 0

Para j=1 mientras j<Q paso j=j+1

resul[fila,i] = resul[fila][i] + m1[fila][j] \* m2[j][i]

Fin Para

```

        Fin Para
    Fin Para
Fin multiplicar

Procedimiento visualizar (resul [][]: entero)
Var
    i, j: entero
Inicio
    Para i=1 mientras i<M paso i=i+1
        Para j=1 mientras j<Q paso j=j+1
            Escribir (resul[i][j]+"" )
        Fin Para
    Fin Para
Fin visualizar

```

```

static int M = 2;
static int N = 2;
static int P = 2;
static int Q = 2;

public static void main(String[] args) {
    int mat1 [][] = new int [M][N];
    int mat2 [][] = new int [P][Q];
    int res  [][] = new int [M][Q];
    pedir_datos(mat1, mat2);
    multiplicar(mat1, mat2, res);
    visualizar(res);
}

static void pedir_datos (int m1[][], int m2[][]){
    Scanner reader = new Scanner(System.in);
    int i, j;
    for (i=0; i<M; i++){
        for (j=0; j<N; j++){
            System.out.print("Introduce el elemento [" + i + "][" + j + "] de la matriz 1: ");
            m1[i][j] = reader.nextInt();
        }
    }
    for (i=0; i<P; i++){
        for (j=0; j<Q; j++){
            System.out.print("Introduce el elemento [" + i + "][" + j + "] de la matriz 2: ");
            m2[i][j] = reader.nextInt();
        }
    }
}

static void multiplicar (int m1[][], int m2[][], int res[][]){
    int fila, i, j;
    for (fila=0; fila<N; fila++){
        for (i=0; i<P; i++){
            res[fila][i] = 0;
            for (j=0; j<Q; j++){
                res[fila][i] = res[fila][i] + m1[fila][j] * m2[j][i];
            }
        }
    }
}

```

### Problema 13

Una empresa consta de 5 departamentos con 20 empleados cada departamento, si tengo todas las ventas en una estructura, determinar:

- Ventas de un determinado departamento en un determinado mes.
- Ventas de un determinado empleado en un determinado departamento.
- Cuál es el departamento con más ventas.

Algoritmo empresa

Const

D=5

E=20

M=12

Var

ventas [D][E][M]: real;

Inicio

Pedir\_datos (ventas)

Dep\_mes (ventas)

Dep\_empleado (ventas)

Mejor\_dep (ventas)

Fin

Procedimiento pedir\_datos (a[][][]: real)

Var

i, j, k: entero

Inicio

Para i=0 mientras i<D paso i=i+1

Para j=0 mientras j<E paso j=j+1

Para k=0 mientras k<M paso k=k+1

Escribir ("Ventas del departamento " + i + " empleado " + j + " mes " + k)

Leer (a[i][j][k])

Fin Para

Fin Para

Fin Para

Fin pedir\_datos

Procedimiento dep\_mes (a[][][]:real)

Var

dep, j, mes: entero

v: real

Inicio

v = 0.0

Escribir ("¿De que departamento quieres ver las ventas (0-“+D-1+”)?”)

Leer (dep)

Escribir ("¿Que mes (0-“+M-1+”)?”)

Leer (mes)

Para j=0 mientras j<E paso j=j+1

v = v + a[dep][j][mes]

Fin Para

Escribir ("Las ventas del departamento "+dep+" en el mes "+mes+" son "+v)

Fin dep\_mes

Procedimiento dep\_empleado (a[][][]: real)

Var

dep, empleado, k: entero

v: real

Inicio

v = 0.0

Escribir ("¿Que empleado quieres visualizar (0-“+E-1+“)?")

Leer empleado

Escribir ("¿De que departamento (0-“+ D-1 + “)? ")

Leer dep

Para k=0 mientras k<M paso k=k+1

v = v + a[dep][empleado][k]

Fin Para

Escribir ("Las ventas del empleado " + empleado + " del departamento "+dep+ " son "+v)

Fin dep\_empleado

Procedimiento mejor\_dep (a[][][]: real)

Var

i, j, k, dep: entero

mayor, v: real

Inicio

mayor = 0.0

Para i=0 mientras i<D paso i=i+1

V = 0

Para j=0 mientras j<E paso j=j+1

Para k=0 mientras k<M paso k=k+1

v = v + a[i][j][k]

Fin Para

Fin Para

Si v > mayor entonces

mayor = v

Dep = i

Fin si

Fin Para

Escribir ("El mejor departamento es el "+dep+ " con "+mayor)

Fin mejor\_dep

```
static int D = 2;
static int E = 2;
static int M = 2;

public static void main(String[] args) {
    double ventas [][][] = new double [D][E][M];
    pedir_datos(ventas);
    dep_mes(ventas);
    dep_empleado(ventas);
    mejor_dep(ventas);
}

static void pedir_datos (double a[][][]){
    int i, j, k;
    Scanner reader = new Scanner(System.in);
    for (i=0; i<D; i++) {
        for (j=0; j<E; j++){
            for (k=0; k<M; k++){
                System.out.print("Ventas del departamento " + i + " empleado " + j + " mes " + k + ": ");
                a[i][j][k] = reader.nextDouble();
            }
        }
    }
}
```

```
static void dep_mes (double a[][][]){
    Scanner reader = new Scanner (System.in);
    int dep, j, mes;
    double v = 0.0;
    System.out.print("¿De que departamento quieres ver las ventas (0-"+(D-1)+")? ");
    dep = reader.nextInt();
    System.out.print("¿De que mes (0-"+(M-1)+")? ");
    mes = reader.nextInt();
    for (j=1; j<E; j++)
        v = v + a[dep][j][mes];
    System.out.println("Las ventas del departamento " + dep + " en el mes " + mes + " son " + v);
}

static void dep_empleado (double a[][][]){
    int dep, empleado, k;
    Scanner reader = new Scanner(System.in);
    double v = 0.0;
    System.out.print("¿Que empleado quieres ver las ventas (0-"+(E-1)+")? ");
    empleado=reader.nextInt();
    System.out.print("¿De que departamento (0-"+(D-1)+")? ");
    dep = reader.nextInt();
    for (k=0; k<M; k++)
        v = v + a[dep][empleado][k];
    System.out.println("Las ventas del empleado " + empleado + " del departamento "+dep+" son "+v);
}
```

```
static void mejor_dep (double a[][][]){
    int i, j, k, dep=0;
    double mayor, v;
    mayor = 0.0;
    for (i=0; i<D; i++){
        v = 0;
        for (j=0; j<E; j++){
            for (k=0; k<M; k++){
                v = v + a[i][j][k];
            }
        }
        if (v > mayor){
            mayor = v;
            dep = i;
        }
    }
    System.out.println("El mejor departamento es el " + dep + " con " + mayor);
}
```

## Problema 14

Vamos a hacer el juego de "hundir la flota". Crea un algoritmo que se invente una cuadrícula de 10 x 10, y dentro añada 6 barcos de 1 casilla. Posteriormente, el programa deberá pedirnos por teclado la casilla que queremos atacar, y decirnos si hemos acertado o fallado. Cuando hundamos todos los barcos, deberá avisar de que se terminó la partida, y acabar el programa. La comprobación de si hemos acertado o fallado en el ataque, deberá hacerse con una función. Por supuesto, cuando hundimos un barco debe borrarse del mapa, y la próxima vez que se ataque esa posición el resultado debe ser un fallo.

### Algoritmo empresa

Var

FLOTA [10][10]: entero

Inicio

var fila, columna, cont\_aciertos: ENTERO

var barco: BOOL

crea\_partida()

Escribe("Empieza a jugar...")

Mientras (cont\_aciertos < 6)

    Escribe("Introduce Fila (1-10): ")

    Leer (fila)

    Escribe("Introduce Columna (1-10): ")

    Leer (columna)

    barco = disparo (fila-1, columna-1)

    SI (barco == TRUE) ENTONCES

        Escribe ("TOCADO!!!")

        cont\_aciertos = cont\_aciertos + 1

    SINO

        Escribe("AGUA")

Fin mientras

Escribe ("FIN Partida")

Fin

FUNCION disparo (X:ENTERO,Y: ENTERO): BOOL

    BOOL: tocado

    SI (FLOTA [X][Y]) == 1) ENTONCES

        tocado = true

        FLOTA [X][Y] = 0

    SINO

        tocado = FALSE

    FIN SI

    return tocado;

FIN disparo

Procedimiento crea\_partida ()

    num\_barcos, x, y:entero

    num\_barcos = 1

    MIENTRAS (num\_barcos <= 6)

        Escribe ("Introduce fila barco numero " + num\_barcos)

        Leer(x)

        Escribe ("Introduce columna barco numero " + num\_barcos)

```

Leer(y)
Si FLOTA [x-1][y-1] = 1 entonces
    Escribe("Coordenadas barco repetidas")
SINO
    FLOTA [x-1][y-1] = 1
    num_barcos = num_barcos + 1
Fin Si
FIN MIENTRAS

```

FIN crea\_partida

```

static int FLOTA[][] = new int [10][10];
public static void main(String[] args) {
    Scanner reader = new Scanner(System.in);
    int fila, columna, cont_aciertos=0;
    boolean barco = false;
    crea_partida();
    System.out.println("Empieza a jugar...");
    while (cont_aciertos< 6){
        System.out.print("Introduce Fila (1-10): ");
        fila = reader.nextInt();
        System.out.print("Introduce Columna (1-10): ");
        columna = reader.nextInt();
        barco = disparo(fila-1, columna-1);
        if (barco == true){
            System.out.println("TOCADO!!!");
            cont_aciertos = cont_aciertos + 1;
        }
        else
            System.out.println("AGUA");
    }
    System.out.println("FIN Partida");
}

```

```

public static boolean disparo (int X, int Y){
    boolean tocado;
    if (FLOTA[X][Y]==1){
        tocado = true;
        FLOTA[X][Y] = 0;
    }
    else tocado = false;
    return tocado;
}

public static void crea_partida(){
    int num_barcos, x , y;
    Scanner reader = new Scanner(System.in);
    num_barcos = 1;
    System.out.println("Iniciando juego...");
    while (num_barcos<= 6) {
        System.out.print("Introduce fila barco numero " + num_barcos+" (1-10): ");
        x = reader.nextInt();
        System.out.print("Introduce columna barco numero " + num_barcos+" (1-10): ");
        y = reader.nextInt();
        if (FLOTA[x-1][y-1] == 1){
            System.out.println("Coordenadas barco repetidas");
        }
        else {
            FLOTA[x-1][y-1] = 1;
            num_barcos = num_barcos + 1;
        }
    }
}

```