

**101.-** Diseñar un algoritmo para calcular el número de aprobados de una clase de 10 alumnos (se introduce una nota por cada alumno y suponemos que es correcta).

```
/**
 * Calcula el número de aprobados de una clase de 10 alumnos
 * Una notaDelAlumno por cada alumno
 */

import java.util.*; // permite E/S de datos

public class Prog101
{
    public static void main ( String[] args )
    {
        // Inicializamos variables
        Scanner teclado;
        teclado = new Scanner(System.in);
        byte notaDelAlumno;
        byte alumnosProcesados=0;
        byte contadorAprobados=0;

        // Procesamos 10 nota Del Alumnos
        while ( alumnosProcesados < 10)
        {
            alumnosProcesados++;
            do
            {
                System.out.print ("Introduce nota Del Alumno: ");          // Pedimos el dato por pantalla
                notaDelAlumno = teclado.nextByte();                        // Recogemos el dato por teclado
            }while ( notaDelAlumno<0 || notaDelAlumno > 10);                // Nos aseguramos de que sea un valor permitido

            if ( notaDelAlumno >= 5)          // Contamos los aprobados
            {
                contadorAprobados++ ;
            }
        } // fin del while

        // Mostramos el resultado final
        System.out.println ( "El número de aprobados es: " + contadorAprobados );

    }
}
```

**102.-** Construir un algoritmo que introduzca por teclado un número y a continuación pedir 10 números más. Visualizar todos aquellos que sean mayores que el inicial y cuantos son.

```
/**
 * Se pide un número y a continuación 10 números más.
 * Se visualizan los números mayores que el primero y se indica cuántos son.
 */
import java.util.*; // permite E/S de datos

public class Prog106
{
    public static void main ( String[] args )
    {
        // Inicializamos variables
        Scanner teclado;
        teclado = new Scanner(System.in); // Para poder pedir los datos por teclado

        int primerValor;
        int numComparado;
        int contMayores=0;
        byte i;

        // Pedimos el primer valor
        System.out.print ( "Introduce el valor para comparar con los demás: " );
        primerValor = teclado.nextInt();

        // Pedimos el resto y los procesamos
        for ( i=0 ; i<10 ; i++ )
        {
            System.out.print ( "Introduce el número " +(i+1)+ ": " );
            numComparado = teclado.nextInt();
            if ( numComparado > primerValor )
            {
                System.out.println ( numComparado+ "es mayor que el número inicial" );
                contMayores++;
            } // fin del if
        } // fin del for
        // Mostramos por pantalla el total
        System.out.println ();
        System.out.println ( "El total de números mayores que " +primerValor+ " es " +contMayores);
    }
}
```

**106.-** Se leen dos números enteros y positivos, que han de ser comprobados. Se considera el primero de ellos el dividendo y el segundo el divisor. Construir un algoritmo para calcular el cociente y el resto mediante restas sucesivas. Mostrar posteriormente los resultados.

```
/**
 * Calcula el cociente y el resto de la división de dos números enteros y positivos, mediante restas sucesivas
 */

import java.util.*; // permite E/S de datos

public class Prog104
{
    public static void main ( String[] args )
    {
        // Inicializamos variables
        Scanner teclado;
        teclado = new Scanner(System.in);

        int dividendo,divisor;
        int cociente,resto;

        // Pedimos y almacenamos los datos
        do
        {
            System.out.print("Introduce el dividendo: ");
            dividendo = teclado.nextInt();
        } while ( dividendo <= 0 );
        do
        {
            System.out.print("Introduce el divisor: ");
            divisor = teclado.nextInt();
        } while ( divisor <= 0 );

        // Asignamos valores
        resto = dividendo;
        cociente = 0;

        // Realizamos la división mediante restas sucesivas
        while ( resto >= divisor )
        {
            resto -= divisor;
            cociente++;
        }

        // Mostramos los resultados
        System.out.println ("La división de " +dividendo+ " entre " +divisor+ " es: ");
        System.out.println (" Cociente: " +cociente);
        System.out.println (" Resto: " +resto);
    }
}
```

**107.-** Construir un algoritmo para calcular la nota media de una serie de notas introducidas por teclado. Las notas tienen que ser mayores o iguales a cero y menores o iguales a 10. Cualquier otra nota se deberá rechazar. El proceso finaliza cuando se introduzca como nota el valor -1.

```
/**
 * Calcula la nota media de una serie de notas introducidas por teclado, con valores de 0 a 10. -1 es testigo de finalización.
 */
import java.util.*; // permite E/S de datos
public class Prog107
{
    public static void main ( String[] args )
    {
        // Inicializamos variables
        Scanner teclado;
        teclado = new Scanner(System.in); // Para poder pedir los datos por teclado

        byte nota;
        int cuentaNotas=0,sumaNotas=0;
        float notaMedia;          // No es necesaria, pero clarifica el código

        // Pedimos la primera nota
        do
        {
            System.out.print ( "Introduce nota entre 0 y 10 (-1 para terminar): " );
            nota = teclado.nextByte();
        }while ( nota<-1 || nota>10 );    //-1 no debe procesarse pero debe pasar

        // Mientras la nota no sea -1, procesamos los datos
        while ( nota != -1 )
        {
            cuentaNotas++;
            sumaNotas += nota;
            // Nos preparamos para la siguiente iteración
            do
            {
                System.out.print ( "Introduce nota entre 0 y 10 (-1 para terminar): " );
                nota = teclado.nextByte();
            }while ( nota<-1 || nota>10 );
        }

        // Si hubo notas, calculamos la media (EVITAR LA DIVISIÓN POR CERO)
        if ( cuentaNotas > 0 )
        {
            notaMedia = (float) sumaNotas/cuentaNotas;
            System.out.println ( "La nota media es " + notaMedia );
        }
        else
            System.out.println( "No hay notas para calcular la media." );
    }
}
```

112.- Construir un algoritmo que nos escriba los 10 primeros términos de la serie de Fibonacci. 1, 1, 2, 3, 5, 8, 13, ..... n (donde  $n = (n-1) + (n-2)$ )

Los dos primeros términos por definición son siempre 1, y el resto de los mismos se obtienen haciendo la suma de los dos términos anteriores.

```
/**
 * Calcula los 10 primeros términos de la serie de Fibonacci
 */

import java.util.*; // permite E/S de datos

public class Prog112
{
    public static void main ( String[] args )
    {
        // Declaramos las variables
        Scanner teclado;
        teclado = new Scanner( System.in );

        int terminoAnterior=1, terminoNuevo=1, terminoCalculado;
        byte i;

        System.out.println ( "Los 10 primeros términos de la serie de Fibonacci son: " );
        // Ya tenemos los dos primeros, los mostramos
        System.out.print ( terminoAnterior + " " + terminoNuevo );
        // Calculamos los nuevos términos
        for ( i=3 ; i<=10 ; i++ )
        {
            terminoCalculado = terminoNuevo + terminoAnterior;
            System.out.print ( " " + terminoCalculado );
            // El término recién calculado pasa a ser el anterior
            terminoAnterior = terminoNuevo;
            terminoNuevo = terminoCalculado;
        }
        System.out.println ( "" );
    }
}
```

114.- Construir un algoritmo que calcule el valor de la constante  $\pi$  (3,1415926...) a partir de la aproximación siguiente: (El número de términos será un entero positivo leído por teclado.)

$$\pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 - \dots)$$

```
/**
 * Calcula el valor de pi por aproximación. El número de términos es pedido por teclado.
 */

import java.util.*; // Permite E/S de datos

public class Prog114
{
    public static void main ( String[] args )
    {
        // Declaramos variables
        Scanner teclado;
        teclado = new Scanner(System.in);

        double pi=0;
        int numTer, i, denominador=1;
        byte signo;

        // Pedimos el número de términos por teclado
        do
        {
            System.out.print ( "Introduce número de términos (mayor que 10): " );
            numTer = teclado.nextInt();
        } while ( numTer <= 10 );

        // Calculamos la aproximación
        for ( i=1 ; i<=numTer ; i++ )
        {
            if ( i%2 != 0 )
                signo = 1;
            else
                signo = -1;

            pi += (double)signo / denominador;
            denominador += 2;
        }
        pi *= 4;

        // Mostramos los resultados por pantalla
        System.out.println ( "La aproximación de pi por " + numTer + " términos es: " + pi );
    }
}
```