

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

15. ¿Qué hay incorrecto en el siguiente fragmento de código?

```
if (max = MAXIMO_ENCONTRADO) {  
    System.out.println (max+ " es el maximo encontrado");  
}
```

- a) Sobran las llaves del if porque sólo contiene una sentencia
- b) El nombre de MAXIMO\_ENCONTRADO es incorrecto, debe escribirse con minúsculas
- c) Despues de la } debe ir un ;
- d) En la condición del if debe usarse el operador de comparación y no el de asignación

Solución: *d*

#### 3.1.2. Bucles While

16. ¿Cuál es la salida del siguiente fragmento de código?

```
index=1;  
while (index<10) {  
    System.out.println (index);  
    index ++;  
}
```

Solución:

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

17. ¿Cuánto vale *count* al acabar el bucle while?

```
int count=50;
while (count >= 0) {
    System.out.println (count);
    count=count+1;
}
```

- a) El fragmento de código no es válido porque el bucle while es infinito
- b) 50
- c) El fragmento de código no es válido porque no se usa el operador incremento para count
- d) 51

Solución: *a*

18. ¿Qué hay incorrecto en el siguiente fragmento de código? Indique tres formas de solucionar el problema.

```
count=50;
while (count >= 0) {
    System.out.println (count);
    count=count+1;
}
```

Solución:

Es un bucle infinito y el count no se escribiría bien. Si simplemente queremos realizar el bucle 51 veces (50+0) se podría solucionar así:

a)

```
count=50;
while (count != -1) {
    System.out.println (count);
    count=count-1;
}
```

b)

```
count=50;
while (count >= 0) {
    System.out.println (count);
    count=count-1;
}
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

c)

```
count=0;
while (count <= 50) {
    System.out.println (count);
    count=count+1;
}
```

19. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args){
        int c = 1;
        final int N=5;
        while (c <= N) {
            System.out.print(c);
            c--;
        }
    }
}
```

- a) Error: N no puede llevar el modificador final
- b) 12345
- c) 54321
- d) Error: Bucle infinito

Solución: *d*

20. ¿Cuál es la salida del siguiente fragmento de código?

```
double x,y;
x=13.0;
y=7.0;
int v1 =(int) ((x/4)+1);
int v2 = (int) (y/4);
while (v1 > 0) {
    v1 = v1-1;
    v2 = v2 * v1;
    System.out.print(v2 + " ");
}
```

Solución: 3 6 6 0

21. Seleccionar la opción correcta cuando se ejecute el siguiente programa:

```
import java.io.*;
class Ejercicio{
    public static void main (String[] args) throws IOException{
        int x;
        BufferedReader stdin = new BufferedReader
            (new InputStreamReader(System.in));
        System.out.print("Introduzca 0 o 1: ");
        x=Integer.parseInt(stdin.readLine());
        while (x!=0 || x!=1) {
            System.out.print("Opcion erronea, pruebe de nuevo\n");
            x= Integer.parseInt(stdin.readLine());
        }
    }
}
```

- a) Entra en el while cuando lee un valor de  $x$  distinto de 0 ó de 1.
- b) No entra en el while cuando lee un valor de  $x$  igual a 0
- c) Siempre entra en el bucle while
- d) No entra en el while cuando lee un valor de  $x$  igual a 1

Solución: *c*

22. ¿Cuál es la salida del siguiente programa si  $car=a$ ? ¿Y si  $car=t$ ?

```
class Ejercicio {
    public static void main (String[] args) {
        char car='1';
        car=args[0].charAt(0);
        while (car!='a' || car!='t') {
            car++;
        }
        System.out.println(car);
    }
}
```

- a) Si  $car$  es  $a$  o  $t$  imprime 2
- b) Siempre entra en el bucle while y se produce un bucle infinito
- c) Si  $car$  es  $a$  o  $t$  imprime 1
- d) Si  $car$  es  $a$  imprime 1 y si es  $t$  imprime 2

Solución: *b*

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

23. ¿Cuál es la salida del siguiente programa?

```
import java.io.*;
class Ejercicio {
    public static void main (String [ ] args) throws IOException{
        char car;
        BufferedReader stdin = new BufferedReader
            (new InputStreamReader(System.in));
        System.out.print("Introduzca SI o NO: ");
        car=stdin.readLine().charAt(0);
        while (car !='S' || car!='n') {
            System.out.print("Opcion erronea, pruebe de nuevo\n");
            car=stdin.readLine().charAt(0);
        }
    }
}
```

- a) Entra en el while solamente cuando lee SI
- b) Entra en el while exclusivamente cuando lee SI o NO
- c) Entra en el while sólo cuando lee NO
- d) Siempre entra en el bucle while

Solución: *d*

24. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String[] args) {
        int r=2,c;
        while (r>=1){
            c=1;
            while(c<=2) {
                System.out.print (" "+(r%2==1?"1":"2"));
                ++c;
            }
            --r;
        }
    }
}
```

Solución: 2 2 1 1

25. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        int x=0,y=0;
        boolean sigue=true;
        while(sigue) {
            x=(int)Math.pow(2,2); //Math.pow(m,n) es m elevado a n
            switch (x) {
                case 4:
                    y=2;
                case 6:
                    y=4;
                    break;
                case 8:
                    y=8;
                    break;
                default:
                    y=0;
            }
            System.out.println (y);
            sigue=false;
        }
    }
}
```

- a) Daría error: bucle infinito
- b) 4
- c) 2
- d) 0

Solución: *b*

### 3.1.3. Bucles Do-While

26. ¿Cuál es la salida del siguiente código?

```
index=1;
do {
    index++;
    System.out.println (index);
} while (index < 10);
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

Solución:

```
2
3
4
5
6
7
8
9
10
```

27. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        char c='M';
        int m=0;
        boolean x=true;
        do{
            if (!x){
                c='m';
            }
            do {
                m=m*2+1;
            } while(m<2);
            x=false;
        }while(c!='m');
        System.out.println(m);
    }
}
```

Solución: 7

28. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [ ] args){
        int contador=0;
        do {
            System.out.println (contador);
            contador++;
        } while (contador <0);
    }
}
```

Solución: 0

29. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String[] args){
        int contador=5;
        if (contador!=5) {
            contador=contador+2;
        } else {
            do {
                System.out.println (contador);
                contador++;
            } while (contador <0);
        }
    }
}
```

- a) 7
- b) 6
- c) 0
- d) 5

Solución: *d*

30. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        int var=1;
        int m=1;
        boolean x=true;
        do{
            if (!x){
                var=2;
            }
            do {
                m++;
            } while(m<3);
            x=false;
        }while(var!= 2 );
        System.out.println(m);
    }
}
```

Solución: 4



### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

31. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        boolean fin=false;
        int i=3;
        do {
            if (i%2==0){
                fin=!fin;
            }
            System.out.print (i+" ");
            i++;
        } while (i<6 && !fin);
    }
}
```

- a) 3 4
- b) 3 4 5
- c) Error
- d) 3

Solución: *a*

32. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        double m=3.37;
        int cont=0,i=0;
        do {
            if ((m/2!=0) || (i/2)==0) {
                cont++;
                m++;
            }
        } while(cont<2);
        do {
            i++;
            m--;
        }while (i<1);
        System.out.printf ( "%.1f", m);
    }
}
```

- a) 4.4
- b) 5.4
- c) 4.37
- d) 5.37

Solución: *a*

33. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {  
    public static void main (String [ ] args){  
        int a=4;  
        if (a>=5) {  
            a--;  
        } else {  
            do {  
                a++;  
                System.out.println (a);  
            } while (a<4);  
        }  
    }  
}
```

- a) 4
- b) 3
- c) 5
- d) No imprime nada porque el bucle no se ejecuta debido a inicialmente a=4

Solución: *c*

34. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio{  
    public static void main (String [] args) {  
        double a=3.22;  
        int i=0,j=0;  
        while (i<2) {  
            if ((a/2!=0) || (j/2)==0) {  
                i++;  
                a++;  
            }  
        }  
    }  
}
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

```
do {  
    j++;  
    a--;  
}while (j<1);  
System.out.printf ( "%.1f", a);  
}  
}
```

- a) 4.22
- b) 4.2
- c) 5.2
- d) 5.22

Solución: *b*

#### 3.1.4. Bucles For

35. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {  
    public static void main (String [] args) {  
        for (int i=0; i<12; i++)  
            switch (i) {  
                case 0:  
                case 1:  
                case 2:  
                case 3:  
                case 4:  
                    System.out.println (" i es menor que 5");  
                    break;  
                case 5:  
                case 6:  
                case 7:  
                case 8:  
                case 9:  
                    System.out.println (" i es menor que 10");  
                    break;  
                default:  
                    System.out.println (" i es 10 o mayor");  
            }  
        }  
    }  
}
```

Solución:

```
i es menor que 5
i es menor que 5
i es menor que 5
i es menor que 5
i es menor que 5
i es menor que 10
i es menor que 10
i es menor que 10
i es menor que 10
i es menor que 10
i es 10 o mayor
i es 10 o mayor
```

36. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        int m=5;
        {
            int n;
            if (Integer.parseInt(args[0]) == 7)
                n=7;
        }
        for (int i=0; i<n; i++)
            System.out.println (i);
    }
}
```

Solución:

Da error de compilación porque n sólo está definida dentro del bloque que la engloba. No puede encontrar el símbolo n porque está declarado en el primer bloque y no es visto en el bloque del for.

37. ¿Cuántos asteriscos imprimirá el bucle siguiente? ¿Por qué?

```
for (int star = 9; star<0; star++)
    System.out.print('*');
```

Solución: No imprime nada porque no se cumple la condición del bucle

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

38. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        for (int i=0; i<=7; i++)
            switch (i) {
                case 0:
                case 1:
                case 2:
                    System.out.println (i+" es menor que 3");
                    break;
                case 3:
                case 4:
                case 5:
                    System.out.println (i+" es menor que 6");
                    break;
                default:
                    System.out.println (i+" es 6 o mayor");
            }
    }
}
```

Solución:

```
0 es menor que 3
1 es menor que 3
2 es menor que 3
3 es menor que 6
4 es menor que 6
5 es menor que 6
6 es 6 o mayor
7 es 6 o mayor
```

39. ¿Cuál es la salida del siguiente programa? ¿Por qué?

```
class Ejercicio {
    public static void main (String [ ] args){
        int contador;
        for (contador=1; contador>4; contador++)
            System.out.println(contador);
    }
}
```

Solución: No imprime nada porque no se cumple la condición del bucle

40. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [ ] args){
        for (int i=1; i<4; i++) {
            switch (i) {
                case 1:
                    System.out.print ("a");
                    break;
                case 2:
                    System.out.print ("b");
                case 3:
                    System.out.print ("c");
                    break;
                default:
                    System.out.print ("d");
            }
        }
    }
}
```

- a) Daría error porque falta la sentencia break dentro de case 2
- b) abcd
- c) abc
- d) abcc

Solución: *d*

41. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String [] args) {
        for (int i=5; i>1; i--) {
            for (int j=0; j<i; j++) {
                System.out.print("*");
            }
            System.out.println(" ");
        }
    }
}
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

a) \*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

b)

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

c) \*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

d) Ninguna de las anteriores

Solución: *a*

42. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main(String [] args) {
        for (int i=0; i<=6; i++)
            switch (i) {
                case 0:
                case 1:
                    break;
                case 2:
                    System.out.print("1 ");
                case 3:
                    break;
                case 5:
                    System.out.print("5 ");
                    break;
                default:
                    System.out.print("d ");
            }
    }
}
```

Solución: 1 d 5 d

43. ¿Cuál es la salida del siguiente fragmento de código?

```
for (int j = 0; j < 3; j++) {  
    switch (j) {  
        case 1:  
            System.out.print (1);  
            break;  
        case 2:  
            System.out.print (2);  
        default:  
            System.out.print(3);  
    }  
}
```

- a) 3123
- b) 1233
- c) 312
- d) 123

Solución: *a*

44. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {  
    public static void main (String [] args) {  
        for (int i=20;i<=26;i++){  
            if (i % 2 != 0){  
                System.out.println(i);  
            }  
        }  
    }  
}
```

- a) 21 23 25
- b) 22 24 26
- c) 20 21 22 23 24 25 26
- d) 21  
23  
25

Solución: *d*



### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

45. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {  
    public static void main (String [] args) {  
        int x=1;  
        for (int i=0;i<3; i++){  
            x=x+2*i;  
        }  
        System.out.println(x+" "+i);  
    }  
}
```

- a) 7 3
- b) 7 2
- c) 6 3
- d) Daría error: Variable no declarada

Solución: *d*

46. ¿Cuál es la salida de este programa? ¿Y si la variable a=1?

```
class Ejercicio {  
    public static void main (String [] args) {  
        int a=2;  
        int n=6;  
        int cont=0;  
        for (int i=0; i<n;i++) {  
            if ((a/2!=0) || (i/2)!=0) {  
                cont++;  
            }  
        }  
        System.out.println(cont);  
    }  
}
```

- a) Si a=2 imprime 6; Si a=1 imprime 4
- b) Si a=2 imprime 6; Si a=1 imprime 5
- c) Si a=2 imprime 12; Si a=1 imprime 4
- d) Si a=2 imprime 6; Si a=1 imprime 3

Solución: *a*

47. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main(String[] args){
        for (int i=0;i<3;i++){
            for (int j=0;j<i;j++) {
                System.out.print(" ");
            }
            for (int k=0;k<5-2*i;k++){
                System.out.print("x");
            }
            System.out.println();
        }
    }
}
```

- a) xxxxxx  
xxx  
x
- b) xxxxxx  
xxx  
x
- c) xxxxxx  
xxx
- d) xxxxxx  
x

Solución: *b*

48. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main(String [] args) {
        for (int i=0; i<=5; i++) {
            switch (i) {
                case 0:
                case 1:
                    System.out.print("++ ");
                    break;
                case 3:
                case 4:
                    System.out.print("$$ ");
            }
        }
    }
}
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

```
        break;
    default:
        System.out.print("@@ ");
    }
}
}
```

- a) Error: No puede haber ningún case vacío
- b) Error: falta una sentencia break después de cada case
- c) ++ ++ @@ \$\$ \$\$ @@
- d) ++ ++ \$\$ \$\$ @@ \$\$

Solución: *c*

49. ¿Cuál es la salida de este programa? ¿Y si la variable x=5?

```
class Ejercicio{
    public static void main (String [] args) {
        int x=4;
        int y=4;
        int j=0;
        for (int i=0; i<y;i++) {
            if ((x/2!=0) || (i/2)!=0) {
                j--;
            }
        }
        System.out.println(j);
    }
}
```

- a) Si x=4 imprime -4; Si x=5 imprime -6
- b) Si x=4 imprime -4; Si x=5 imprime -4
- c) Si x=4 imprime 6; Si x=5 imprime 4
- d) Si x=4 imprime 4; Si x=5 imprime 4

Solución: *c*

50. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main(String [] args) {
        int i=0,k=0;
        for (i=0;i<2;i++){
            for (int j=0; j<2; j++){
                k=k+(i+j);
            }
        }
        System.out.println(i + " + " + j + " = " + k);
    }
}
```

- a)  $2 + 2 = 4$
- b)  $2 + 2 = 2$
- c)  $0 + 0 = 0$
- d) Error

Solución: *d*. La variable *j* no está declarada fuera del bucle donde está el println

51. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main(String [] args) {
        int r,z=0;
        for (r=3;r<=4;r++){
            for (int p=0; p<2; p++){
                z=z+(p+r);
            }
        }
        System.out.println(p + " " + r + " " + z);
    }
}
```

- a) 2 5 16
- b) Error
- c) 2 5 15
- d) 2 2 4

Solución: *b*. La variable *p* no está declarada fuera del bucle donde está el println

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

52. ¿Cuál es la salida de este programa? ¿Y si la variable  $m=5$ ?

```
class Ejercicio {
    public static void main (String [] args) {
        int m=4;
        int n=3;
        int cont=0;
        for (int i=0; i<n;i++) {
            if ((m/2!=0) || (i/2)!=0) {
                cont++;
            }
        }
        System.out.println(cont);
    }
}
```

- a) Si  $m=4$  imprime 3; Si  $m=5$  imprime 4
- b) Si  $m=4$  imprime 4; Si  $m=5$  imprime 3
- c) Si  $m=4$  imprime 3; Si  $m=5$  imprime 3
- d) Si  $m=4$  imprime 6; Si  $m=5$  imprime 3

53. ¿Cuál es la salida del siguiente programa?

```
class Ejercicio {
    public static void main (String[] args) {
        int contador=1;
        for (;contador>4; contador++){
            System.out.print ("Contador dentro del bucle:"+contador);
        }
        System.out.println ("Contador fuera del bucle:"+contador);
    }
}
```

Solución: Contador fuera del bucle:1

54. ¿Cuál es la salida de este programa? ¿Y si la variable  $a$  fuera par?

```
class Ejercicio {
    public static void main (String [] args) {
        int a=7;
        int n=20;
        int cont=0;
        for (int var=0; var<n;var++) {
```

```

        if ((a%2==0) || (var%2) ==0) {
            cont++;
        }
    }
    System.out.println(cont);
}
}

```

Solución: 10; si  $a$  es par: 20

55. Completar el siguiente programa para que la salida sea  $\text{suma}=15$

```

class Ejercicio {
    public static void main (String[] args) {
        int suma;
        .....
        for (int j=1;j<=5;j++) {
            suma=suma+j;
        }
        System.out.print ("suma="+suma);
    }
}

```

Solución:  $\text{suma} = 0$

56. ¿Cuánto vale  $c$  al acabar el bucle for?

```

for (c=1;c>=0;c++) {
    System.out.println (c);
}

```

- a) 1
- b) El fragmento de código no es válido porque no se declara  $c$  en la cabecera del bucle for
- c) El fragmento de código no es válido porque el bucle for es infinito
- d) 2

Solución:  $c$

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

57. ¿Cuál es resultado del siguiente programa?

```
class Ejercicio {  
    public static void main (String [ ] args){  
        int j=1;  
        for (int i=0; i<5; i++) {  
            switch (i) {  
                case 0:  
                case 2:  
                    j=j+i;  
                    break;  
                case 1:  
                    j=j*i;  
                case 3:  
                    j=j-i;  
                    break;  
                default:  
                    j=i/j;  
            }  
        }  
        System.out.println (j);  
    }  
}
```

- a) Daría error porque falta la sentencia break dentro de case 0 y case 1
- b) 0
- c) -4
- d) Daría error porque en la parte default divide por 0

Solución: *c*

## Problemas

### 3.2.1. Sentencias de selección

1. Escribir un programa que reciba 2 números reales por línea de órdenes, compruebe cuál es el mayor y lo imprima. Si los dos números son iguales también lo debe indicar.

Solución:

- Análisis: La lectura se realizará por línea de órdenes. La tarea es comprobar cual es el número mayor y mostrarlo. Indicar si son iguales.
- Diseño: La lectura se realizará usando el argumento args del método main. Se usarán if-else anidados para que no se comprueben cada vez todas las condiciones.
- Implementación:

```
class Ejercicio {
    public static void main(String [] args) {
        double n1,n2;
        n1=Double.parseDouble(args[0]);
        n2=Double.parseDouble(args[1]);

        if (n1>n2) {
            System.out.println(n1);
        } else {
            if (n1<n2){
                System.out.println(n2);
            } else {
                System.out.println("Los dos numeros son iguales");
            }
        }
    }
}
```

2. Suponer que r1 y r2 son dos números reales. Escribir el código necesario para determinar si son iguales suponiendo que la precisión de la representación numérica es p.

Cuando la comparación se realiza con números en punto flotante la igualdad de dos números no se debe comparar directamente como `numero1==numero2`. Esto es



### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

porque sólo van a ser iguales si todos los bits que los representan son iguales, y en cálculos con números reales hay siempre error de redondeo. Lo que es necesario comprobar es si son muy parecidos, usando un valor límite adecuado a cada problema. En este caso, el límite viene dado por la precisión de la representación numérica,  $p$ .

Solución:

```
if (Math.abs (r1-r2) < p) {  
    System.out.println("Se consideran iguales");  
} else {  
    System.out.println("Se consideran distintos");  
}
```

**Aclaración:** Se ha usado el método valor absoluto (`abs`) de la clase `Math` que contiene métodos matemáticos. La clase `Math` está en el paquete `java.lang` y, por lo tanto, no hay que importarla explícitamente. Tomamos el valor absoluto para que no afecte el signo de la diferencia entre `r1` y `r2`.

3. Escriba un programa que calcule la frecuencia,  $v(s^{-1})$ , de oscilación de un péndulo dada su masa  $m$  y longitud  $l$ . La expresión correspondiente es

$$v = \frac{1}{2\pi} \sqrt{\frac{g}{l}}$$

donde  $g$  es la aceleración normal de la gravedad en el campo gravitatorio terrestre ( $9,8m/s^2$ ). Aunque la frecuencia es independiente de la masa, consideremos que si ésta es mayor de 1kg, el hilo del que cuelga la misma se romperá. El programa debe distinguir esta situación y calcular la frecuencia sólo si  $m < 1$  kg.

Solución:

*a)* Análisis

El enunciado nos sirve como documento de análisis indicando claramente la funcionalidad necesaria. Baste indicar que la lectura y la escritura se realizaran por teclado y pantalla, respectivamente.

*b)* Diseño

La distinción del caso de masa mayor de 1 kg se realizará con un `if`. El cálculo de la frecuencia se realizará con la expresión genérica, sólo si la masa es menor que 1 kg. La entrada y salida de información se realizará usando la clase `BufferedReader` y `System.out.println()`, respectivamente.

## c) Implementación

```

import java.io.*;
class Ejercicio {
    public static void main(String [] args) throws IOException {
        double l, m, frecuencia;
        final double g=9.8 ; // (m/s^2) Sistema internacional

        BufferedReader leer =new BufferedReader
            (new InputStreamReader(System.in));

        // Lectura inicial de datos
        System.out.println("Introduzca la masa del pendulo (kg):");
        m=Double.parseDouble(leer.readLine());

        if (m < 1.0 ) {
            System.out.println("Introduzca la longitud del" +
                               "pendulo (m):");
            l=Double.parseDouble(leer.readLine());

            frecuencia = Math.sqrt(g/l);
            frecuencia = frecuencia /(2.0*Math.PI);

            System.out.println("Frecuencia (1/s): " + frecuencia);
        }
        else {
            System.out.println("La masa debe ser menor de 1 kg");
        }
    }
}

```

## 3.2.2. Bucles While

4. Escribir un programa que calcule la suma de los cuadrados comprendidos entre 1 y N donde N es un entero que se lee por teclado.

**Solución:**

## a) Análisis

El enunciado define fielmente el análisis. Las labores de lectura y escritura se realizarán por teclado y por monitor, respectivamente. La tarea es única, evaluar un sumatorio de cuadrados.

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

#### b) Diseño

La entrada y salida de información se realizará con la clase `BufferedReader` y `System.out.println()`, respectivamente. El sumatorio de los cuadrados se evaluará con un bucle de tipo `while` y los cuadrados se evaluarán como productos ( $n^2 = n * n$ ).

#### c) Implementación

```
import java.io.*;
class Ejercicio {
    public static void main(String [] args) throws IOException {
        int n, n_2, i;
        BufferedReader leer =new BufferedReader
            (new InputStreamReader(System.in));

        // Lectura de datos
        System.out.println("Introduzca el numero n");
        n=Integer.parseInt(leer.readLine());
        System.out.println("Calculando la suma de cuadrados "
            +"desde 1 hasta "+n);

        // Suma de los cuadrados de 1 a n usando un bucle while
        i=1; // Variable usada como contador
        n_2=0;
        while (i<=n) {
            n_2=n_2+i*i;
            i++;    // Incremento del contador
        }

        // Salida de resultados
        System.out.println("Suma de los cuadrados: "+n_2);
    }
}
```

5. Desarrollar un programa que analice los resultados (calificaciones) de exámenes. El programa deberá:

- Aceptar una serie de puntuaciones
- Evaluar la media
- Determinar la nota máxima
- Mostrar la media y la nota máxima

**Solución:***a)* Análisis

Los puntos anteriores representan los requisitos iniciales del sistema. Estos requisitos se deben refinar pues hay puntos no claros. Por ejemplo, ¿la serie de notas tienen siempre la misma longitud o no? ¿De donde se leen los datos, por teclado o de un fichero? ¿La salida debe mostrarse con un formato determinado? ¿Los cálculos necesitan realizarse con alguna precisión?

Respuestas: La serie de notas no tiene siempre la misma longitud. Los datos se leen del teclado. El formato puede ser cualquiera y la media no tiene formato especial.

*b)* Diseño

¿Cómo se consiguen estos objetivos? ¿Cómo indicamos que no hay más notas? Varias posibilidades:

- Después de cada nota preguntar si hay alguna más
- Dar al principio el número total de notas
- Introducir algún carácter especial para indicar el fin de la serie

La opción más cómoda es la tercera. Para ello hay que saber cuál es el intervalo permitido de valores de notas y usar algo fuera de ese intervalo. Este valor se denomina valor centinela. Si no puede haber notas negativas usaremos  $-1$  como centinela.

*c)* Implementación

```
import java.io.*;

//=====
//  Determina la media y el maximo de un conjunto de notas.
//  Segunda version
//=====
class Ejercicio {

    public static void main (String[] args) throws IOException {

        int nota = 9999, count = 0, max = 0;
        float suma = 0, media;
        String nota_string;
        BufferedReader stdin = new BufferedReader
            (new InputStreamReader (System.in));
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

```
// Leer y procesar todas las notas
while (nota >= 0) {
    System.out.print ("Introduzca nota (-1 para acabar):");
    nota_string = stdin.readLine();
    nota = Integer.parseInt (nota_string);
    if (nota >= 0) {
        count = count + 1;
        suma = suma + nota;
        if (nota > max)
            max = nota;
    }
}

if (count == 0)
    System.out.println ("No se introdujeron notas validas");
else {
    media = suma / count
    System.out.println();
    System.out.println ("Numero de alumnos: " + count);
    System.out.println ("Nota media: " + media);
    System.out.println ("Nota maxima: " + max);
}
}
```

6. Realizar un programa que dado un numero entero introducido por línea de órdenes, calcule y visualice por pantalla la suma y el número de sus cifras y además muestre el resultado de invertir el orden de sus cifras.

**Solución:**

*a)* Análisis

La lectura se realizará por línea de órdenes. Se debe mostrar por pantalla suma, número de cifras y el número con sus cifras invertidas.

*b)* Diseño

La entrada de datos se hará utilizando el vector de cadenas *args*. Para realizar el ejercicio hará falta usar una condición para comprobar que sólo se introdujo un parámetro por línea de órdenes. Además se necesario un bucle para ir calculando las cifras del número.

c) Implementación:

```
class Ejercicio {
    public static void main (String args[ ]) {
        int num, resto;
        int cifras = 0;
        int suma = 0;

        if (args.length == 1) {
            num = Integer.parseInt(args[0]);

            //controlar si el numero es negativo.
            System.out.println("El numero es : " + num);
            System.out.print(" Su inverso es : ");

            while ( num > 0 ) {
                cifras++;
                resto = num % 10;
                suma = suma + resto;
                System.out.print(resto);
                num = num / 10;
            }
            System.out.println();
            System.out.println("El numero de cifras es: " + cifras);
            System.out.println("La suma de sus cifras es: " + suma);
        } else
            System.out.println("El programa solo " +
                               "admite un parametro");
    }
}
```

7. Escribir el código JAVA de un programa que admita como parámetro en la línea de órdenes un número  $n$  de tipo byte. Si el número de parámetros no es exactamente 1 debe escribir un mensaje de error y terminar. Si no hay error, el programa debe leer números de tipo double hasta que el usuario introduzca el 0.0 o hasta que se hayan leído  $n$  números. Después, se informará por pantalla del número total de datos leídos, y de la media aritmética de los mismos.

**Solución:**

a) Análisis

La lectura inicial se debe realizar por línea de órdenes y se debe mostrar un mensaje de error si el número de argumentos es distinto de uno. La lectura de

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

los demás números será por teclado. Se mostrará por pantalla el número total de datos leídos y su media aritmética.

#### b) Diseño

Se usarán los parámetros del main (args) y un objeto de la clase `BufferedReader` para realizar la lectura de datos. Para controlar el número de parámetros introducidos se usará una condición. También hará falta un bucle while que se repetirá hasta que se introduzca 0.0 o se lean n números.

#### c) Implementación

```
import java.io.*;
class Ejercicio {

    public static void main(String args[])throws IOException {

        // 1. COMPROBAR NUMERO DE PARAMETROS
        if (args.length != 1) {
            System.out.println("Error en el numero de parametros");
        }
        else {
            int n;           // Para convertir el parametro
            double lec;      // Para leer numeros
            double sum=0.0;  // Para acumular los numeros leidos
            int cont=0;      // Para contar los numeros leidos

            // 1. CONVERTIR EL PARAMETRO
            n = Byte.parseByte( args[0] );

            // 2. LEER EL PRIMER NUMERO
            BufferedReader entrada=new BufferedReader
                (new InputStreamReader(System.in));
            lec = Double.parseDouble(entrada.readLine());

            if (lec!=0){
                // 3. BUCLE DE LECTURA CON MAXIMO CONTROLADO POR CONTADOR
                while( lec != 0.0 && cont < n-1) {
                    // 3.1. USAR EL VALOR LEIDO
                    sum = sum + lec;
                    // 3.2. CONTAR
                    cont = cont + 1;
                    // 3.3. LEER SIGUIENTE
                    = Double.parseDouble(entrada.readLine());
                }
                // 4. CALCULAR Y MOSTRAR RESULTADOS
                double media = sum / cont;
            }
        }
    }
}
```

```
        System.out.println("Se han leído "+cont+" números.");
        System.out.println("Su media es de " + media);
    } else {
        System.out.println("leer vale inicialmente 0.0")
        System.out.println("\nFin del programa");
    }
}
}
```

8. Reestructurar el siguiente fragmento de código para evitar el uso de saltos incondicionales.

```
while (i < n) {
    j=Integer.parseInt(leer.readLine());
    if (j==-1) break;
    i++;
}
```

Solución:

```
j=0;
while (i<n && j != -1) {
    j= Integer.parseInt(leer.readLine());
    if (j!=-1){
        i++;
    }
}
```

9. Reescribir el siguiente fragmento de código sin usar la sentencia continue pero manteniendo la funcionalidad.

```
meses: while (m<=3) {
    m++;
    d=1;
    while (d<=5) {
        if (m==2 && d==4) continue meses;
        System.out.println(m+" "+d);
        d++;
    }
}
```



### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

Solución:

```
while (m<=3) {  
    m++;  
    d=1;  
    while (d<=5 && !(m==2 && d==4)) {  
        System.out.println(m+" "+d);  
        d++;  
    }  
}
```

10. Realizar un programa que calcule el número de anillos que aparecen en los dígitos de un número entero introducido por línea de órdenes. Se considera que los dígitos 0, 6 y 9 están formados por un anillo y el 8 por dos anillos.

Solución:

a) Análisis

La lectura se realizará por línea de órdenes. Se tendrán que controlar las apariciones de números con anillos.

b) Diseño

Para controlar los números con anillos se necesita selección en este caso se usará un if y un switch. Para obtener cada cifra del número se realizará un while.

c) Implementación

```
class Ejercicio {  
    public static void main (String[] args) {  
        int anillos=0, numero;  
        if (args.length==1) {  
            numero=Integer.parseInt(args[0]);  
            // Se toma el valor absoluto para poder usar numeros negativos  
            numero=Math.abs (numero);  
            // Si el numero es cero el numero de anillos es 1  
            if (numero==0)  
                anillos=1;  
            else {  
                while (numero>0) {  
                    switch (numero%10) {  
                        case 0: case 6: case 9: {  
                            anillos=anillos+1;  
                            break;  
                        }  
                    }  
                    numero=numero/10;  
                }  
            }  
        }  
    }  
}
```

```

        case 8:
            anillos=anillos+2;
        }
        numero=numero/10;
    }
    System.out.println ("Anillos: " +anillos);
} else
    System.out.println ("Numero de parametros" +
                        "introducidos incorrecto");
}
}

```

### 3.2.3. Bucles Do-While

11. Reestructure el siguiente fragmento de código para evitar el uso de saltos incondicionales.

```

j=0;
i=0;
do {
    i=metodo1();
    if (i==-1) break;
    j+=i;
} while (j<n);

```

Solución:

```

j=0;
i=0;
do {
    j+=i;
    i=metodo1();
} while (j<n && i != -1);

```

12. Para una disolución de un ácido débil,  $HA$ , cuya constante de disociación sea  $Ka$ , el  $pH$  viene dado por la expresión (aproximada):

$$pH \cong \frac{1}{2}(pKa - \log [HA])$$

donde  $pKa$  es el menos logaritmo decimal de  $Ka$ ,  $\log$  representa el logaritmo decimal y  $[HA]$  es la concentración molar (moles/litro) del ácido.

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

Escribir un programa en Java que acepte la constante de acidez de un ácido débil y luego pregunte por la concentración de la disolución, evaluando el pH. El programa debe solicitar un valor de concentración tras cada cálculo hasta que el usuario indique que no desea calcular el pH de ninguna nueva disolución.

Solución:

a) Análisis

La entrada y salida se realiza por teclado y monitor y la funcionalidad del programa queda definida en el enunciado.

b) Diseño

La entrada y salida de información se gestionarán usando la clase `BufferedReader` y `System.out.println()`, respectivamente. El pH se calculará de acuerdo a la expresión dada en el enunciado. La petición continuada de concentraciones se implementará con un bucle `while`, que se repetirá hasta que el usuario indique que no desea más cálculos. En este caso, pondremos a false la condición que controla el bucle.

c) Implementación

```
import java.io.*;
class Ejercicio {
    public static void main(String [] args) throws IOException {
        double Ka, pKa, c, pH;
        boolean sigue=true;        // Variable de control del bucle
        int opcion=0;

        BufferedReader leer = new BufferedReader
            (new InputStreamReader(System.in));

        // Lectura de datos
        System.out.println("Introduzca la constante de acidez:");
        Ka=Double.parseDouble(leer.readLine());

        /* logaritmo base 10 de x = 0.43429448 * logaritmo natural
        de x. */

        pKa=-0.43429448* Math.log(Ka);

        do {
            System.out.println();
            System.out.println("Concentracion (M):");
            c=Double.parseDouble(leer.readLine());
            pH= (pKa-0.43429448* Math.log(c))/2.0;
```

```

        System.out.println();
        System.out.println("pH: "+pH);
        System.out.println();
        System.out.println("Desea usar otra concentracion?");
        System.out.println("Teclee 1 para si, otra opcion para no");
        opcion=Integer.parseInt(leer.readLine());
        if (opcion!=1) {
            sigue=false;
        }
    } while(sigue);
}
}

```

13. Escribir el siguiente código usando un bucle do\_while:

```

index=1;
while (index<10) {
    System.out.println (index);
    index++;
}

```

Solución:

```

index=1;
do {
    System.out.println (index);
    index++;
} while (index < 10);

```

### 3.2.4. Bucles For

14. Escribir un programa que calcule el productorio de un número  $n$ .

Solución:

*a)* Análisis

El enunciado deja claro que el objetivo es el cálculo del productorio de un número  $n$  según la siguiente fórmula:

$$\prod_{i=1}^n i$$

Siendo  $N$  un número entero que se introduce por teclado.

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

b) Diseño

La lectura se realizará usando la clase `BufferedReader`. Para la iteración el enunciado indica que se use un bucle `for`.

c) Implementación

```
import java.io.*;

class Ejercicio {

    public static void main(String [] args) throws IOException {
        int n, producto, i;
        BufferedReader lee =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.print ("Introduzca el valor de N: ");
        n=Integer.parseInt(lee.readLine());
        producto=1;
        for (i=1;i<=n;i++){
            producto=producto*i;
        }
        System.out.println("Producto= "+producto);
    }
}
```

15. Realizar un programa que dado un día, mes y año introducidos como enteros por línea de órdenes, nos devuelva el número de días que restan hasta final de año.

Solución:

a) Análisis

La lectura se realizará por línea de órdenes. Para devolver el número de días que restan hasta final de año se deberá tener en cuenta todos los casos, es decir, que febrero tiene 28 días y que algunos meses tienen 30 días y otros 31. Además habrá que controlar si el año es bisiesto.

b) Diseño

Se utilizará una condición para comprobar que se han introducido los 3 argumentos por línea de órdenes. Se comprobará el mes que se introduce para ver el número de días que tiene ese mes. Además, hace falta un bucle para calcular el número de días que tienen el resto de los meses que faltan hasta final de año. Con otra condición se comprobará si el año es bisiesto.

c) Implementación:

```
import java.io.*;

class Ejercicio {
    public static void main (String argv[]) {
        int dia, mes, anio;
        int totaldias, i;

        if (argv.length == 3) {
            dia = Integer.parseInt(argv[0]);
            mes = Integer.parseInt(argv[1]);
            anio = Integer.parseInt(argv[2]);

            if (mes==2) totaldias = 28 - dia;
            else
                if ((mes==4) || (mes==6) || (mes==9) || (mes==11))
                    totaldias = 30 - dia;
                else totaldias = 31 - dia;

            for(i = mes+1; i <= 12; i++)
                switch (i) {
                    case 1 : case 3 : case 5 :
                    case 7 : case 8 : case 10 :
                    case 12 :      totaldias += 31; break;
                    case 2 :      totaldias += 28; break;
                    case 4 : case 6 :
                    case 9 : case 11 :      totaldias += 30; break;
                }

            if ((anio % 4 == 0) && (mes <= 2)) {
                totaldias++;
            }
            System.out.println("Faltan " + totaldias +
                               " dias para fin de anno.");
        }
        else { System.out.println("El programa " +
                                   "solo admite tres parametros"); }
    }
}
```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

16. Construir un programa que devuelva los distintos números primos existentes desde el 1 hasta un número entero introducido como parámetro por la línea de órdenes.

Solución:

```
class Ejercicio {
    public static void main (String args[]){
        //definicion de variables locales del programa
        int hasta;
        boolean primo;
        int numero,factor;

        //lectura del parametro introducido por linea de ordenes
        hasta=Integer.parseInt(args[0]);

        //mensaje inicial con los dos primeros numeros primos
        System.out.println("El numero 1 es primo.");
        System.out.println("El numero 2 es primo.");

        //calculo del resto de numeros primos
        for (numero=3; numero<=hasta; numero++) {
            factor=2;
            primo=true;
            while (factor<=numero/2 && primo){
                if (numero%factor==0)
                    primo=false;
                else factor++;
            }
            if (primo)
                System.out.println("El numero "+numero+" es primo.");
        }
    }
}
```

17. Escribir un programa que escriba las 10 primeras potencias de 2.

Solución:

```
class Ejercicio {
    public static void main (String [] args) {
        final int DOS=2;
        int potencia;
        potencia=1;

        System.out.println ("Las diez primeras potencias de 2:");
```

```
    for (int i=0; i<=9; i++) {  
        System.out.println ("2^"+i+": "+potencia);  
        potencia=potencia*DOS;  
    }  
}  
}
```

18. Construir un programa que imprima todos los números comprendidos entre el 30 y el 60.

Solución:

- a) Usando bucle for

```
class Ejercicio {  
    public static void main (String [] args) {  
        for (int i=30;i<=60;i++){  
            System.out.println(i);  
        }  
    }  
}
```

- b) Usando bucle while

```
class Ejercicio {  
    public static void main (String [] args) {  
        int i=60;  
        while (i>=30){  
            System.out.println(i);  
            i--;  
        }  
    }  
}
```

- c) Usando un bucle do-while

```
class Ejercicio {  
    public static void main (String [] args) {  
        int i=30;  
        do {  
            System.out.println(i);  
            i++;  
        } while (i<61);  
    }  
}
```



### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

19. Construir un programa que muestre los números impares comprendidos entre el 20 y el 40.

Solución:

```
class Ejercicio {
    public static void main (String [] args) {
        for (int i=20;i<=40;i++){
            if (i % 2 != 0){
                System.out.println(i);
            }
        }
    }
}
```

20. Escribir el siguiente código usando el bucle for:

```
index=1;
while (index<10) {
    index ++;
    System.out.println (index);
}
```

Solución:

```
for (index=2; index<11; index++)
    System.out.println (index);
```

21. Escribir un programa en Java que produzca la siguiente salida:

```
0101
011011
01110111
```

Utilizar una sola sentencia de impresión para el 0 y otra para el 1 que impriman un sólo 0 ó 1.

Solución:

```
class Ejercicio {
    public static void main(String [] args){
        for (int i=4; i<9;i=i+2){
            for (int j=1;j<i+1;j++){
                if (j==1 || j==(i/2)+1)
```

```

        System.out.print('0');
    else
        System.out.print('1');
    }
    System.out.println("");
}
}
}

```

Otra posible solución:

```

class Ejercicio {
    public static void main(String [] args){
        for(int i=1;i<=3;i++){
            for(int j=1;j<=2;j++){
                System.out.print('0');
                for(int k=1;k<=i;k++){
                    System.out.print('1');
                }
            }
            System.out.println();
        }
    }
}

```

22. El número armónico de orden  $n$  ( $H_n$ ) es la suma de los recíprocos de los  $n$  primeros números naturales:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{k=1}^n \frac{1}{k}$$

Escribir un programa en Java que solicite por teclado el orden,  $n$ , de un número armónico, calcule este último y lo imprima con cinco decimales de precisión. Para ello, el programa deberá comprobar que el valor  $n$  introducido cumple que  $n \geq 1$ . Si no es así volverá a solicitar el número  $n$  hasta que se introduzca un valor correcto.

Solución:

```

import java.util.*;
class Ejercicio {
    public static void main (String[] args){
        int n;
        double H=1;
        boolean seguir=true;
    }
}

```

### 3. ESTRUCTURAS DE CONTROL Y PROGRAMACIÓN ESTRUCTURADA

---

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

// Lectura de datos
do{
    System.out.println(
        "Introduzca el orden del numero armonico: ");
    n=leer.nextInt();
    if (n<1) {
        System.out.println(
            "El orden debe ser >0. Introduzcalo de nuevo");
    } else {
        seguir=false;
    }
}while (seguir);

// Calculando el numero armonico de orden n
for (int i=2; i<=n; i++) {
    H=H+1./i;
}
System.out.printf(Locale.US, "Hn= %9.5f \n", H);
}
```