



Docente: Mg. Bryan Fernández Ebrath.

Practica. Estructuras repetitivas.

Actividad del proyecto: Programación utilizando estructuras repetitivas.

Estructura de contenido

- Introducción.
- Objetivos.
- Procedimientos.
- Bibliografía.



Introducción

La presente practica se realizará procedimientos y ejercicios de programación relacionado con las estructuras repetitivas con el lenguaje de programación Java usando el entorno de desarrollo Netbeans para la codificación de los programas y ejecución de pruebas.

Realizar las practicas del lenguaje de programación Java es base requerida para ser desarrollador de programas o aplicaciones con uno de los lenguajes de programación más populares. Para el desarrollo del laboratorio es necesario que el estudiante tenga disponible un computador con acceso a internet.

Objetivos

Al completar esta práctica, usted podrá:

- ✓ Identificar el uso de variables contadoras y acumuladoras.
- ✓ Conocer el concepto y estructura repetitivas.
- ✓ Codificar algoritmos con condicionales y selectivas con el lenguaje de programación Java.



Procedimiento

Para el desarrollo de las practicas es necesario tener en cuenta los siguientes conceptos.

Variables contadoras

Una variable contadora en algoritmos hace precisamente esto mismo, lleva la contabilidad de repeticiones, eventos, accesos, etc... Lo importante acá es que en el contador incrementa o decremento de forma constante. Los ciclos para llevan un contador interno que mide el avance de las repeticiones.

$$con = con + 1$$

$$con = con - 1$$

Ejemplo: Suponga que está en la entrada de un evento y su jefe le pide que le diga al final cuántas personas han venido al evento

Variables acumuladoras

Un acumulador en programación es una versión ampliada de un contador. El acumulador tiene las mismas características que un contador excepto el valor de incremento que es un valor **variable**.

$$acu = acu + (valor)$$

Ejemplo: una cuenta de ahorros puede representarse en un algoritmo mediante un acumulador, pues la persona no siempre podrá ahorrar una cantidad fija en la cuenta, un día deposita 10, otro día deposita 30, otro deposita 5.

Teniendo en cuenta los conceptos de contadora y acumuladora, llenar la siguiente tabla indicando si se debe aplicar contadora o acumuladora

Situación	Tipo de variable
Cantidad de hijos	
Ahorro en un banco	
Cantidad de estudiantes en el salón	
Monto recolectado para donar	
Números de partidos ganados de futbol	
Recarga de tarjeta de transporte	

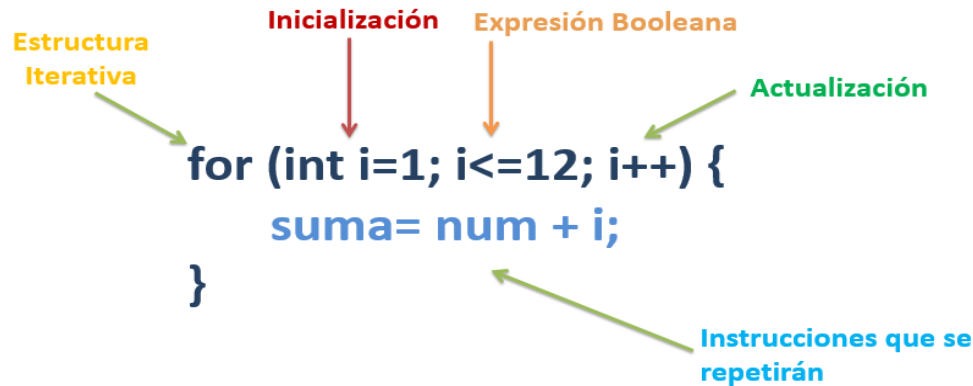
Estructuras repetitivas

Las estructuras de control repetitivas son aquellas que permiten ejecutar un conjunto de instrucciones varias veces, de acuerdo al valor que genere la expresión relacional y/o lógica. Esta estructura repetitiva se utiliza principalmente cuando no se conoce el número de veces que las acciones o el ciclo deben repetirse, aunque también puede utilizarse en otros casos. Las estructuras repetitivas, permiten repetir una acción (o grupo de acciones) varias veces.

Las estructuras repetitivas en Java son

- ✓ Estructura **for**
- ✓ Estructura **while**
- ✓ Estructura **do while**

Estructura for



En su forma más típica y básica, esta estructura requiere una variable entera que cumple la función de un CONTADOR de vueltas. En la sección indicada como "iniciación contadora", se suele colocar el nombre de la variable que hará de contador, asignándole a dicha variable un valor inicial. En la sección de "condición" se coloca la condición que deberá ser verdadera para que el ciclo continúe (en caso de un falso, el ciclo se detendrá). Y finalmente, en la sección de "incremento contador" se coloca una instrucción que permite modificar el valor de la variable que hace de contador (para permitir que alguna vez la condición sea falsa). Cuando el ciclo comienza, antes de dar la primera vuelta, la variable del for toma el valor indicado en la sección de "iniciación contadora". Inmediatamente se verifica, en forma automática, si la condición es verdadera. En caso de serlo se ejecuta el bloque de operaciones del ciclo, y al finalizar el mismo se ejecuta la instrucción que se haya colocado en la tercera sección. Seguidamente, se vuelve a controlar el valor de la condición, y así prosigue hasta que dicha condición entregue un falso.

Ejemplo

Realizar un algoritmo que permita obtener la edad de N personas entre hombres y mujeres. El valor de cada edad debe ser acumulado para los siguientes cálculos

- ✓ Cantidad de hombres y promedio de edad de hombres
- ✓ Cantidad de mujeres y promedio de edad de hombres
- ✓ Promedio general de edad.

```
1  Algoritmo sin_titulo
2      con_h←0
3      acu_h←0
4      Escribir "digite cantidad de personas"
5      leer n
6      Para i←1 Hasta n Con Paso 1 Hacer
7          Escribir "digite edad de la persona ",i
8          leer edad
9          escribir "1.- hombre o 2.- mujer"
10         leer g
11         Si g=1 Entonces
12             con_h←con_h+1
13             acu_h←acu_h+edad
14         SiNo
15             Si g=2 Entonces
16                 con_m←con_m+1
17                 acu_m←acu_m+edad
18             Fin Si
19         Fin Si
20     Fin Para
21     Si con_h=0 Entonces
22         Escribir "no se registraron hombres"
23     SiNo
24         pro_h←acu_h/con_h
25     Fin Si
26     Si con_m=0 Entonces
27         Escribir "no se registraron mujeres"
28     SiNo
29         pro_m←acu_m/con_m
30     Fin Si
31     pro_g←(acu_h+acu_m)/(con_h+con_m)
32     Escribir "cantidad de hombres : ",con_h
33     Escribir "promedio de edad de hombres : ",pro_h
34     Escribir "cantidad de mujeres : ",con_m
35     Escribir "promedio de edad de mujeres : ",pro_m
36     Escribir "promedio general: ",pro_g
37
38 FinAlgoritmo
```



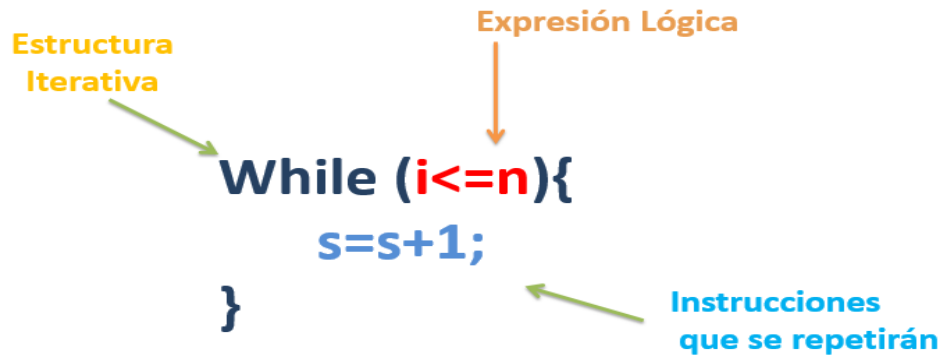
```
package javaapplication1;

import java.util.Scanner;

public class JavaApplication1 {

    public static void main(String[] args) {
        Scanner lectura = new Scanner(System.in);

        int n = 0, g = 0;
        double edad = 0, con_h = 0, con_m = 0, acu_h = 0;
        double acu_m = 0, pro_h = 0, pro_m = 0, pro_g = 0;
        System.out.println("Digite cantidad de personas");
        n = lectura.nextInt();
        for (int i = 0; i < n; i++) {
            System.out.println("digite edad de la persona " + i);
            edad = lectura.nextDouble();
            System.out.println("1.- hombre o 2.- mujer");
            g = lectura.nextInt();
            if (g == 1) {
                con_h = con_h + 1;
                acu_h = acu_h + edad;
            } else if (g == 2) {
                con_m = con_m + 1;
                acu_m = acu_m + edad;
            }
        }
        if (con_h == 0) {
            System.out.println("no se registraron hombres");
        } else {
            pro_h = acu_h / con_h;
        }
        if (con_m == 0) {
            System.out.println("no se registraron mujeres");
        } else {
            pro_m = acu_m / con_m;
        }
        pro_g = (acu_h + acu_m) / (con_h + con_m);
        System.out.println("cantidad de hombres : " + con_h);
        System.out.println("promedio de edad de hombres : " + pro_h);
        System.out.println("cantidad de mujeres : " + con_m);
        System.out.println("promedio de edad de mujeres : " + pro_m);
        System.out.println("promedio general : " + pro_g);
    }
}
```

Estructura **while**

While es una sentencia de control de flujo que permite que el código se ejecute repetidamente en función de una condición booleana dada. El bucle while se puede considerar como una instrucción if repetitiva.

El while comienza con la verificación de la condición. Si se evalúa como verdadero, las instrucciones del cuerpo del bucle se ejecutan; de lo contrario, se ejecuta la primera instrucción que le sigue al bucle. Por esta razón, también se llama bucle de control de entrada. Una vez que la condición se evalúa como verdadera, se ejecutan las instrucciones en el cuerpo del bucle.

Normalmente, las declaraciones contienen un valor de actualización para la variable que se procesa para la siguiente iteración.

Cuando la condición se vuelve falsa, el ciclo finaliza y marca el final de su ciclo de vida.

Ejemplo

Calcular el salario de un grupo de n trabajadores dada la cantidad de horas trabajadas y la tarifa por hora para cada uno de los trabajadores. Indicar cuantos hombres y mujeres se registraron.


```
1  Algoritmo sin_titulo
2      acu←0
3      Escribir "digite cantidad de trabajadores"
4      leer n
5      i←1
6      Mientras i≤n Hacer
7          Escribir "digite cantidad de horas "
8          leer h
9          Escribir "digite valor de la hora"
10         leer v
11         sueldo←v*h
12         acu=acu+sueldo
13         escribir "1.- hombre o 2.- mujer"
14         leer g
15         Si g=1 Entonces
16             con_h←con_h+1
17         SiNo
18             Si g=2 Entonces
19                 con_m←con_m+1
20             Fin Si
21         Fin Si
22         i←i+1
23     Fin Mientras
24     Escribir "acumulado de sueldos : ",acu
25     Escribir "cantidad de hombres : ",con_h
26     Escribir "cantidad de mujeres : ",con_m
27 FinAlgoritmo
```



```
package pkgwhile;

import java.util.Scanner;

public class While {

    public static void main(String[] args) {

        Scanner lectura = new Scanner(System.in);

        int n = 0, g = 0, i = 0;
        double con_h = 0, con_m = 0, acu = 0;
        double h = 0, v = 0, sueldo = 0;
        System.out.println("Digite cantidad de personas");
        n = lectura.nextInt();
        i = 1;
        while (i <= n) {
            System.out.println("Digite cantidad de horas ");
            h = lectura.nextDouble();
            System.out.println("Digite valor de horas ");
            v = lectura.nextDouble();
            sueldo = v * h;
            acu = acu + sueldo;
            System.out.println("1.- hombre o 2.- mujer");
            g = lectura.nextInt();
            if (g == 1) {
                con_h = con_h + 1;
            } else if (g == 2) {
                con_m = con_m + 1;
            }
            i = i + 1;
        }
        System.out.println("cantidad de hombres : " + con_h);
        System.out.println("cantidad de mujeres : " + con_m);
        System.out.println("Sueldo acumulado : " + acu);
    }
}
```

Estructura **do while**

Estructura
Iterativa

do{

s=s+1;

}

While (nota<=0 || nota>=20)

Instrucciones
que se repetirán

Expresión Lógica

El bucle **do while** es similar al **while** con la única diferencia de que comprueba la condición después de ejecutar las instrucciones.

El bucle **do while** comienza con la ejecución de la(s) declaración(es). No hay verificación de ninguna condición la primera vez.

Después de la ejecución de los enunciados, y la actualización del valor de la variable, la condición se verifica para el valor verdadero o falso. Si se evalúa como verdadero, comienza la siguiente iteración del ciclo.

Cuando la condición se vuelve falsa, el ciclo finaliza y marca el final de su ciclo de vida.

Es importante tener en cuenta que el bucle **do-while** ejecutará sus declaraciones al menos una vez antes de que se verifique cualquier condición, y por lo tanto es un ejemplo de bucle de control de salida.

Ejemplo

Mostrar una lista de opciones con 3 opciones:

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

y finalizar el algoritmo sí y solo sí se elija la opción para salir.

```
1  Algoritmo sin_titulo
2      Repetir
3          Escribir "MENU DE OPCIONES"
4          Escribir "1. SUMAR "
5          Escribir "2. RESTAR "
6          Escribir "3. MULTIPLICAR "
7          Escribir "4. DIVIDIR "
8          Escribir "0. SALIR"
9          leer opcion
10         Segun opcion Hacer
11             1:
12                 Escribir "digite numero 1"
13                 leer n1
14                 Escribir "digite numero 2"
15                 leer n2
16                 s←n1+n2
17                 Escribir "La suma es ",s
18             2:
19                 Escribir "digite numero 1"
20                 leer n1
21                 Escribir "digite numero 2"
22                 leer n2
23                 r←n1-n2
24                 Escribir "La resta es ",r
25             3:
26                 Escribir "digite numero 1"
27                 leer n1
28                 Escribir "digite numero 2"
29                 leer n2
30                 m←n1*n2
31                 Escribir "La multiplicacion es ",m
32             4:
33                 Escribir "digite numero 1"
34                 leer n1
35                 Escribir "digite numero 2"
36                 leer n2
37                 d←n1/n2
38                 Escribir "La division es ",d
39             0:
40                 Escribir "Gracias por usar el programa"
41         De Otro Modo:
42             Escribir "verificar datos ingresados"
43     Fin Segun
44     Hasta Que opcion = 0
45 FinAlgoritmo
```



```
package dowhile;

import java.util.Scanner;

class Dowhile {

    public static void main(String[] args) {
        Scanner lectura = new Scanner(System.in);

        int opcion = 0, g = 0;
        double n1 = 0, n2 = 0, s = 0;
        double r = 0, d = 0, m = 0;

        do {
            System.out.println("Menu de opciones \n"
                + "1.- suma \n"
                + "2.- resta \n"
                + "3.- multiplicacion \n"
                + "4.- division \n"
                + "0.- salir \n"
                + "DIGITE OPCION");

            opcion = lectura.nextInt();
            switch (opcion) {
                case 1:
                    System.out.println("digite numero 1");
                    n1 = lectura.nextDouble();
                    System.out.println("digite numero 2");
                    n2 = lectura.nextDouble();
                    s = n1 + n2;
                    System.out.println("la suma es " + s);
                    break;

                case 2:
                    System.out.println("digite numero 1");
                    n1 = lectura.nextDouble();
                    System.out.println("digite numero 2");
                    n2 = lectura.nextDouble();
                    r = n1 - n2;
                    System.out.println("la resta es " + r);
                    break;

                case 3:
                    System.out.println("digite numero 1");
                    n1 = lectura.nextDouble();
                    System.out.println("digite numero 2");
                    n2 = lectura.nextDouble();
                    m = n1 * n2;
                    System.out.println("la multiplicacion es " + m);
                    break;

                case 4:
                    System.out.println("digite numero 1");
                    n1 = lectura.nextDouble();
                    System.out.println("digite numero 2");
                    n2 = lectura.nextDouble();
                    if (n2 == 0) {
                        System.out.println("error en divisio entre cero");
                    } else {
                        d = n1 / n2;
                        System.out.println("la division es " + d);
                    }
                    break;
                case 0:
                    System.out.println("Gracias por usar el programa");
                    break;
                default:
                    System.out.println("verificar datos ingresados");
            }
        } while (opcion != 0);
    }
}
```



Codificar los siguientes algoritmos en el lenguaje de programación Java

Algoritmo 1

```
Proceso contadorDeSalarios|
    c50 <- 0
    c15 <- 0

    Para x <- 1 Hasta 50 Hacer
        Leer salario
        Si (salario >= 500000 Y salario <= 1000000) Entonces
            c50 <- c50 + 1
        Sino Si ( salario > 1500000) Entonces
            c15 <- c15 + 1
        FinSi
    FinPara

    Escribir "Empleados con salario entre 500000 y 1000000: ", c50,
    " empleados con salario superior a 1500000: ", c15

FinProceso
```

q

Algoritmo 2

```
Algoritmo Repetitivos05
    XPROM_MAY = 0

    Para Cont <-1 Hasta 5 Con Paso 1 Hacer
        Escribir "Ingrese Nombre ", Cont , " : "
        Leer NOMBRE
        Escribir "Ingrese Promedio ", Cont , " : "
        Leer PROMEDIO

        Si PROMEDIO >= 0 Y PROMEDIO <= 20 Entonces
            Si XPROM_MAY < PROMEDIO Entonces
                XPROM_MAY = PROMEDIO
                XNOM = NOMBRE
            Fin Si
        SiNo
            CONT = CONT - 1
        Fin Si
    Fin Para

    Escribir "ALUMNO : ", XNOM
    Escribir "PROMEDIO : ", XPROM_MAY

FinAlgoritmo
```

Algoritmo 3

```
Proceso SALARIODEEMPLEADORES
Escribir 'NUMEROS DE TRABAJADORES';
Leer N;
STE<-0;
I<-0;
EGM<-100000000;
Mientras I<N Hacer
    Escribir 'SUELDO BASICO MENSUAL';
    Leer SE;
    STE<-STE+SE;
    SI EGM>SE ENTONCES
        EGM<-SE;
    FINSI
    I<-I+1;
FinMientras
Escribir 'SALARIO TOTAL DE EMPLADORES';
Escribir STE;
Escribir 'EMPLEADOR QUE GANA MENOS';
Escribir EGM;
FinProceso
```

Algoritmo 4

```
Proceso sin titulo
    A<-0;
    Mientras A>10 Hacer
        A<-A+1;
        B<-0;
        Mientras B>10 Hacer
            B<-B+1;
            Escribir A,'X',B,'=', 'A*B';
        FinMientras
    FinMientras
FinProceso
```

Algoritmo 5

```
Algoritmo Repetitivo11

  Escribir "Ingrese Números de 3 Cifras"

  Para Cont <-1 Hasta 30 Con Paso 1 Hacer

    Escribir "Ingrese Número " Cont " : "
    Leer Num

    Si Num <= 999 Entonces
      C1 = (Num - (Num MOD 100)) / 100
      R1 = Num MOD 100
      C2 = (R1 - (R1 MOD 10)) / 10
      R2 = R1 MOD 10

      Escribir R2 " " C2 " " C1
    SiNo
      Cont = Cont - 1
    Fin Si
  Fin Para

FinAlgoritmo
```

Algoritmo 6

```
Algoritmo ejercicio2repetir
  Sum<-0
  Repetir
    Escribir "Ingrese el precio del articulo tomado"
    Leer articulo
    Escribir "Ingrese la cantidad que va a comprar del articulo"
    Leer cant
    p<-articulo*cant
    Sum<-Sum+p
    Escribir "¿Ya tiene ud todos los articulos necesarios?,¿si o no?"
    Leer end
  Hasta Que end="si"
  Escribir "El total de su compra es ",Sum
FinAlgoritmo
```




Algoritmo 7

```
Repetir
    Escribir "Ingrese el precio de su compra"
    Leer precio
    Escribir "Ingrese el color de la bolita que saco: roja, amarilla o blanca "
    Leer color
    Si color="roja" o color="rojo" Entonces
        p1<-precio*0.4
        desc1<-precio-p1
        Escribir "El total de su compra es ",desc1
    Fin Si
    Si color="amarilla" o color="amarillo" Entonces
        p2<-precio*0.25
        desc2<-precio-p2
        Escribir "El total de su compra es ",desc2
    Fin Si
    Si color="blanca" o color="blanco" Entonces
        Escribir "Su compra no tiene descuento, por lo tanto el total es ",precio
    Fin Si
    Escribir "¿La tienda esta cerrada o abierta?"
    Leer tienda
Hasta Que tienda="cerrada"
```



Referencias Bibliográficas

Martinez,J. (2011)Fundamentos de programación en Java. Madrid España: EME.

Deitel, P., & Deitel, H. (2012). Cómo programar Java (9a. ed. --.). México: Pearson.