



C. F. G. S. Desarrollo de Aplicaciones Web - Distancia -

Examen de Programación de la 1ª Evaluación. 10-12-2019

“First, solve the problem. Then, write the code.” - John Johnson –

- 1) Hacer un programa que pida por teclado un número impar y mayor o igual que 3, obligando al usuario a que el valor introducido cumpla estas condiciones, visualizando un mensaje de error si el número no es correcto. Dibujar a continuación un diamante de asteriscos con un número de filas igual al valor introducido. Entre cada asterisco hay un espacio en blanco.

Ejemplos de ejecución del programa:

a)

Introduce un número impar y mayor o igual que 3: 7

```
      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

b)

Introduce un número impar y mayor o igual que 3: 8

Número incorrecto, introduce otro valor: 9

```
      *
    * * *
  * * * * *
* * * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```



Solución 1:

```
import java.util.Scanner;

public class Ejercicio1 {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Introduzca un número impar y mayor o igual que 3: ");
        int num;
        int huecos;

        num = teclado.nextInt();
        while (num % 2 == 0 || num < 3) {
            System.out.print("Número incorrecto, introduce otro valor: ");
            num = teclado.nextInt();
        }

        huecos = (num - 1) / 2;
        System.out.println(huecos);
        for (int i = 1; i <= num / 2 + 1; i++) {
            for (int j = 0; j < huecos; j++) {
                System.out.print(" ");
            }
            huecos--;
            for (int h = 1; h <= i; h++) {
                System.out.print("* ");
            }
            for (int h = 1; h < i; h++) {
                System.out.print("* ");
            }
            System.out.println();
        }
        huecos = 1;
        for (int i = num / 2; i >= 1; i--) {
            for (int j = 0; j < huecos; j++) {
                System.out.print(" ");
            }
            huecos++;
            for (int h = 1; h <= i; h++) {
                System.out.print("* ");
            }
            for (int h = 1; h < i; h++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Solución 2:



```
import java.util.Scanner;

public class Ejercicio_1 {

    public static void main(String[] arg) {
        int numero, i, k;
        double j, n;
        String salida;
        boolean cont = false;
        Scanner leo = new Scanner(System.in);
        System.out.print("Introduce un número impar y mayor o igual que 3: ");
        do { //Selección
            numero = leo.nextInt();
            if (numero < 3) {
                System.out.print("Número incorrecto, introduce otro valor: ");
            } else {
                if (numero % 2 == 0) {
                    System.out.print("Número incorrecto, introduce otro valor: ");
                } else {
                    cont = true;
                }
            }
        } while (cont == false);
        // Triangulo Superior
        for (i = 0; i < (numero / 2); i++) {
            salida = "";
            for (j = (numero / 2) - i; j > 0; j--) {
                salida += " ";
            }
            for (k = 0; k < (2 * i + 1); k++) {
                salida += " *";
            }
            System.out.println(salida);
        }
        //Centro
        salida = "";
        for (i = 0; i < numero; i++) {
            salida += " *";
        }
        System.out.println(salida);
        //Triangulo Inferior
        for (n = (numero / 2); n > 0; n--) {
            salida = "";
            for (j = (numero / 2) - n + 1; j > 0; j--) {
                salida += " ";
            }
            for (k = 0; k < (2 * n - 1); k++) {
                salida += " *";
            }
            System.out.println(salida);
        }
    }
}
```



```
}  
}
```

Solución 3:

```
import java.util.Scanner;  
  
public class Ejercicio1 {  
  
    public static void main(String[] args) {  
  
        int x;  
        Scanner leo = new Scanner(System.in);  
        System.out.print("Introduce un número impar mayor o igual que 3: ");  
        x = leo.nextInt();  
  
        while (x < 3 || x % 2 == 0) {  
            System.out.print("El número introducido no es correcto. \nIntroduce un número impar mayor o igual  
que 3: ");  
            x = leo.nextInt();  
        }  
  
        for (int i = 0; i < x; i++) {  
  
            for (int j = 1; j <= Math.abs((x / 2) - i); j++) {  
                System.out.print(" ");  
            }  
  
            for (int j = 0; j < (x - 2 * Math.abs((x / 2) - i)); j++) {  
                System.out.print(" *");  
            }  
            System.out.println("");  
        }  
    }  
}
```

- 2) Hacer un programa que pida por teclado un número entero, y que calcule y escriba la suma de los dígitos del número que sean pares. El programa estará ejecutándose hasta que se introduzca un valor negativo. Es obligatorio utilizar la sentencia for para implementar los bucles.

Ejemplo de ejecución del programa:

Introduce un número entero, teclea un valor negativo para finalizar: 411728
Suma de los dígitos pares: 4 + 2 + 8 = 14

Introduce un número entero, teclea un valor negativo para finalizar: 6
Suma de los dígitos pares: 6 = 6

Introduce un número entero, teclea un valor negativo para finalizar: 612
Suma de los dígitos pares: 6 + 2 = 8

Introduce un número entero, teclea un valor negativo para finalizar: -5



Solución 1:

```
import java.util.Scanner;

public class Ejercicio2 {

    public static void main(String[] args) {
        String stNumero;
        String stDigito;
        int suma;
        int numero;
        int digito;
        boolean flagPrimera;
        Scanner reader = new Scanner(System.in);
        for (;;) {
            System.out.print("Introduce un número entero, teclea un valor negativo para finalizar:");
            numero = reader.nextInt();
            stNumero = Integer.toString(numero);
            if (numero < 0) {
                break;
            }
            flagPrimera = true;
            suma = 0;
            for (int i = 0; i <= stNumero.length() - 1; i++) {
                stDigito = String.valueOf(stNumero.charAt(i));
                digito = Integer.parseInt(stDigito);
                if (digito % 2 == 0 && digito != 0) {
                    if (flagPrimera == false) {
                        System.out.print(" " + digito);
                    } else {
                        System.out.print("Suma de los dígitos pares: " + digito);
                    }
                }
                flagPrimera = false;
                suma = suma + digito;
            }
            System.out.println(" = " + suma);
        }
    }
}
```

Solución 2:

```
import java.util.Scanner;

public class Ejercicio2 {

    public static void main(String[] args) {

        int x;
        Scanner leo = new Scanner(System.in);
        System.out.print("Introduce un número entero, teclea un valor negativo para finalizar: ");
```



```
x = leo.nextInt();
int aux;
int suma = 0;
String textoSuma = "Suma de los dígitos pares: ";

for (; x > 0;) {
    String y = String.valueOf(x);
    for (int i = 0; i < y.length(); i++) {
        aux = Integer.parseInt(y.substring(i, i + 1));
        if (aux % 2 == 0) {
            suma += aux;
            textoSuma += aux + " + ";
        }
    }
}

textoSuma = textoSuma.substring(0, textoSuma.length() - 2) + "= ";

System.out.println(textoSuma + suma);
suma = 0;
textoSuma = "Suma de los dígitos pares: ";
System.out.println("");
System.out.print("Introduce un número entero, teclea un valor negativo para finalizar: ");
x = leo.nextInt();
}
}
}
```

Solución 3:

```
import java.util.Scanner;

public class Ejercicio_2 {

    public static void main(String[] arg) {
        int numero, i, k, suma, count;
        String numeral, salida;
        Scanner leo = new Scanner(System.in);

        for (numero = 0; numero >= 0;) {
            System.out.print("Introduce un número entero, teclea un valor negativo para finalizar: ");
            suma = 0;
            count = 0;
            salida = "";
            numero = leo.nextInt();
            if (numero > 0) {
                numeral = String.valueOf(numero);
                for (i = 0; i < numeral.length(); i++) {
                    k = Integer.parseInt(numeral.substring(i, i + 1));
                    //System.out.println(numeral.substring(i,i+1));
                    if (k % 2 == 0) {
                        if (count == 0) {
                            salida += k + " ";
                        }
                    }
                }
            }
        }
    }
}
```

```
        count++;  
    } else {  
        salida += " " + k + " ";  
        count++;  
    }  
    suma += k;  
}  
}  
System.out.println("Suma de los dígitos pares: " + salida + "= " + suma);  
}  
}  
}
```

- 3) Para implementar el juego de Mario Bros, es necesario contar cuántas veces salta hacia arriba, y cuántas hacia abajo, teniendo en cuenta que para hacer los cálculos Mario se encuentra situado encima del primer muro. Hacer un programa que pida por teclado cuántos muros tiene que saltar Mario. A continuación, tiene que pedir las alturas de cada muro y escribir el número de saltos que hizo hacia arriba y hacia abajo.



Ejemplos de ejecución del programa:

a)

```
Introduce cuantos muros tiene que saltar Mario: 8  
Introduce la altura del muro 1: 1  
Introduce la altura del muro 2: 4  
Introduce la altura del muro 3: 2  
Introduce la altura del muro 4: 2  
Introduce la altura del muro 5: 3  
Introduce la altura del muro 6: 5  
Introduce la altura del muro 7: 3  
Introduce la altura del muro 8: 4
```

```
Salto hacia arriba: 4  
Salto hacia abajo: 2
```

b)

```
Introduce cuantos muros tiene que saltar Mario: 2  
Introduce la altura del muro 1: 9  
Introduce la altura del muro 2: 9
```

```
Salto hacia arriba: 0  
Salto hacia abajo: 0
```

c)



```
Introduce cuantos muros tiene que saltar Mario: 5
Introduce la altura del muro 1: 1
Introduce la altura del muro 2: 2
Introduce la altura del muro 3: 3
Introduce la altura del muro 4: 4
Introduce la altura del muro 5: 5
```

```
Salto hacia arriba: 4
```

```
Salto hacia abajo: 0
```

Solución 1:

```
import java.util.Scanner;

public class Ejercicio3 {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int numMuros;
        int saltosArriba = 0;
        int saltosAbajo = 0;
        int muroActual;
        int muroAnterior = 0;
        System.out.print("Introduce cuantos muros tiene que saltar Mario: ");

        numMuros = teclado.nextInt();
        for (int i = 1; i <= numMuros; i++) {
            System.out.print("Introduzca la altura del muro " + i + " :");
            muroActual = teclado.nextInt();
            if (muroAnterior > 0) {
                if (muroActual > muroAnterior) {
                    saltosArriba++;
                }
                if (muroActual < muroAnterior) {
                    saltosAbajo++;
                }
            }
            muroAnterior = muroActual;
        }

        System.out.println("Saltos hacia arriba : " + saltosArriba);
        System.out.println("Saltos hacia abajo : " + saltosAbajo);
    }
}
```

Solución 2:

```
import java.util.Scanner;

public class Ej3 {

    public static void main(String[] args) {
```




```
int nMuros;
int alturaInicial;
int alturaFinal;
int nSaltosArriba;
int nSaltosAbajo;
Scanner input = new Scanner(System.in);

//Supongo el numero de saltos mayor o igual a 1
System.out.println("Introduce el numero de muros que debe saltar Mario");
nMuros = input.nextInt();

//la casilla inicial hay que leerla siempre
System.out.println("Introduce la altura del muro 1:");
alturaInicial = input.nextInt();

nSaltosArriba = 0;
nSaltosAbajo = 0;
for (int i = 2; i <= nMuros; i++) {
    System.out.println("Introduce la altura del muro " + i + ":");
    alturaFinal = input.nextInt();

    if (alturaFinal > alturaInicial) {
        nSaltosArriba++;
    } else if (alturaFinal < alturaInicial) {
        nSaltosAbajo++;
    }

    alturaInicial = alturaFinal;
}

System.out.println("Saltos hacia arriba: " + nSaltosArriba);
System.out.println("Saltos hacia abajo: " + nSaltosAbajo);
}
```

- 4) Las máquinas son incansables. Pensemos en un ascensor, por ejemplo. Empieza el día en el bajo; el del primero le llama y le hace subir al quinto. Justo después le llaman en el segundo y le toca bajar a la planta baja. Luego el del tercero quiere ir a ver al del noveno... Al final del día ha hecho un largo recorrido de arriba a abajo, y de abajo a arriba. ¿Cuál es la longitud de ese recorrido?

El programa tiene que preguntar por el piso inicial en el que empieza el ascensor (el 0 marca la planta baja; no hay plantas por debajo de ella). A continuación, nos pedirá varias parejas de números, representando cada una de ellas el uso del ascensor por parte de un usuario, con el piso desde el que llama al ascensor y con el piso de destino. Para indicar que ese día no hay más movimientos, se introducirá el valor -1 como piso desde el que se llama al ascensor.

Ejemplos de ejecución del programa:

```
a)
Introduce el piso inicial del ascensor: 0
Introduce el piso desde el que se llama: 1
```



```
Introduce el piso de destino: 5
Introduce el piso desde el que se llama: 2
Introduce el piso de destino: 0
Introduce el piso desde el que se llama: 3
Introduce el piso de destino: 9
Introduce el piso desde el que se llama: -1
```

Pisos recorridos por el ascensor a lo largo del día: 19

b)

```
Introduce el piso inicial del ascensor: 5
Introduce el piso desde el que se llama: 5
Introduce el piso de destino: 4
Introduce el piso desde el que se llama: -1
```

Pisos recorridos por el ascensor a lo largo del día: 1

c)

```
Introduce el piso inicial del ascensor: 1
Introduce el piso desde el que se llama: 2
Introduce el piso de destino: 3
Introduce el piso desde el que se llama: 4
Introduce el piso de destino: 5
Introduce el piso desde el que se llama: -1
```

Pisos recorridos por el ascensor a lo largo del día: 4

Solución 1:

```
import java.util.*;

public class Ejercicio4 {

    public static void main(String[] args) {

        int llama = 0;
        int destino = 0;
        int actual = 0;
        int pisos = 0;
        int recorrido = 0;
        Scanner entrada = new Scanner(System.in);
        System.out.print("Introduce el piso inicial del ascensor: ");
        actual = entrada.nextInt();

        while (llama != -1) {
            System.out.print("Introduce el piso desde el que se llama: ");
            llama = entrada.nextInt();

            if (llama != -1) {

                if (actual < llama) {
                    pisos = llama - actual;
                } else if (actual > llama) {
                    pisos = actual - llama;
                }
            }
        }
    }
}
```



```
}  
recorrido = recorrido + pisos;  
actual = llama;  
  
System.out.print("Introduce el piso de destino: ");  
destino = entrada.nextInt();  
if (actual < destino) {  
    pisos = destino - actual;  
} else if (actual > destino) {  
    pisos = actual - destino;  
}  
recorrido = recorrido + pisos;  
actual = destino;  
}  
}  
System.out.println("Pisos recorridos por el ascensor a lo largo del día: " + recorrido);  
}  
}
```

Solución 2:

```
import java.util.Scanner;  
  
public class Ej4 {  
  
    public static void main(String[] args) {  
        int pisoInicial;  
        int pisoOrigen;  
        int pisoDestino;  
        int movimientos;  
        Scanner input = new Scanner(System.in);  
  
        while (true) {  
            System.out.println("Introduce el piso inicial del ascensor: ");  
            pisoInicial = input.nextInt();  
            if (pisoInicial < 0) {  
                System.out.println("Piso no valido, debe ser mayor o igual a 0");  
            } else {  
                break;  
            }  
        }  
  
        movimientos = 0;  
  
        while (true) {  
            System.out.println("Introduce el piso de origen: ");  
            pisoOrigen = input.nextInt();  
  
            if (pisoOrigen < 0) {  
                break;  
            }  
        }  
    }  
}
```



```
        System.out.println("Introduce el piso de destino: ");
        pisoDestino = input.nextInt();

        movimientos += Math.abs(pisoOrigen - pisoInicial) + Math.abs(pisoDestino - pisoOrigen);
        pisoInicial = pisoDestino;
    }

    System.out.println("Pisos recorridos a lo largo del día: " + movimientos);
}
}
```

Solución 3:

```
import java.util.Scanner;

public class Ejercicio4 {

    public static void main(String[] args) {

        int pisoInicial;
        int pisoLlamada;
        int pisoDestino;
        int suma = 0;

        Scanner leo = new Scanner(System.in);
        System.out.print("Introduce el piso inicial del ascensor: ");
        pisoInicial = leo.nextInt();

        do {
            System.out.print("Introduce el piso desde el que se llama: ");
            pisoLlamada = leo.nextInt();
            if (pisoLlamada < 0) {
                break;
            }
            System.out.print("Introduce el piso de destino: ");
            pisoDestino = leo.nextInt();
            suma += Math.abs(pisoLlamada - pisoInicial) + Math.abs(pisoDestino - pisoLlamada);
            pisoInicial = pisoDestino;
        } while (true);

        System.out.println("Pisos recorridos por el ascensor a lo largo del día: " + suma);
    }
}
```

Puntuación de cada ejercicio:

Ejercicio 1:	2.5 puntos
Ejercicio 2:	2.5 puntos
Ejercicio 3:	2.5 puntos
Ejercicio 4:	2.5 puntos

Criterios de corrección y cómo entregar el examen:



Aunque un programa no funcione, entregarlo igual, ya que puede ocurrir que esté bien planteado y falle por un pequeño detalle. De la misma manera, si se pide hacer un programa con una determinada sentencia para implementar un bucle y no os sale, hacerlo de otra manera. Evidentemente no puntuará lo mismo, pero al menos podéis completar el ejercicio.

Hay que subir la solución del examen como tarea, en formato de procesador de texto en un único documento. El documento tiene que incluir el código fuente completo de cada uno de los ejercicios. Nombrar al documento con vuestro nombre y apellidos.



AYUDA PARA EL EXAMEN

A la hora de programar, netbeans nos ayuda a escribir código de manera más rápida mediante decenas de atajos (shortcuts). NetBeans incluye un archivo pdf con la lista completa de dichos atajos. Esta lista se encuentra dentro del directorio de instalación de NetBeans en una carpeta llamada "nb". Allí encontraremos 2 pdfs:

shortcuts.pdf

shortcuts_mac.pdf

El primero contiene los atajos para PC (Windows/Linux), el segundo para los usuarios de Mac.

Acordaros dónde está el fichero shortcuts.pdf para poder consultarlo el día del examen. Os recomiendo que os familiaricéis con el uso de estos atajos. De todas maneras, os pongo a continuación algunos de los atajos más útiles:

1. `sout + Tab`, genera la línea: `System.out.println("");`
2. `psvm` genera un método main

```
public static void main(String[] args) {  
    }  
}
```

3. `sw + Tab`, genera una sentencia switch

```
switch (var) {  
    case val:  
  
        break;  
    default:  
        throw new AssertionError();  
}
```

4. `for`, `while`, `if + ctrl+spacio` dan sugerencias para construir cualquiera de estas estructuras de control

Tampoco es necesario saberse de memoria todos los métodos de cada una de las clases que tiene el lenguaje. Para eso podemos consultar la API del lenguaje y los IDEs nos ayudan. Si no recordamos como se calcula el valor absoluto de un valor, simplemente escribiendo `Math`. (clase utilizada en los apuntes), nos aparecen todos los métodos de la clase, con lo que podemos localizar el que nos interesa utilizar. De la misma manera, si después de una variable de tipo `String` pongo el punto también me aparecen todos los métodos (extraer un carácter, convertir a mayúsculas.....).



Math.

	E	double
	PI	double
	IEEEremainder(double d, double d1)	double
	abs(double d)	double
	abs(float f)	float
	abs(int i)	int
	abs(long l)	long
	acos(double d)	double
	asin(double d)	double
	atan(double d)	double
	atan2(double d, double d1)	double
	cbrt(double d)	double
	ceil(double d)	double
	cos(double d)	double

String nombre;

nombre.

	charAt(int i)	char
	codePointAt(int i)	int
	codePointBefore(int i)	int
	codePointCount(int i, int i1)	int
	compareTo(String string)	int
	compareToIgnoreCase(String string)	int
	concat(String string)	String
	contains(CharSequence cs)	boolean
	contentEquals(CharSequence cs)	boolean

Plantilla de un programa que lee un valor por teclado.

```
import java.util.Scanner;
```

```
public class Plantilla {
```

```
    public static void main(String[] args) {
```

```
        Scanner teclado = new Scanner(System.in);
```

```
        int numero;
```

```
        System.out.println("Introduce un primer número:");
```

```
        numero = teclado.nextInt();
```

```
    }
```

```
}
```