

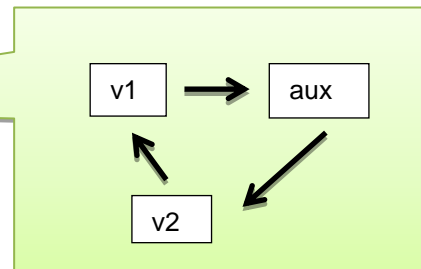
PROGRAMACIÓN MODULAR

21.- Dados dos valores numéricos reales almacenados en dos variables de nombre a y b, hacer un programa que llame a una función de nombre Cambiar(v1, v2) a la que se le pasan como parámetros dichas variables e intercambia sus valores, mostrando el resultado.

```

1  SubProceso Cambiar(v1, v2)
2      aux<-v1
3      v1<-v2
4      v2<-aux
5      Escribir "El primer valor es: ",v1
6      Escribir "El segundo valor es: ",v2
7  Fin SubProceso
8
9  Proceso principal
10     Escribir "Introduce el primer valor"
11     Leer a
12     Escribir "Introduce el segundo valor"
13     Leer b
14     Cambiar(a,b)
15 FinProceso

```



```

1  import java.util.Scanner;
2
3  public class Ejercicio21
4  {
5      static void cambiar(double v1, double v2)
6      {
7          double aux;
8          aux=v1;
9          v1=v2;
10         v2=aux;
11         System.out.println("El primer valor es :"+v1);
12         System.out.println("El segundo valor es :"+v2);
13     }
14     public static void main(String[] args)
15     {
16         double a,b;
17         Scanner teclado = new Scanner(System.in);
18         System.out.print("Introduce el primer valor: ");
19         a=teclado.nextDouble();
20         System.out.print("Introduce el segundo valor: ");
21         b=teclado.nextDouble();
22         cambiar(a,b);
23     }
24 }

```

22.- Escribe un programa con diseño modular que recoja por teclado dos números reales y presente una serie de opciones correspondientes a operaciones aritméticas a realizar con las mismas:

1. Sumar
2. Restar
3. Multiplicar
4. Dividir

En función de la operación elegida, se llama a la función correspondiente que retorna el valor del resultado de dicha operación, que se muestra en el programa principal.

```

1  SubProceso suma<-Sumar(v1, v2)
2      suma<-v1+v2
3  Fin SubProceso
4
5  SubProceso dif<-Restar(v1, v2)
6      dif<-v1-v2
7  Fin SubProceso
8
9  SubProceso producto<-Multiplicar(v1, v2)
10     producto<-v1*v2
11  Fin SubProceso
12
13 SubProceso division<-Dividir(v1, v2)
14     division<-v1/v2
15 Fin SubProceso
16
17 Proceso principal
18     Definir n1,n2,resultado como real
19     Definir opcion como entero
20     Escribir "Introduce el primer valor: "
21     Leer n1
22     Escribir "Introduce el segundo valor: "
23     Leer n2
24     //Mostrar menú
25     Escribir "1. Sumar"
26     Escribir "2. Restar"
27     Escribir "3. Multiplicar"
28     Escribir "4. Dividir"
29     Leer opcion
30     Segun opcion Hacer
31     ..... 1:
32             resultado<-Sumar(n1,n2)
33     ..... 2:
34             resultado<-Restar(n1,n2)
35     ..... 3:
36             resultado<-Multiplicar(n1,n2)
37     ..... 4:
38             resultado<-Dividir(n1,n2)
39     De Otro Modo:
40         Escribir "Opción incorrecta"
41     Fin Segun
42     Escribir "Resultado: ",resultado
43 FinProceso

```

```
1  import java.util.Scanner;
2  public class Ejercicio22
3  {
4      static double sumar(double v1, double v2)
5      {
6          double suma;
7          suma=v1+v2;
8          return suma;    // con una sólo sentencia return v1+v2;
9      }
10     static double restar(double v1, double v2)
11     {
12         return v1-v2;
13     }
14     static double multiplicar(double v1, double v2)
15     {
16         return v1*v2;
17     }
18     static double dividir(double v1, double v2)
19     {
20         return v1/v2;
21     }
22     public static void main(String[] args)
23     {
24         double resultado=0.0,n1,n2;
25         int opcion;
26         Scanner teclado = new Scanner(System.in);
27         System.out.print("Introduce el primer valor: ");
28         n1=teclado.nextDouble();
29         System.out.print("Introduce el segundo valor: ");
30         n2=teclado.nextDouble();
31         //Mostrar menú
32         System.out.println("1. Sumar");
33         System.out.println("2. Restar");
34         System.out.println("3. Multiplicar");
35         System.out.println("4. Dividir");
36         System.out.print("Opción elegida: ");
37         opcion=teclado.nextInt();
38         switch(opcion){
39             case 1:
40                 resultado=sumar(n1,n2);
41                 break;
42             case 2:
43                 resultado=restar(n1,n2);
44                 break;
45             case 3:
46                 resultado=multiplicar(n1,n2);
47                 break;
48             case 4:
49                 resultado=dividir(n1,n2);
50                 break;
51             default:
52                 System.out.println("Opción incorrecta");
53         }
54         System.out.println("Resultado: "+resultado);
55     }
56 }
```

Ejercicio 22b:

Modifica el programa para que siga pidiendo opciones hasta que introduzcamos un cero.

23.- Escribir un programa que utilice una función recursiva para calcular y retornar la potencia de un número real distinto de cero, siendo su exponente entero y positivo.

$$X^n = X * X^{n-1} \quad X^0 = 1$$

```

1  // Implementación del cálculo de una potencia mediante una función recursiva
2  // El paso recursivo se basa en que  $A^B = B * (A^{(B-1)})$ 
3  // El paso base se basa en que  $A^0 = 1$ 
4
5  SubProceso resultado <- Potencia (base, exponente)
6      Si exponente=0 Entonces
7          resultado <- 1;
8      sino
9          resultado <- base*Potencia(base,exponente-1);
10     FinSi
11 FinSubProceso
12
13 Proceso Principal
14     Escribir "Base"
15     Leer base
16     Escribir "Exponente"
17     Leer exponente
18     Escribir "El resultado es ",Potencia(base,exponente)
19 FinProceso

```

```

1  import java.util.Scanner;
2
3  public class Potencia
4  {
5      //función recursiva
6      static float potencia(float x, int n)
7      {
8          float resultado;
9          if (n==0)
10             resultado=1;
11          else
12             resultado = x * potencia(x, n-1);
13
14          return resultado;
15      }
16
17      public static void main(String[] args)
18      {
19          float base;
20          int exponente;
21          Scanner teclado = new Scanner(System.in);
22          System.out.print("Base: ");
23          base=teclado.nextFloat();
24          System.out.print("Exponente: ");
25          exponente=teclado.nextInt();
26          System.out.println("Potencia: "+potencia(base,exponente));
27
28      }
29
30 }

```

24.- Realizar un programa que disponga de una función llamada **esBisiesto** que reciba un año y devuelva cierto si es bisiesto y falso si no lo es.

Realizado en el Ejercicio 13: Los años que sean divisibles por 4 serán bisiestos; aunque no serán bisiestos si son divisibles entre 100 (como los años 1700, 1800, 1900, 2100) a no ser que sean divisibles por 400 (como los años 1600, 2000, 2400).

```

1  Subproceso bisiesto <- esBisiesto(anio)
2      Si (anio%4=0) & (anio%100<>0 | anio%400=0) Entonces
3          bisiesto<-Verdadero
4      Sino
5          bisiesto<-Falso
6      FinSi
7  FinSubProceso
8
9  Proceso Ejercicio24
10     Escribir "Introduce un año"
11     Leer anio
12     Si esBisiesto(anio) Entonces
13         Escribir "Es bisiesto"
14     Sino
15         Escribir "No es bisiesto"
16     FinSi
17 FinProceso

```

```

1  import java.util.Scanner;
2
3  public class Ejercicio24
4  {
5      static boolean esBisiesto(short año)
6      {
7          if ((año%4==0)&&((año%100!=0)||(año%400==0)))
8              return true;
9          else
10             return false;
11     }
12     public static void main(String[] args)
13     {
14         short año;
15         System.out.print("Introduce un año y te diré si es bisiesto: ");
16         Scanner teclado = new Scanner(System.in);
17         año = teclado.nextShort();
18         if (esBisiesto(año))
19             System.out.println("Es bisiesto");
20         else
21             System.out.println("No es bisiesto");
22     }
23 }

```

25.- Construir un programa para un proceso que lea un número entero mayor de 3 y menor de 3000 (se debe comprobar que se encuentre en este rango de valores, pidiendo otro valor en su caso) y nos muestre los números primos que sean menores que dicho número. Para la realización de este algoritmo se debe crear una función **esPrimo**, a la que se le envíe un número entero como parámetro y devuelva un 1 si el número enviado es primo y un 0 en el caso de que no lo sea.

Un número es primo si sólo es divisible por sí mismo y por 1.

```
1  import java.util.Scanner;
2
3  public class Ejercicio25
4  {
5      static byte esPrimo(int numero)
6      {
7          byte esprimo=1;
8          for (int i=2;i<numero;i++){
9              if (numero%i==0){
10                 esprimo=0;
11                 break;    //sale del bucle for en cuanto encuentra un divisor
12             }
13         }
14         return esprimo;
15     }
16
17     public static void main(String[] args)
18     {
19         short numero;
20         do{
21             System.out.print("Introduce un número (entre 3 y 3000): ");
22             Scanner teclado = new Scanner(System.in);
23             numero = teclado.nextShort();
24         }while (numero <=3 || numero >3000);
25         for(int i=numero-1;i>=1;i--){
26             if (esPrimo(i)==1)
27                 System.out.println(i);
28         }
29     }
30 }
```

En el ejercicio siguiente utilizaremos un bucle **while** para la resolución de este mismo problema.

26.- Realizar un programa que incluya una función llamada **siguientePrimo**. La función recibirá un parámetro que será un número entero y deberá devolver el siguiente número primo que sea mayor al número que se le ha enviado. En el caso de que el parámetro enviado fuera negativo o cero, deberá devolver un 1.

```
1  import java.util.Scanner;
2
3  public class Ejercicio26
4  {
5      //en este caso la función devuelve un boolean
6      static boolean esPrimo(int numero) {
7          boolean esprimo=true;
8          int i=2;
9          while (i<numero && esprimo){
10             if (numero%i==0)
11                 esprimo=false;
12             i++;
13         }
14         return esprimo;
15     }
16     static int siguientePrimo(int numero) {
17         if (numero <=0){
18             return 1;
19         }
20         else{
21             do {
22                 numero++;
23             }while (!esPrimo(numero));
24             return numero;
25         }
26     }
27     public static void main(String[] args) {
28         int numero,siguiente;
29         System.out.print("Introduce un número: ");
30         Scanner teclado = new Scanner(System.in);
31         numero = teclado.nextInt();
32         siguiente=siguientePrimo(numero);
33         System.out.println("Siguiente primo: "+siguiente);
34     }
35 }
```

No es necesario preguntar si
esprimo == true

puesto que es de tipo boolean y
su valor es true o false. Sería
como preguntar true==true

27.- Construir un programa que pida dos números enteros, obligatoriamente positivos, y nos muestre por pantalla los números perfectos que pudiera haber entre los mismos. El programa debe contar con una función llamada **esDivisor** que controle si un número es divisor de otro, devolviendo verdadero o falso en razón de que lo sea o no, y otra que se denomine **esPerfecto**, que reciba un número y que nos indique si es perfecto o no.

Un número es perfecto si la suma de sus divisores, excepto él mismo, es igual al propio número. (6 es perfecto; $1+2+3=6$)

```
1  import java.util.Scanner;
2
3  public class Ejercicio27
4  {
5      static boolean esDivisor(int numero1, int numero2)
6      {
7          if (numero1%numero2==0)
8              return true;
9          else
10             return false;
11     }
12
13     static boolean esPerfecto(int numero)
14     {
15         int suma=0;
16         for (int i=1;i<numero;i++){
17             if(esDivisor(numero,i))
18                 suma+=i;
19         }
20         if (suma==numero)
21             return true;
22         else
23             return false;
24     }
25
26     public static void main(String[] args)
27     {
28         int n1,n2;
29         Scanner teclado = new Scanner(System.in);
30         do{
31             System.out.print("Introduce un número positivo: ");
32             n1 = teclado.nextInt();
33             System.out.print("Introduce otro número positivo: ");
34             n2 = teclado.nextInt();
35         }while (n1<=0 || n2<=0);
36         System.out.println("Posibles números perfectos entre ambos:");
37         //intercambiar valores si n1>n2
38         if (n1>n2){
39             int aux=n1;
40             n1=n2;
41             n2=aux;
42         }
43         for (int i=n1;i<=n2;i++){
44             if (esPerfecto(i))
45                 System.out.print(i+" ");
46         }
47     }
48 }
```

Probar con 2 y 9000:

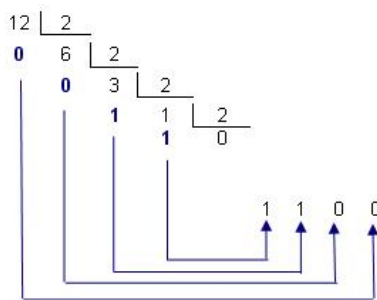
Resultado:
6, 28, 496, 8128

28.- Construir un programa que lea un número entero positivo comprendido entre 1 y 15 (debe controlarse este rango) y escriba tantas líneas como indique el número de la forma que se detalla a continuación.

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

```
1  import java.util.Scanner;  
2  
3  public class Ejercicio28  
4  {  
5      public static void main(String[] args)  
6      {  
7          byte numero;  
8          Scanner teclado = new Scanner(System.in);  
9          do{  
10             System.out.print("Introduce un número entero positivo entre 1 y 15: ");  
11             numero = teclado.nextByte();  
12             }while (numero<1 || numero>15);  
13             for (int i=1;i<=numero; i++){  
14                 for (int j=1;j<=i;j++){  
15                     System.out.print("* ");  
16                 }  
17                 System.out.println();  
18             }  
19         }  
20     }
```

29.- Codificar un programa que lea un número entero positivo en el sistema decimal y muestre el número en sistema binario. Utiliza una función **pasaBinario**.



Proceso: dividir entre 2 sucesivamente y tomar los restos en orden inverso

```

1  import java.util.Scanner;
2
3  public class Ejercicio29{
4
5      static long pasaBinario(int numDecimal){
6          int posicion=0, digito;
7          long numBinario=0;
8          while (numDecimal!=0){
9              digito=numDecimal%2;
10             numBinario=numBinario+digito*(long)Math.pow(10,posicion);
11             posicion++;
12             numDecimal=numDecimal/2;
13         }
14         return numBinario;
15     }
16
17     public static void main(String[] args)
18     {
19         int numero;
20         Scanner teclado = new Scanner(System.in);
21         do{
22             System.out.print("Introduce un numero entero positivo o nulo: ");
23             numero = teclado.nextInt();
24         }while(numero<0);
25         System.out.println("En binario es: "+ pasaBinario(numero));
26     }
27 }

```

30.- Realiza un programa que proporcione 15 resultados para rellenar una quiniela de futbol (posibles resultados 1, x, 2). Los resultados deben generarse de forma aleatoria.

```
1 public class Ejercicio30
2 {
3     static int generaNumero(){
4         int numero = (int)(1+Math.random()*3);
5         return numero;
6     }
7
8     public static void main(String[] args)
9     {
10         int resultado;
11         for (int i=1;i<=15;i++){
12             resultado=generaNumero();
13             if (resultado==3)
14                 System.out.println("x");
15             else
16                 System.out.println(resultado);
17         }
18     }
19
20 }
```

Math.random() devuelve un double entre 0.0 y 1.0

Para generar números entre 1 y 3:
 $1 + \text{Math.random()} * 3$