Contenido

1. VARIABLES AUXILIARES DE UN PROGRAMA.

En bucles las usaremos muy a menudo.

Son objetos que utiliza un programa y por la función que realizan dentro del mismo toman un nombre especial, modelizando su funcionamiento debido a su frecuente utilización. En estas variables se almacenan datos, por ejemplo, **la suma de una serie de importes**, o **la cuenta del número de alumnos que hay en clase**, o **la nota máxima y mínima de un grupo de alumnos**, etc

Contadores.

Un contador es un objeto que se utiliza para **contar cualquier evento que pueda ocurrir dentro de un programa.** En general **suelen contar de forma natural desde 0 y de 1 en 1,** aunque se pueden realizar otros tipos de cuenta necesarios en algunos procesos.

Se utilizan realizando sobre ellos dos operaciones básicas:

- **Inicialización**: Todo contador se inicializa a 0 si realiza una cuenta natural, o a otro valor si se desea realizar otro tipo de cuenta.

```
int C1 = 0;
int CP = 2;
```

- **Contabilización o incremento:** Cada vez que aparece el evento a contar se ha de incrementar el contador en 1 si se realiza cuenta natural o en otro valor si se realiza otro tipo de cuenta.

```
C1 =  C1 + 1;
CP =  CP + 2;
```

Acumuladores.

Son objetos que se utilizan en un programa para **acumular elementos sucesivos con una misma operación**. En general se utilizan para calcular sumas y productos, sin descartar otros posibles tipos de acumulación.

Para utilizarlos se realizarán sobre ellos dos operaciones básicas:

- **Inicialización**: Todo acumulador se inicializa a 0 si realiza sumas y a 1 si realiza productos.

```
double SUMA = 0;
int PRODUC = 1;
```

- **Acumulación**: Cuando se hace presente en la memoria el elemento a acumular por la realización de una lectura o un cálculo, se efectúa la acumulación del mismo por medio de la asignación:

```
SUMA = SUMA + CANTIDAD;
PRODUC = PRODUC * NUMERO;
```

Máximos y Mínimos.

Un **máximo es una variable que guarda el máximo valor de un conjunto de valores**. El máximo se inicializa a un valor muy pequeño para que tengamos la seguridad de que el primer valor que se almacene en esta variable sea el mayor.

Un **mínimo es similar al máximo solo que coge el valor más pequeño de una serie de valores**. El mínimo se inicializa a un valor máximo.

```
int maximo = -9999;
int minimo = 9999
```

Ejemplo:

```
if(nota > maximo ) {
      maximo = nota;
}

if(nota < minimo ) {
      minimo = nota;
}
```

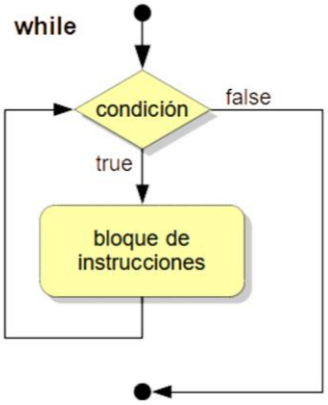
Switches. Banderas. Flags

Son variables que se utilizan en un programa y sólo pueden tomar dos valores (true y false, 0 y 1) realizando la función de transmitir información de un punto a otro dentro del programa. Se utilizan iniciándolos con un valor y en los puntos en que corresponda se cambian al valor contrario.

```
int sw = 0;

if(sw == 0) {
      //acciones
      sw = 1;
}
```

Por ejemplo, se pueden usar para pintar una cabecera solamente una vez dentro de un bucle. Si el switch es igual 0 se pinta la cabecera y se cambia el valor del switch para que no vuelva a pasar por ahí la ejecución y solo se pinte una vez la cabecera.

2. WHILE



1. Se evalúa condición.
2. Si la evaluación resulta true, se ejecuta el bloque de instrucciones.
3. Tras ejecutarse el bloque de instrucciones, se vuelve al primer punto.
4. Si, por el contrario, la condición es false, terminamos la ejecución del bucle.

```
while (condición) {
     bloque de instrucciones
     . . . . . .
}
```

Ejemplos:

```
int i = 1; //valor inicial
while(i <= 3 ) {
     System.out.println(i);
     i++;
}
```

Hacer la Traza de ejecución:

i	Salida en pantalla
1	

```
int suma = 0;
int c = 1;

while (c <= 5) {
     suma += c;
     c++;
}
System.out.println("Valor de c: " + c);
System.out.println("Valor de suma: " + suma);
```

Hacer la Traza de ejecución:

suma	c	Salida en pantalla
0	1	

EJERCICIO: MOSTRAR EL PRODUCTO DE LOS NÚMEROS DEL 1 AL N (N SE LEE DE TECLADO).

EJEMPLO DE:

Bucle que nunca se ejecuta:
```
int x = 0;
while (x > 0) {
      System.out.println(x);
      x++;
}
```

Bucle infinito: PELIGRO!!
```
int y = 0;
while (y >= 0) {
      System.out.println(y);
      y++;
}
```

Propuesta 3.1. Muestra la edad máxima y la edad mínima de un grupo de alumnos. El usuario introduce las edades y termina escribiendo un -1.

Ejemplo: Proceso repetitivo en el que se leen números de teclado. El proceso repetitivo finaliza cuando el número leído es 0. Antes de entrar en el proceso repetitivo leemos un número. Por cada número leído (DISTINTO DE 0) se pide:

- Visualizar su cuadrado y su cubo.
- Sumarlos a una variable **acumulador** que tendrás que inicializar a 0.
- Contar los números leidos, usa un **contador** que también tendrás que inicializar a 0.
- Después de realizar esas operaciones se vuelve a leer por teclado el número y serepite el proceso.

Al finalizar el proceso repetitivo se debe visualizar el número de números introducidos por teclado distintos de 0 y la suma de los números.

Solucion:
```
import java.util.Scanner;
public class whileSumas {
    static Scanner sc = new Scanner(System.in);

     public static void main(String[] args) {
          int N, SUMA, CUENTA, CUAD, CUBO;
          SUMA=0; CUENTA=0;
          System.out.print("Dame un número: ");
          N=sc.nextInt(); sc.nextLine();

          while (N !=0) {
                CUENTA = CUENTA + 1;
                SUMA = SUMA + N;
                CUAD = N * N;
                CUBO= CUAD * N;
                System.out.println("\tCuadrado: " + CUAD+ " Cubo: " + CUBO);
                System.out.print("Dame un número: ");
                N=sc.nextInt(); sc.nextLine();
          }

          System.out.println("SUMA: " + SUMA +
                             ", NUMEROS DISTINTOS 0: " + CUENTA);
     }//main
}//
```

Actividad resuelta 3.1

Diseñar un programa que muestre, para cada número introducido por teclado, si es par, si es positivo y su cuadrado. El proceso se repetirá hasta que el número introducido sea 0.

Actividad resuelta 3.2

Implementar una aplicación para calcular datos estadísticos de las edades de los alumnos de un centro educativo. Se introducirán datos hasta que uno de ellos sea negativo, y se mostrará: la suma de todas las edades introducidas, la media, el número de alumnos y cuántos son mayores de edad.

Actividad resuelta 3.3

Codificar el juego «el número secreto», que consiste en acertar un número entre 1 y 100 (generado aleatoriamente). Para ello se introduce por teclado una serie de números, para los que se indica: «mayor» o «menor», según sea mayor o menor con respecto al número secreto. El proceso termina cuando el usuario acierta o cuando se rinde (introduciendo un −1).

Actividad resuelta 3.4

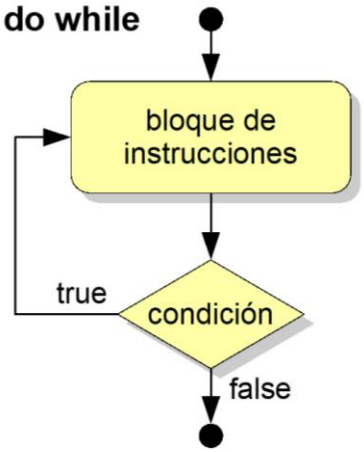
Un centro de investigación de la flora urbana necesita una aplicación que muestre cuál es el árbol más alto. Para ello se introducirá por teclado la altura (en centímetros) de cada árbol (terminando la introducción de datos cuando se utilice −1 como altura). Los árboles se identifican mediante etiquetas con números únicos correlativos, comenzando en 0. Diseñar una aplicación que resuelva el problema planteado.

- (Ejemplo1while.java) Proceso repetitivo donde se leerá un número de teclado, el proceso finalizará cuando el número sea > de 0. Este tipo de bucle se puede usar para validar la entrada por teclado de un número que cumpla una condición, en este caso que sea > que 0.
- (Ejemplo2while.java) Proceso repetitivo donde se leerá un número de teclado, el proceso finalizará cuando el número esté entre 1 y 7. Igual que antes, este tipo de bucle se puede usar para validar la entrada por teclado de un número que esté en un rango, en este caso entre 1 y 7.
- (Ejemplo3while.java) Proceso repetitivo en el que se pinta un menú, por ejemplo:
    1. Sumar Números
    2. Restar Números
    3. Multiplicar Números
    4. Dividir Números.
    5. Fin

  A continuación, se leen dos números enteros y la operación a realizar con ellos (1, para sumar, 2 para restar, 3 para multiplicar, 4 para dividir y 5 para terminar).
  Se realiza la operación y se muestra el resultado.
  Si la operación es la división y el denominador es 0, mostrar un mensaje y no realizar la división, asignarla valor 0.
  Si la operación es 5 el programa finaliza.
  ¿Qué condición tenemos que poner en el bucle para que el proceso se realice siempre y cuando la operación introducida no sea 5?

- (Ejemplo4while.java) Proceso repetitivo donde se leerán N números de teclado. N también se leerá y debe ser mayor que 0. Visualizar después de la lectura el máximo y el mínimo valor introducido por teclado.

3. DO-WHILE



1. Se ejecuta el bloque de instrucciones.
2. Se evalúa condición.
3. Según el valor obtenido, se termina el bloque o se vuelve al punto 1.

```
do {
    bloque de instrucciones
} while (condición);
```

Actividad resuelta 3.5

Desarrollar un juego que ayude a mejorar el cálculo mental de la suma. El jugador tendrá que introducir la solución de la suma de dos números aleatorios comprendidos entre 1 y 100. Mientras la solución introducida sea correcta, el juego continuará. En caso contrario, el programa terminará y mostrará el número de operaciones realizadas correctamente.

Usar **do while** en `whileSumas` (ejemplo resuelto del apartado anterior)

Proceso repetitivo en el que se leen números de teclado. El proceso finaliza cuando el número leído es 0. Por cada número leído (DISTINTO DE 0) se pide:

- Visualizar su cuadrado y su cubo.
- Sumarlos a una variable **acumulador** que tendrás que inicializar a 0.
- Contar los números leídos, usa un **contador** que también tendrás que inicializar a 0.

Al finalizar el proceso repetitivo se debe visualizar el número de números introducidos por teclado distintos de 0 y la suma de los números.

(dowhile20_30.java) Visualizar los números del 20 al 30, su cuadrado y su cubo, la salida es la siguiente:

```
Numero    Cuadrado       Cubo
==========================
    99      99.999    999.999
```

La salida debe mostrar:

```
Numero    Cuadrado       Cubo
==========================
    20         400     8.000
    21         441     9.261
    22         484    10.648
```

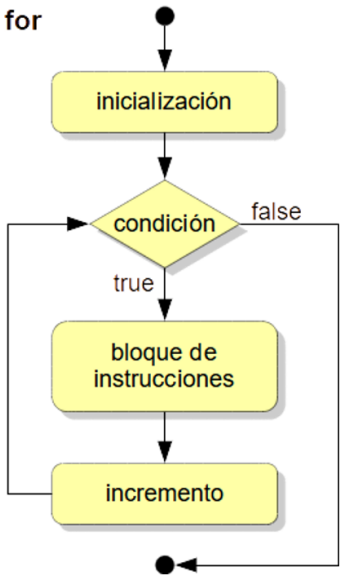
```
    23          529     12.167
    24          576     13.824
    25          625     15.625
    26          676     17.576
    27          729     19.683
    28          784     21.952
    29          841     24.389
    30          900     27.000
```

- (`datosAlumnos.java`) Realiza un programa Java para el siguiente proceso. Proceso repetitivo de lectura de datos de alumnos. Los datos a leer son: *nombre (String), curso (int 1 a 3) , sexo (1 carácter)*. Hay que controlar las siguientes situaciones:
  - ✓ Que el sexo tenga el valor H o M o h o m
  - ✓ Que el curso tenga el valor 1, 2 o 3

  Curso y sexo tiene que tener uno de esos valores, si no es así se volverán a pedir.

- ➢ Después de pedir estos datos se debe pedir por teclado si se desean introducir más datos de alumnos (visualizando un mensaje *"Desea introducir más datos (n/N): "* , si es **N** o **n**, finaliza el proceso, en caso contrario se repite el proceso de entrada de datos de alumnos.
- ➢ Al final visualizar el número de alumnos introducidos por teclado.
- ➢ Mostar al final el número de hombres y el número de mujeres.
- ➢ Mostrar al final número de alumnos de curso 1, de curso 2 y de curso 3.

4. FOR



1. Se ejecuta la inicialización.
2. Se evalúa la condición: false, salimos del bucle; en caso de que la evaluación sea true, se ejecuta todo el bloque de instrucciones.
3. Se ejecuta el incremento.
4. Se vuelve de nuevo al punto 2.

```
for (Inicialización; Condición; Incremento) {
   bloque de instrucciones
   . . . . . ..
};
```

Ejemplos:

```
for (int n = 0; n < 10; n++) {
     System.out.print(n + " ");
}
```

Visualiza: ***0 1 2 3 4 5 6 7 8 9***

Ejemplos (foriniciales.java**):**

Visualizar los números del 1 al 10.

Visualizar los números de 1 a N, donde N se lee de teclado. Comprobar que ocurre si N es menor o igual a 0.

Partes del bucle for

- **Inicialización** de la variable de control. La inicialización puede tener varias expresiones separadas por comas. También se pueden declarar variables en esta zona.
- **Condición**: condición que se debe de cumplir para que se realice el bucle y se ejecuten las sentencias. Normalmente es una comprobación del valor de la variable de control en una expresión condicional.
- **Incremento:** de la variable de control o de las variables que controlan el bucle. Puede tener varias expresiones separadas por comas.

```
int a, b;
for (a = 0, b = 0; a < 10; a++, b += 2) {
     System.out.print(a + " + " + b + ": ");
     System.out.println(a + b);
}
```

Visualiza:

```
0 + 0: 0
1 + 2: 3
2 + 4: 6
3 + 6: 9
4 + 8: 12
5 + 10: 15
6 + 12: 18
7 + 14: 21
```

```
8 + 16: 24
9 + 18: 27
```

Actividad propuesta 3.2

Implementa la aplicación Eco, que pide al usuario un número y muestra en pantalla la salida:

Eco...

Eco...

Eco...

Se muestra «Eco...» tantas veces como indique el número introducido. La salida anterior sería para el número 3.

Actividad resuelta 3.6

Escribir una aplicación para aprender a contar, que pedirá un número *n* y mostrará todos los números del 1 a *n*.

Actividad resuelta 3.7

Escribir todos los múltiplos de 7 menores que 100.

Actividad resuelta 3.8

Pedir diez números enteros por teclado y mostrar la media.

Actividad resuelta 3.9

Implementar una aplicación que pida al usuario un número comprendido entre 1 y 10. Hay que mostrar la tabla de multiplicar de dicho número, asegurándose de que el número introducido se encuentra en el rango establecido.

(Main3_9_2.java) MOSTRAR LA TABLA DE MULTIPLICAR DE LOS NÚMEROS COMPRENDIDOS ENTRE 1 Y N, DONDE N SE LEERA DE TECLADO. SE PUEDE HACER UNA FUNCIÓN QUE RECIBA UN NÚMERO Y MUESTRE SU TRABLA DE MULTIPLICAR

Actividad resuelta 3.10

Diseñar un programa que muestre la suma de los 10 primeros números impares.

(Main3_10_2.java) MOSTRAR LA SUMA DE LOS 10 PRIMEROS NÚMEROS IMPARES Y LA SUMA DE LOS 10 PRIMEROS NUMEROS PARES

Actividad resuelta 3.11

Pedir un número y calcular su factorial. Por ejemplo, el factorial de 5 se denota 5! y es igual a $5 \times 4 \times 3 \times 2 \times 1 = 120$.

Actividad resuelta 3.12

Pedir 5 calificaciones de alumnos y decir al final si hay algún suspenso.

Actividad resuelta 3.13

Dadas 6 notas, escribir la cantidad de alumnos aprobados, condicionados (nota igual a cuatro) y suspensos.

Propuesta 3.3

LEER UN NÚMERO ENTERO DE TECLADO Y MOSTRARLO AL REVES. POR EJEMPLO SI SE LEE `1234`, SE DEBE MOSTRAR `4321`. SI SE LEE `1000`, SE DEBE MOSTRAR `0001`

- (`for1Suma.java`) Realiza un programa que obtenga y visualice el resultado de esta suma: 1 + 2 + 3 + 4+ 5 + ….+ **n**. Donde **n** se leerá de teclado. Si **n** es negativo pasarlo a positivo. Si **n** es 0 no se realiza la suma, y se muestra mensaje indicándolo. Ejemplo de cómo debe de quedar la salida si se introduce por teclado el 10:

```
Número: 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55
```

- (`for2Factorial.java`) Leer un número entero de teclado y mostrar el factorial. Si el número es negativo pasarlo a positivo. El número tiene que ser >= 0 para poder calcular el factorial. Ejemplos de la salida del programa:

  Si se introduce un 0: `0! = 1`

  Si se introduce un 1: `1! = 1`

  Si se introduce un 4: `4! = 1 * 2 * 3 * 4 = 24`

  Si se introduce un 7: `7! = 1 * 2 * 3 * 4 * 5 * 6 * 7 = 5.040`

  Para cualquier valor de n: `n! = 1 * 2 * 3 * 4 * 5 * …. * n`

5. BREAK Y CONTINUE

break: interrumpe completamente la ejecución del bucle.

continue: detiene la iteración actual y continua la siguiente.

continue:

- Se emplea sólo en bucles
- Al ejecutarse la iteración en la que se encuentra, esta finaliza y se inicia la siguiente iteración.

```
public class TablaProducto1 {
   public static void main (String [] args) {
   int valor;
```

```
    valor = 8;
    System.out.println("Tabla de multiplicar del numero " + valor);
    for (int i=1; i<=10; i++) {
        if (i==5) continue;
        System.out.println(valor + " * " + i + " = " + valor*i );
        }
    }
}
```

Muestra la siguiente salida, en la que **no aparece el 5**, ya que si la i es 5 no se ejecutan las sentencias que hay debajo y comienza la siguiente iteración:

```
Tabla de multiplicar del numero 8
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
```

break:

- Se usa en sentencias **switch** o en bucles
- Deja el ámbito de la sentencia en la que se encuentra y pasa a la siguiente sentencia. Es decir, si se está ejecutando dentro de un bucle interrumpe completamente la ejecución del bucle.

```
public class TablaProducto2 {
    public static void main (String [] args) {
    int valor;
    valor = 8;
    System.out.println("Tabla de multiplicar del numero " + valor);
    for (int i=1; i<=10; i++) {
        if (i==5) break;
        System.out.println(valor + " * " + i + " = " + valor*i );
        }
    }
}
```

Muestra la siguiente salida, en la que no aparecen las multiplicaciones a partir del 5, ya que cuando la i es 5 el bucle for finaliza:

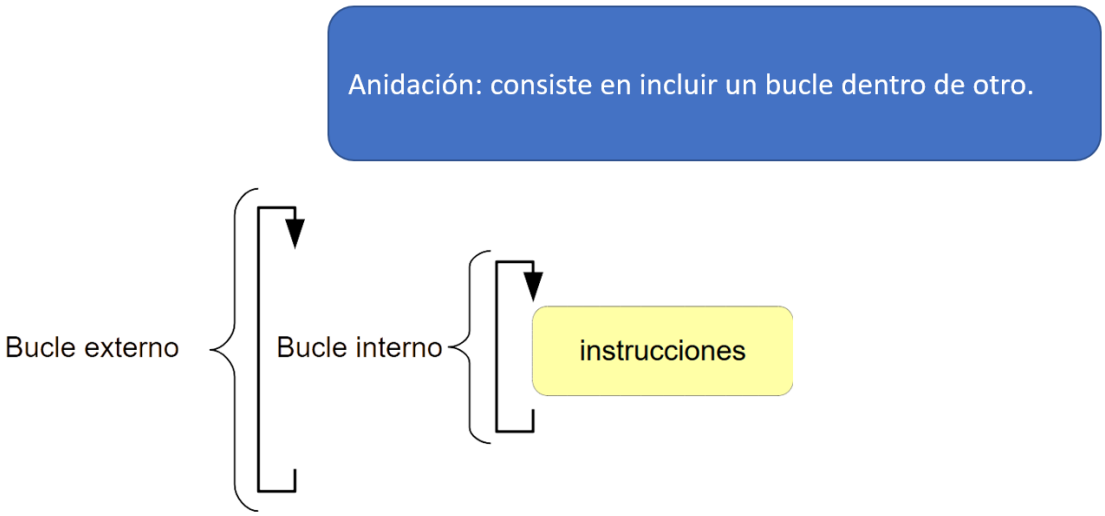
```
Tabla de multiplicar del numero 8
8 * 1 = 8
```

```
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
```

Ejemplo usando break:

```
import java.util.Scanner;

public class dowhileSumas {
	static Scanner sc = new Scanner(System.in);

	public static void main(String[] args) {
		int N, SUMA, CUENTA, CUAD, CUBO;
		SUMA = 0;
		CUENTA = 0;

		do {
			System.out.print("Dame un número: ");
			N = sc.nextInt();

			if (N == 0) break;

			SUMA = SUMA + N;
			CUAD = N * N;
			CUBO = CUAD * N;
			System.out.println("\tCuadrado: " + CUAD + " Cubo: " + CUBO);
			CUENTA = CUENTA + 1;

		} while (N != 0);

		System.out.println("SUMA: " + SUMA +
				", NUMEROS DISTINTOS de 0: " + CUENTA);
	}// main

}//
```

6. BUCLES ANIDADOS



Bucles independientes

LOS BUCLES ANIDADOS NO DEPENDEN UNOS DE OTROS PARA DETERMINAR EL NÚMERO DE ITERACIONES. EN ESTA CASO EL PRIMER BUCLE HACE 4 ITERACIONES Y EL SEGUNDO 3.

```
//bucle independiente
for (int i = 1; i <= 4; i++) {
   System.out.println("B.externo, iteración " + i);
   for (int j = 1; j <= 3; j++) {
     System.out.println("\tB.interno, iteración " + j);
   }
}
```

```
B.externo, iteración 1
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
B.externo, iteración 2
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
B.externo, iteración 3
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
B.externo, iteración 4
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
```

Bucles dependientes

EL NÚMERO DE ITERACIONES DE UN BUCLE INTERNO DEPENDE DE LOS BUCLES EXTERIORES, DE SUS VARIABLES DE CONTROL. EN ESTE CASO EL PRIMER BUCLE HACE 4 ITERACIONES Y EL NÚMERO DE ITERACIONES DEL SEGUNDO BUCLE DEPENDERÁ DE LA VARIABLE DE CONTROL DEL PRIMER BUCLE.

```
//bucle dependiente
for (int i = 1; i <= 4; i++) {
  System.out.println("B.externo, iteración " + i);
  for (int j = 1; j <= i; j++) {
    System.out.println("\tB.interno, iteración " + j);
  }
}
```

```
B.externo, iteración 1
        B.interno, iteración 1
B.externo, iteración 2
        B.interno, iteración 1
        B.interno, iteración 2
B.externo, iteración 3
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
B.externo, iteración 4
        B.interno, iteración 1
        B.interno, iteración 2
        B.interno, iteración 3
        B.interno, iteración 4
```

Se puede anidar cualquier tipo de bucle (for, while, do while)

- MOSTRAR EL FACTORIAL DE LOS NÚMEROS 1 A 10 DE LA SIGUIENTE MANERA:

```
1! = 1
2! = 1 * 2 = 2
3! = 1 * 2 * 3 = 6
4! = 1 * 2 * 3 * 4 = 24
5! = 1 * 2 * 3 * 4 * 5 = 120
6! = 1 * 2 * 3 * 4 * 5 * 6 = 720
7! = 1 * 2 * 3 * 4 * 5 * 6 * 7 = 5.040
8! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 = 40.320
9! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 = 362.880
10! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10 = 3.628.800
```

- CALCULAR LA TABLA DE MULTIPLICAR DE N NÚMEROS LEIDOS DE TECLADO. N TAMBIÉN SE LEERÁ DE TECLADO

Actividad resuelta 3.14

Diseñar una aplicación que muestre las tablas de multiplicar del 1 al 10.

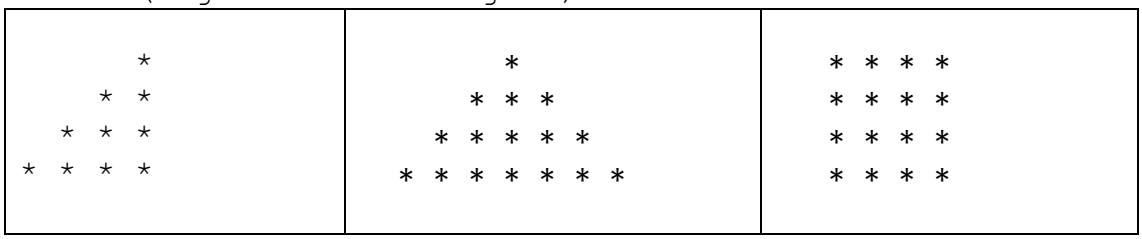
Actividad resuelta 3.15

Pedir por consola un número *n* y dibujar un triángulo rectángulo de *n* elementos de lado, utilizando para ello asteriscos (*). Por ejemplo, para *n* = 4:

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

- A CONTINUACIÓN hacer otras clases java y MOSTRAR las siguientes figuras para n = 4:

(figurasTrianCuad.java)



Hacerlo para cualquier valor de n. Usar una función que reciba el valor de n y pinte la figura.

7. EJERCICIOS

1. Proceso repetitivo donde se leerán los datos de **5 alumnos**. Los datos a leer son:
   Nombre de alumno.
   Número de asignaturas que tiene (entre 1 y 6, si no se cumple se lee de nuevo).

   Después de leer el número de asignaturas que tiene, tenemos que leer el nombre y la nota de cada asignatura (entre 1 y 10, si no se cumple se lee de nuevo). (hacer proceso repetitivo para la lectura de estos datos)
   Una vez leídas las notas y asignaturas mostraremos la nota media del alumno.

Visualizar al final del proceso de lectura de datos el nombre de alumno con mayor nota media y el nombre de alumno con menor nota media. Visualizar también su nota media.
(pseudocódigo al final)

MISMO EJERCICIO PERO AHORA VISUALIZAR TAMBIÉN AL FINAL ESTE LISTADO:

```
=========================================================
NOMBRE     Nº ASIG          ASIG MAS NOTA
========== ==============  =============
UNO             2             INF
DOS             3             INF
TRES            2             FRANCES
CUATRO          2             PROG
CINCO           2             MATE
=========================================================
```

2. Al ejercicio anterior se añade que hay que visualizar la asignatura con más nota para cada alumno. En lugar de ser 5 alumnos, que sean **N alumnos**, donde N se leerá de teclado. EJEMPLO DE salida:

```
NÚMERO DE ALUMNOS A LEER: 4
=========================================
ENTRADA DE DATOS PARA ALUMNO 1:
       Nombre de alumno: UNO
       Número de asignaturas (entre 1 y 6): 2
              Nombre de asignatura: MATES
              Nota (entre 1 y 10): 9
              --
              Nombre de asignatura: INF
              Nota (entre 1 y 10): 10
              --
       Nota media: 9,50
       Asignatura con más nota: INF
=========================================
ENTRADA DE DATOS PARA ALUMNO 2:
       Nombre de alumno: DOS
       Número de asignaturas (entre 1 y 6): 3
              Nombre de asignatura: LENGUA
              Nota (entre 1 y 10): 6
              --
              Nombre de asignatura: MATES
              Nota (entre 1 y 10): 7
              --
              Nombre de asignatura: INF
              Nota (entre 1 y 10): 9
              --
       Nota media: 7,33
       Asignatura con más nota: INF
=========================================
ENTRADA DE DATOS PARA ALUMNO 3:
       Nombre de alumno: TRES
       Número de asignaturas (entre 1 y 6): 2
              Nombre de asignatura: INGLES
              Nota (entre 1 y 10): 5
              --
              Nombre de asignatura: FRANCES
              Nota (entre 1 y 10): 8
              --
       Nota media: 6,50
       Asignatura con más nota: FRANCES
=========================================
ENTRADA DE DATOS PARA ALUMNO 4:
       Nombre de alumno: CUATRO
       Número de asignaturas (entre 1 y 6): 2
              Nombre de asignatura: BD
              Nota (entre 1 y 10): 7
              --
```

```
              Nombre de asignatura: PROG
              Nota (entre 1 y 10): 8
              --
       Nota media: 7,50
       Asignatura con más nota: PROG
=========================================================
NOMBRE     Nº ASIG         ASIG MAS NOTA
========== ==============  =============
UNO           2              INF
DOS           3              INF
TRES          2              FRANCES
CUATRO        2              PROG
=========================================================
Alumno con > nota media: UNO
       Nota media: 9,50
Alumno con < nota media: TRES
       Nota media: 6,50
```

3. Leer datos de departamentos en un proceso repetitivo hasta que el número de departamento leído sea menor o igual 0, los datos a leer son:

   Nº DE DEPARTAMENTO, NOMBRE, LOCALIDAD y Nº DE EMPLEADOS QUE TIENE.

   A continuación leer los datos de sus empleados (tantos como empleados tenga), los datos son:

   NºEMPLEADO, APELLIDO, OFICIO y SALARIO.

   - Visualizar <u>por cada departamento</u> el salario medio y el apellido con más salario.
   - Visualizar <u>al final del todo</u> el nombre de departamento con más empleados y el número de empleados que tiene
   - Visualizar el número de departamentos que se han introducido.
   - Si no se introducen departamentos (el primer número de departamento leído de teclado es 0) mostrar mensaje indicándolo, no se debe mostrar el máximo ni el nº de departamentos introducido.

   (pseudocódigo al final)

4. Realiza un programa Java que muestre esta serie de 0 y 1, usando bucles (seis filas y por cada fila 6 columnas):

```
1 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

5. Repite el ejercicio anterior, pero ahora el número de filas se introducirá por teclado.
6. Repetimos el ejercicio leyendo el número de filas y haciendo un proceso repetitivo que finalice cuando el número de filas sea menor o igual que 0.

7. Realiza un programa que visualice este rombo:

```
;;;;;-;;;;;    Los caracteres del rombo ( ;  - )  se
;;;;---;;;;    pedirán por teclado.
;;;-----;;;
```

```
;;-------;;
;---------;
-----------
;---------;
;;-------;;
;;;-----;;;
;;;;---;;;;
;;;;;-;;;;;
```

Crear función que reciba los caracteres y un valor numérico y pinte el rombo. Ejemplo de llamada para el rombo anterior: *PintaRombo*(';', '-', 5);

```
//llamada a función
PintaRombo('.', 'O', 6);
PintaRombo('*', '-', 7);
PintaRombo('x', '-', 8);
```

8. Hacer un programa Java que visualice la siguiente figura:

```
1
22
333
4444
55555
666666
. . . . .
NNNNNNNNNN
. . . . .
55555
4444
333
22
1
```

Donde N es un valor que se leerá por teclado y debe ser > 0 y < 10 para pintar la figura. Repetir el proceso hasta que N sea 0. Realizar funciones para simplificar el proceso.

9. Realizar un programa que dibuje un tablero de ajedrez con las letras B y N.

```
B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B
B N B N B N B N
N B N B N B N B
```

10. Visualizar **N** términos de esta serie, **N** se leerá de teclado (controlar excepciones al leer el número):

```
A1, AA2, AAA3, AAAA4, AAAAA5, ........  AAAAAA......N
```

Por ejemplo, si N es 4 se debe mostrar: `A1, AA2, AAA3, AAAA4`

11. Pintar las 26 primeras letras, empezando en la letra 'a':

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

12. Pintar la siguiente serie para un numero **n** leído de teclado, por ejemplo para n= 5 pinta;

```
a
ab
abc
abcd
abcde
```

13. Realiza un programa Java de nombre **serieLetras** que lea de teclado un número N mayor que 0 y muestre en pantalla N términos de la siguiente serie separados por espacios en blanco:

```
a  ab  abc  abcd  abcde  a1  ab1  abc1  abcd1  abcde1  a2  ab2  abc2
abcd2  abcde2 …..
```

N términos

Si el número N no es > 0 se debe leer de nuevo.
Por ejemplo, Si N = 2 el programa debe mostrar 2 términos: `a  ab`
Proceso repetitivo en el que se lee N por teclado, finaliza cuando N es 0. Ejemplo de ejecución para distintos valores de N:

Introduce el n° de términos a mostrar: **5** **a ab abc abcd abcde** Introduce el n° de términos a mostrar: **8** **a ab abc abcd abcde a1 ab1 abc1**	Introduce el n° de términos a mostrar: **10** **a ab abc abcd abcde a1 ab1 abc1 abcd1 abcde1** Introduce un el n° de términos a mostrar: **11** **a ab abc abcd abcde a1 ab1 abc1 abcd1 abcde1 a2**

14. Partiendo del ejercicio anterior realiza los cambios necesarios para que además pida por teclado el número máximo de letras a mostrar, debe ser un número > 0, si no es >0 se pide de nuevo. Por ejemplo si el número máximo de letras es 4 se deben mostrar como máximo 4 letras en los grupos de letras (`abcd`), si es 7 se deben mostrar 7 letras (`abcdefg`). Observa los ejemplos de ejecución:

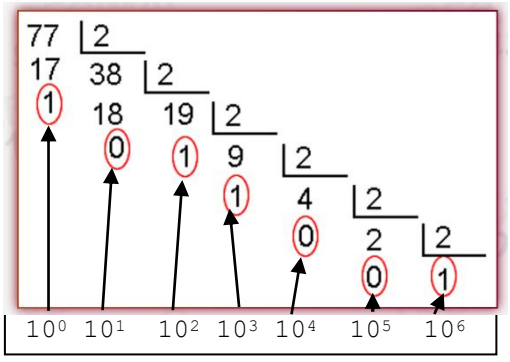
```
Introduce el n° de términos a mostrar: 6
Introduce el número máximo de letras: 4
a  ab  abc  abcd  a1  ab1

Introduce el n° de términos a mostrar: 12
Introduce el número máximo de letras: 1
a  a1  a2  a3  a4  a5  a6  a7  a8  a9  a10  a11

Introduce el n° de términos a mostrar: 7
Introduce el número máximo de letras: 10
a  ab  abc  abcd  abcde  abcdef  abcdefg

Introduce el n° de términos a mostrar: 15
Introduce el número máximo de letras: 7
```

a ab abc abcd abcde abcdef abcdefg a1 ab1 abc1 abcd1 abcd e1 abcdef1 abcdefg1 a2

15. Realiza un proceso repetitivo en el que se lea un número por teclado y se muestre el número de cifras que tiene. El proceso finaliza cuando se introduce el 0. Si el número es negativo se calcula el positivo.

16. Realiza un proceso repetitivo en el que se lea un número por teclado y se muestre en binario. El proceso finaliza cuando se introduce el 0. Si el número es negativo se calcula el positivo.

    Por ejemplo, el número 77 en binario se calcularía asi: dividiendo entre 2, quedándonos con el resto y cada resto multiplicarlo por 10^0 10^1 10^2 10^3 10^4 10^5 ….. y todo eso sumado:



$$1*10^0 + 0*10^1 + 1*10^2 + 1*10^3 + 0*10^4 + 0*10^5 + 1*10^6 =$$

$$1 + 0 + 100 + 1000 + 0 + 0 + 1000000 = \mathbf{1001101}$$

El 77 en binario es igual a: 1001101

17. Realiza un proceso repetitivo en el que se lea un número en binario y se pasa a decimal. El proceso finaliza cuando se introduce el 0.

    En este caso las divisiones se realizan entre 10 y los restos se van multiplicando por 2^0 2^1 2^2 2^3 2^4 2^5 2^6. … Ejemplo 111010101 da como valor 469.

18. Realiza una clase Java que pinte tres triángulos.

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

19. Realiza una clase Java que pinte N triángulos donde de altura H, N y H se leerán por teclado por teclado y deben ser mayores que 0 para que se realice el proceso.

```
Dame el número de triángulos (Mayor que 0): 4
Dame la altura (filas) del triángulo (mayor que 0): 3
  *     *     *     *
```

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

Dame el número de triángulos (Mayor que 0): 7
Dame la altura (filas) del triángulo (mayor que 0): 5
    *         *         *         *         *         *         *
   ***       ***       ***       ***       ***       ***       ***
  *****     *****     *****     *****     *****     *****     *****
 *******   *******   *******   *******   *******   *******   *******
********* ********* ********* ********* ********* ********* *********
```

EJERCICIO ALUMNOS

SOLUCION

```
Programa DATOSALUMNOS
Declaraciones
        Cadena: NOMBREALUM, NOMBREASIG, NOMBREMAXNOTA,NOMBREMINNOTA,ASIGMAX
        Entera: I, J, NUMASIG, MEDIA, NOTA, SUMANOTA, MEDMAX, MEDMIN, NOTAMAX
Inicio
         MEDMAX=-99, MEDMIN=15
        Para I = 1 hasta 5
                Visualizar("Nombre de ALUMNO: "), Leer NOMBREALUM
                Visualizar("Nº DE ASIGNATURAS: "),Leer NUMASIG
                SUMANOTA =0, NOTAMAX =-99
                SI NUMASIG > 0 Entonces
                        Para J = 1 hasta NUMASIG Hacer
                                Visualizar("Nombre de asignatura: "), Leer NOMBREASIG
                                Visualizar("Nota en la asignatura: "),Leer NOTA
                                SUMANOTA = SUMANOTA + NOTA
                                Si NOTA > NOTAMAX Entonces
                                   NOTAMAX = NOTA
                                   ASIGMAX = NOMBREASIG
                                Finsi
                        FinPara
                        MEDIA = SUMANOTA / NUMASIG
                        Si MEDIA > MEDMAX Entonces
                           MEDMAX = MEDIA
                           NOMBREMAXNOTA = NOMBREALUM
                        Finsi
                        Si MEDIA < MEDMIN Entonces
                              MEDMIN = MEDIA
                              NOMBREMINNOTA = NOMBREALUM
                        Finsi
                        Visualizar ("Nota media: ",MEDIA)
                        Visualizar ("ASIGNATURA CON MAS NOTA",ASIGMAX)
                        Visualizar (", LA NOTA ES",NOTAMAX)
                Finsi
        FinPara
        Visualizar ("ALUMNO CON MAS NOTA MEDIA:",NOMBREMAXNOTA)
        Visualizar (", SU NOTA MEDIA ES:",MEDMAX)
        Visualizar ("ALUMNO CON MENOS NOTA MEDIA:",NOMBREMINNOTA)
        Visualizar (", SU NOTA MEDIA ES:",MEDMIN)
Fin-programa
```

EJERCICIO DEPARTAMENTOS

SOLUCIÓN

```
Programa DATOSDEPARTAMENTOS
Declaraciones
        Cadena: NOMBREDEP, LOC, NOMBREEMPLE, NOMDEPMAX, OFICIO,
            APELLIDO, APELLIDOMAX
        Entera: NUMEMPLE, DEPTNO, EMPNO, SALARIO,  NUMEMPLEMAX,
            SALARIOMAX, I,   MEDIA, CONTADOR
Inicio
        NUMEMPLEMAX = -99, SALARIOMAX = -99, CONTADOR=0
        Visualizar("Número de departamento: "), Leer DEPTNO
        Mientras DEPTNO <> 0 HACER
                        Visualizar("Nombre de DEPARTAMENTO: "), Leer NOMBREDEP
                        Visualizar("LOCALIDAD "),Leer LOC
                        Visualizar("Nº DE EMPLEADOS: "), Leer NUMEMPLE
                        //
                        SI NUMEMPLE > NUMEMPLEMAX Entonces
                                        NUMEMPLEMAX = NUMEMPLE
                                        NOMDEPMAX = NOMBREDEP
                        FINSI

                        SUMASALARIO = 0, SALARIOMAX = -99
                        CONTADOR = CONTADOR+1
                        SI NUMEMPLE > 0 Entonces
                                Para I = 1 hasta NUMEMPLE Hacer
                                        Visualizar("Nº empleado: "), Leer EMPNO
                                        Visualizar("Apellido: "), Leer APELLIDO
                                        Visualizar("Oficio: "), Leer OFICIO
                                        Visualizar("Salario: "), Leer SALARIO
                                        SUMASALARIO = SUMASALARIO + SALARIO
                                        SI SALARIO > SALARIOMAX Entonces
                                                        SALARIOMAX = SALARIO
                                                        APELLIDOMAX= APELLIDO
                                        FINSI
                                FinPara
                                MEDIA = SUMASALARIO / NUMEMPLE
                                Visualizar ("SALARIO MEDIO: ",MEDIA)
                                Visualizar("APELLIDO CON MAS SALARIO: ",APELLIDOMAX,
                                        " SU SALARIO ES: ",SALARIOMAX)
                        Finsi

                        Visualizar("Número de departamento: "), Leer DEPTNO
        Finmientras

        SI CONTADOR > 0 Entonces
                Visualizar("NOMBRE DE departamento CON MAS EMPLEADOS:
                ",NOMDEPMAX, " NUMERO DE EMPLEADOS: ",NUMEMPLEMAX)

                Visualizar("Numero de departamentos introducidos:", CONTADOR)
        Finsi


Fin-programa
```