

Instrucciones de salida de datos y de control



Estructura general de un programa

□ Un programa Java básico contiene:

■ Comentarios

■ Una clase pública

□ El programa se llama igual que esta clase

□ Esta clase contiene al método main

```
/*  Ejemplo de un programa básico
    El archivo que contiene este programa
    se llama Ejemplo.java                */
public class Ejemplo {
    public static void main (String args[ ]) {
        System.out.println("Un simple programa Java");
    } // fin método main
}
```

Java diferencia entre mayúsculas y minúsculas

Comentarios

□ Una línea `//`

□ Varias líneas `/* */`

□ Documentación: javadoc

`/**`

`* Descripción programa`

`* @author nombre`

`* @param explicación parámetros`

`* @return explicación valor de salida`

`* @version x.x`

`*/`

Package colaEstatica

Class Summary

<u>ColaIntCircular</u>	Crea una estructura de cola circular para elementos enteros
--	---

Variables: declaración y asignación

□ Declaración

```
[alcance] tipo identificador [=valor]  
[, identificador [=valor]];
```

```
boolean semaforo=true, continuar;
```

```
public class AreaRectangulo {  
    public static void main(String[] args) {  
        float base=5, altura=3f, area;  
        area = base * altura;  
        System.out.println("El area es. "+area);  
    }  
}
```

Buena práctica. El identificador de una variable de tipo primitivo inicia con minúsculas.

Puede contener: caracteres alfanuméricos, \$, _

Inicializando tipos de datos constantes

```
12 public class AreaRectangulo {
13     public static void main(String[] args) {
14         float base=5, altura=3f, area;
15         area = base * altura;
16         System.out.println("El area es. "+area);
17     }
18 }
```

```
12 public class AreaRectangulo {
13     public static void main(String[] args) {
14         float base=5.0, altura=3f, area;
15         area = base * altura;
16         System.out.println("El area es. "+area);
17     }
18 }
```

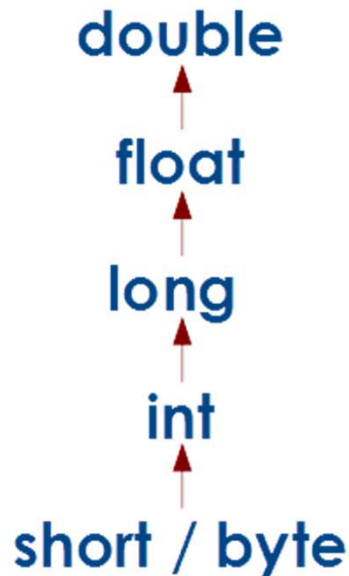
incompatible types: possible lossy conversion from double to float

(Alt-Enter shows hints)

```
12 public class AreaRectangulo {
13     public static void main(String[] args) {
14         double base=5.0, altura=3f, area;
15         area = base * altura;
16         System.out.println("El area es. "+area);
17     }
18 }
```

Conversión de datos implícita

- Promoción aritmética para operadores binarios.
 - Si uno de los operandos tiene mayor precisión, el otro se “promueve” a ese tipo.



```
public class ConversionTipo {  
    public static void main(String[] args) {  
        byte x, a=120, b=10;  
        x = a + b;  
        System.out.print("a+b= "+x);  
    }  
}
```

possible loss of precision
required: byte
found: int

Conversión de datos explícita

□ Operador conversión de tipo (cast)

(tipo de dato) <expresión>

□ Ejemplos:

- (double) 5/3

- (double) (5/3)

```
public class ConversionTipo {  
    public static void main(String[] args) {  
        short x;  
        byte a=120, b=10;  
        x = (byte) (a + b);  
        System.out.print("a+b= "+x);  
        x = (short) (a + b);  
        System.out.println("      a+b= "+x);  
    }  
}
```

```
run-single:  
a+b= -126    a+b= 130  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Salida de datos a pantalla

□ `System.out.print[ln] (expresión) ;`

- Expresión como cadena de caracteres.
- Con el `+` se concatenan cadenas.

□ Ejemplo:

```
public class Bits {  
    public static void main(String[] args) {  
        short numero1 = 0x3A6B, numero2=0x00F0;  
        System.out.println(numero1&numero2);  
        System.out.println("9|14 = "+(9|14));  
        System.out.println("20<<2 = "+(20<<2));  
        System.out.println("20>>2 = "+(20>>2));  
    }  
}
```

run:

96

9|14 = 15

20<<2 = 80

20>>2 = 5

BUILD SUCCESSFUL

Salida con formato

- Se tiene una cadena para proporcionar formato

- `System.out.printf();`

- `System.out.printf("cadena_control" [, expresión] ...)`

- Cadena de control

- Caracteres ordinarios

- Secuencias de escape

- Especificaciones de formato

- `%[bandera][ancho][.precisión]tipo`

```
char c = 'I';
```

```
System.out.printf("El ascii del caracter %c es %d\n",c,c);
```

Formato salida

□ Tipo

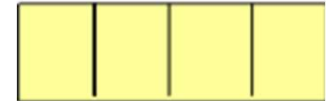
- d entero base 10
- o entero base 8
- X, x entero base 16
- C, c un caracter
- s cadena de caracteres
- f real con decimales
- E, e real notación científica
- G, g según convenga usa formato e o f
- % signo de %
- \n salto de línea

Formato salida (cont)

□ Banderas

- – justifica a la izquierda
- + siempre con signo
- 0 rellena con cero
- (valores negativos entre paréntesis
- , separador de cifras

%4d



%-4d



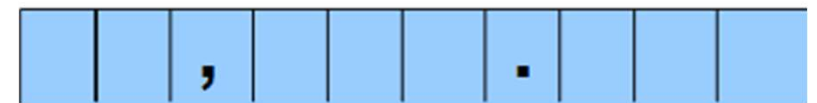
□ Ancho

- Número total de posiciones

%,10.3f

□ Precisión

- Número de decimales
- Por defecto son seis en %f y cinco en %g



Ejemplo salida con formato

```
public class SalidaFormato {  
    public static void main(String[] args) {  
        int a=123;    float b=45.678F;    double d=1234567.123;  
        System.out.printf(" Enteros: %d %d %d %d %d \n",  
                           a, a, -a, -a, a);  
        System.out.printf(" Bases: %o %X %d %d\n",  
                           a, a, 0123, 0x123);  
        System.out.printf(" Flotantes: %f %e %g %g \n",  
                           b, b, b, d);  
        System.out.printf(" Decimales: %9.2f %9.2e %11.4e\n",  
                           b, b, b);  
        System.out.printf(" Comas: %,d %,f %(.2f\n",  
                           12345, d, -d);  
    }  
}
```

Enteros: 123 +123 -123 (123) 00123

Bases: 173 7B 83 291

Flotantes: 45.678001 4.567800e+01 45.6780 1.23457e+06

Decimales: 45.68 4.57e+01 4.5678e+01

Comas: 12,345 1,234,567.123000 (1,234,567.12)

La clase Math

□ Contiene métodos con funciones matemáticas

■ Uso: `Math.metodo`

□ Contiene constantes matemáticas

■ Uso: `Math.constante`

Fórmula en código Java con el mínimo de paréntesis

$$\frac{1}{2} \text{sen} \left(x - \frac{\pi}{\sqrt{y}} \right)$$

```
1.0 / 2.0 * Math.sin(x - Math.PI / Math.sqrt(y))
```

método	argumento	resultado
abs(a)	int long	int long
	float double	float double
cos(a)	double	double
sin(a)	double	double
tan(a)	double	double

método	argumento	resultado
sqrt(a)	double	double
pow(a,b)	double	double
round(a)	float	int
	double	long
random()		double

Ejemplo uso de métodos Math

```
/**
 * Genera una cordenada cartesiana aleatoria en el
 * cuadrante de coordenadas enteras (1,1) a (7,7)
 * Proporciona el equivalente en coordenadas polares
 * @author maria.villar
 */
public class EjemplosMath {
    public static void main(String[] args) {
        int x, y;
        double r, angulo;
        //Genera un número entero entre 1 y 7
        x=1+(int) (Math.random()*7);
        y=1+(int) (Math.random()*7);
        //Obtiene valor de r y ángulo
        r=Math.sqrt(x*x+Math.pow(y, 2));
        angulo=Math.atan( (float)y/x);
        //Convierte el ángulo de radianes a grados
        angulo= angulo * 180.0 / Math.PI;
        //Imprime resultado
        System.out.println("Cartesianas: (" +x+" "+y+" )");
        System.out.printf("Polares: (%.2f,%.1f)\n",r,angulo);
    }
}
```

run:
Cartesianas: (4,4)
Polares: (5.66,45.0)
BUILD SUCCESSFUL (total time: 0s, 1ms)

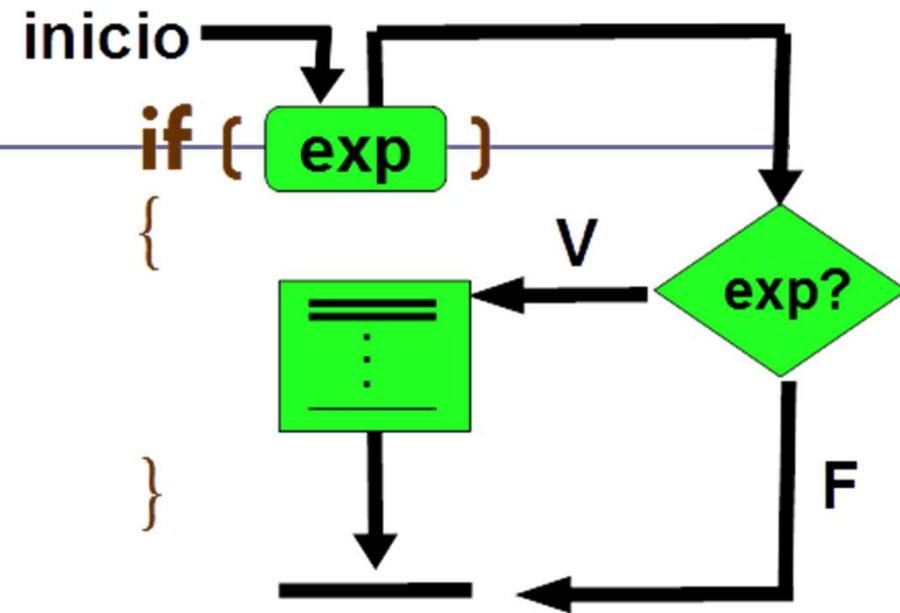
run:
Cartesianas: (5,1)
Polares: (5.10,11.3)
BUILD SUCCESSFUL (total time: 0s, 1ms)

Sentencias de control

- Controlan el flujo de ejecución de un programa
- Secuencias
 - Sentencia sencilla, termina en ;
 - Sentencia compuesta (bloque), conjunto de sentencias encerradas entre llaves
- Sentencias de selección
 - if , if-else, if-elseif-else
 - switch
- Sentencias de repetición
 - for
 - while
 - do-while

Sentencia if

- Ejecuta o no una acción con base en el valor de una expresión booleana
- Ejemplos:



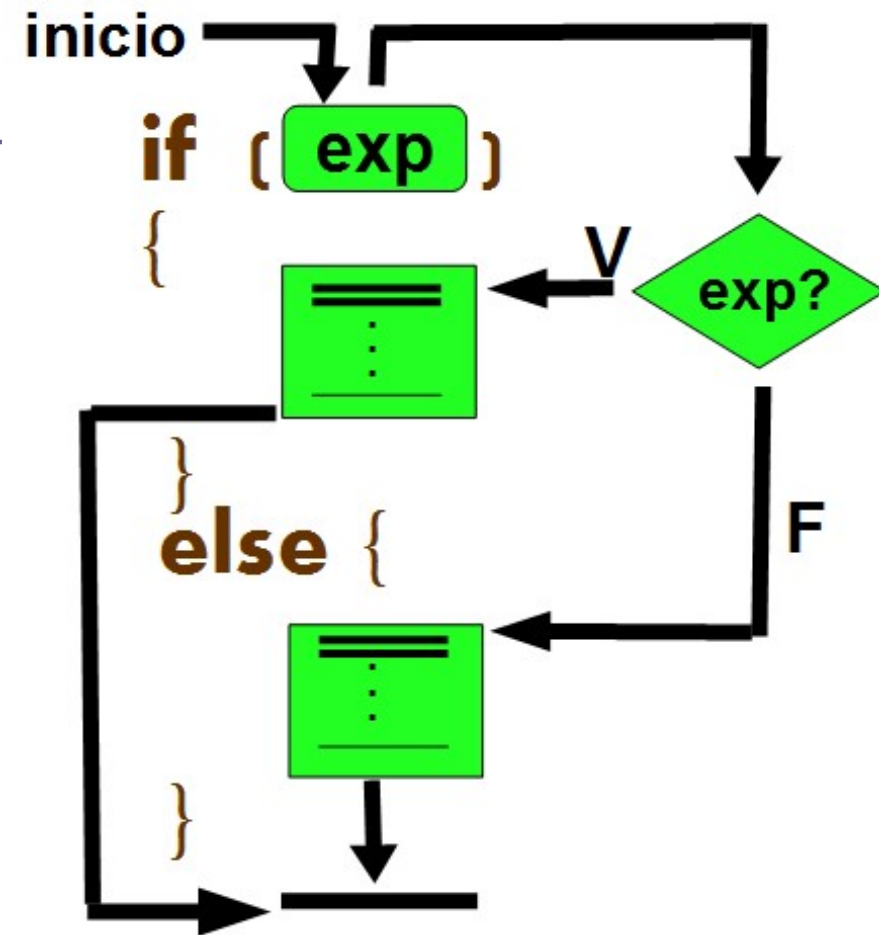
```
if ( x > 0 ) positivo=true;
```

```
if ( x>=0 && x<=100 ) {  
    calif= (float)x/10.0;  
    System.out.printf("%4.1f",calif) ;  
}
```


Sentencia if-else

- Ejecuta una acción u otra con base en el valor de una expresión booleana
- Ejemplo:

```
if ( x > 0 ) {  
    positivo=true;  
}else{  
    positivo=false;  
}
```



Sentencia if-else (cont)

```
if ( x>=0 && x<=100) {  
    calif= (float)x/10.0f;  
    System.out.printf("%4.1f",calif) ;  
}  
else{  
    System.out.printf("Error") ;  
}
```

Ejemplo if - else if - else

Si una compra rebasó \$1000, aplicar 20% descuento, si estuvo entre 600 y 1000 un 10%, si fue menor de 600 pero mayor o igual a 300 un 5%, menor a 300 sin descuento.

```
if (compra > 1000){  
    precio= compra*.8;  
}else if (compra >= 600 ){  
    precio= compra*.9;  
}else if (compra >= 300 ){  
    precio= compra*.95;  
}else {  
    precio= compra;  
}
```

Ejemplo if - else if – else (cont)

b) Contar el número de positivos, negativos y ceros.

```
if ( x > 0 ) {  
    positivos++;  
} else if ( x < 0 ) {  
    negativos++;  
} else {  
    ceros++;  
}
```

Recomendaciones

- ❑ Verificar que se usa el signo == y no =
- ❑ No usar el == o != con tipos de datos flotantes

```
if ( x == 35.7) {  
    System.out.print("ganaste");  
} else {  
    System.out.print("perdiste");  
}
```

```
if ( x = 0) {  
    System.out.print("Cero");  
}
```



Sentencia switch

- Seleccionar una entre múltiples alternativas
 - Depende del valor de una expresión de control
 - Toma un valor único de tipo int o char
 - En java 8, puede ser String
- Cada una de las opciones se coloca como una etiqueta
- Se ejecuta en secuencia a menos que se encuentre una sentencia break

```
switch (opcion) {  
    case 'a': a=0;  
                break;  
    case 'b': b=0;  
                break;  
    case 'c': c=0;  
                break;  
    case 'd': d=0;  
                break;  
}
```


Ejemplos switch

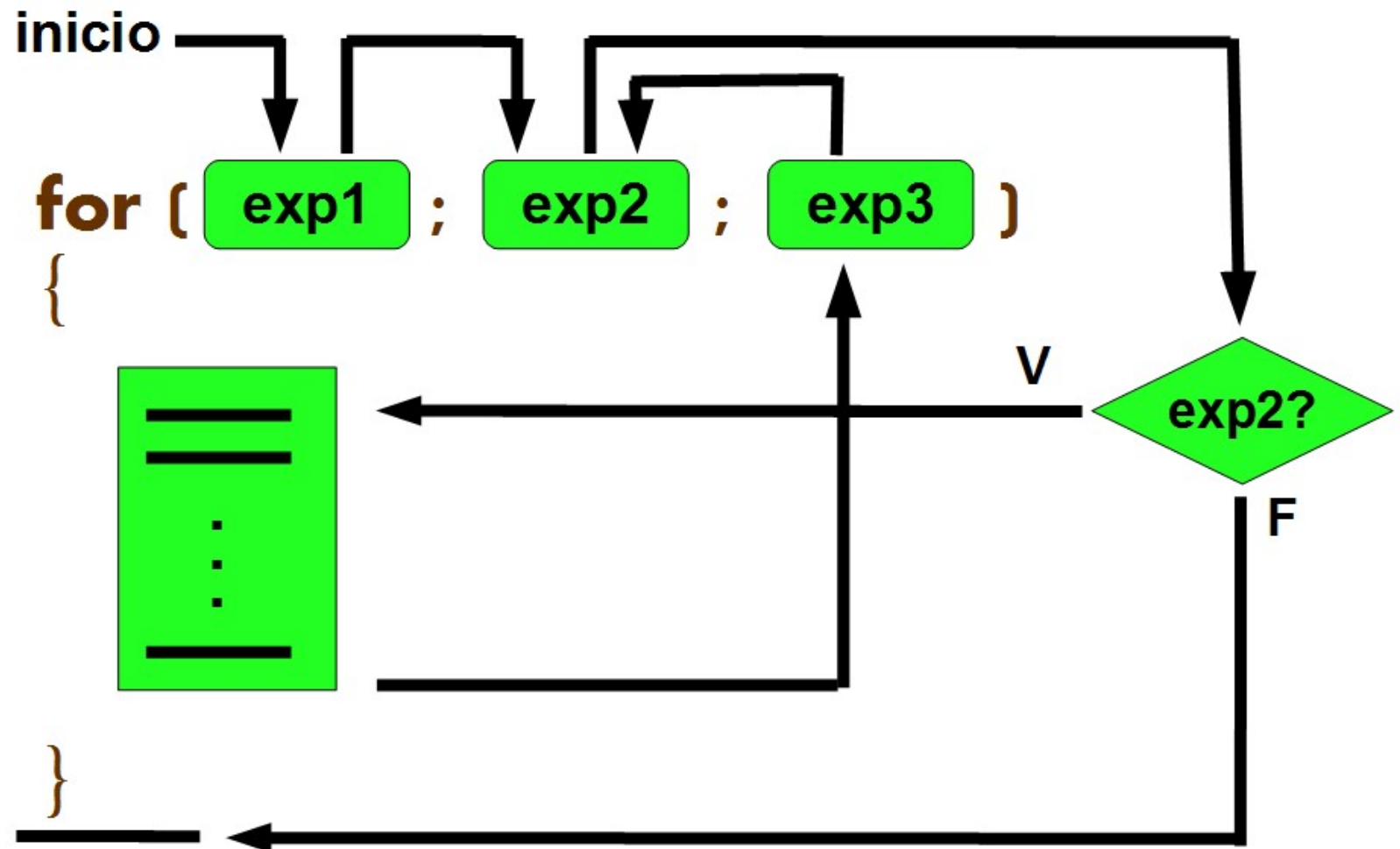
```
-switch (opcion) {  
    case 'a': case 'A': a=0;  
        break;  
    case 'b': case 'B': b=0;  
        break;  
    case 'c': case 'C': c=0;  
        break;  
    case 'd': case 'D': d=0;  
}
```

```
switch (opcion) {  
    case 'a': case 'A':  
    case 'e': case 'E':  
    case 'i': case 'I':  
    case 'o': case 'O':  
    case 'u': case 'U':  
        System.out.print("Vocal\n");  
        break;  
    default:  
        System.out.print("No es vocal\n");  
}
```

```
switch (opcion) {  
    case 0:  
    case 1:  
    case 2:  
        valor=" < 3";  
        break;  
    case 3:  
        valor=" = 3";  
        break;  
    default:  
        valor=" > 3";  
}
```

Diagrama sentencia for

- Ejecuta un bloque de sentencias un número fijo de veces



Ejemplos sentencia for

```
for(int i=1; i<=10; i++)  
    System.out.print(i);
```

```
for(int i=10; i>=0; i--)  
    System.out.print(i);
```

```
int i;  
for(i=0; i<10; i++);  
    System.out.print(i);
```



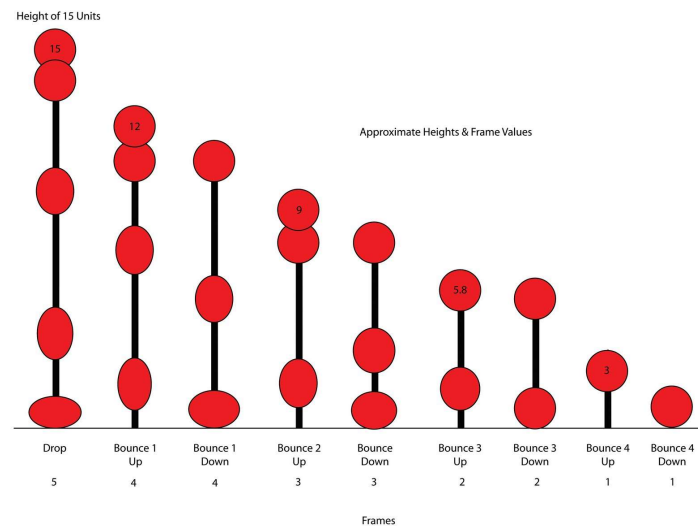
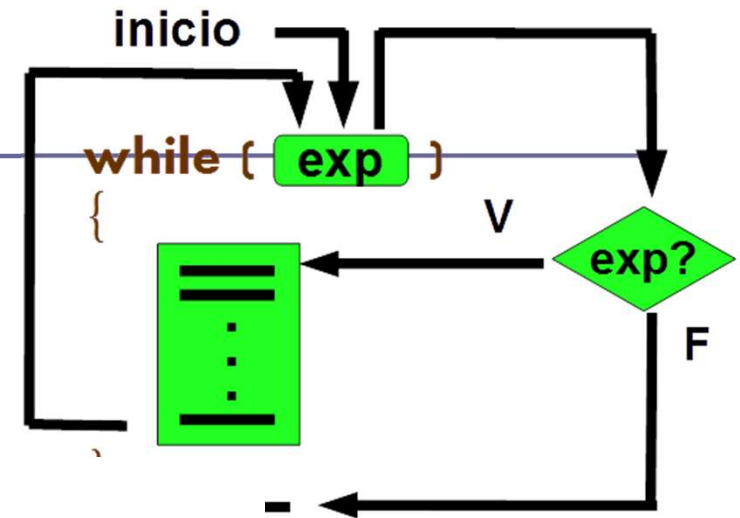
```
double x;  
for(x=-3.2; x<3.2; x+=0.25) {  
    System.out.printf("%f ", x);  
}
```

```
char c;  
for(c='A'; c<='Z'; c++) {  
    System.out.print(c);  
}
```

```
int a,b;  
for(a=256, b=1; b<=512; a/=2, b*=2) {  
    System.out.print("\t"+b+"\t"+a);  
}
```

Diagrama sentencia while

- Ejecuta un bloque de sentencias si se cumple una condición.
- Ejemplo:

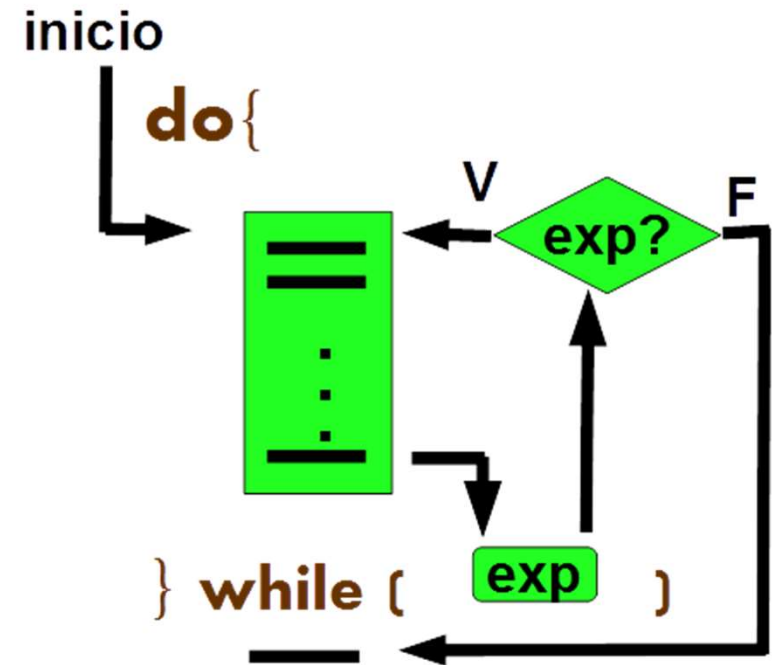


```
double altura=15;
double amortigua=0.8;
while(altura > 0.01){
    System.out.println("Altura = "+altura);
    altura *= amortigua;
}
```

Diagrama sentencia do-while

- Ejecuta un bloque de sentencias si se cumple una condición.
- Ejemplo:

```
double altura=16;  
double amortigua=0.8;  
do{  
    System.out.println("Altura = "+altura);  
    altura *= amortigua;  
}while(altura > 0.01);
```



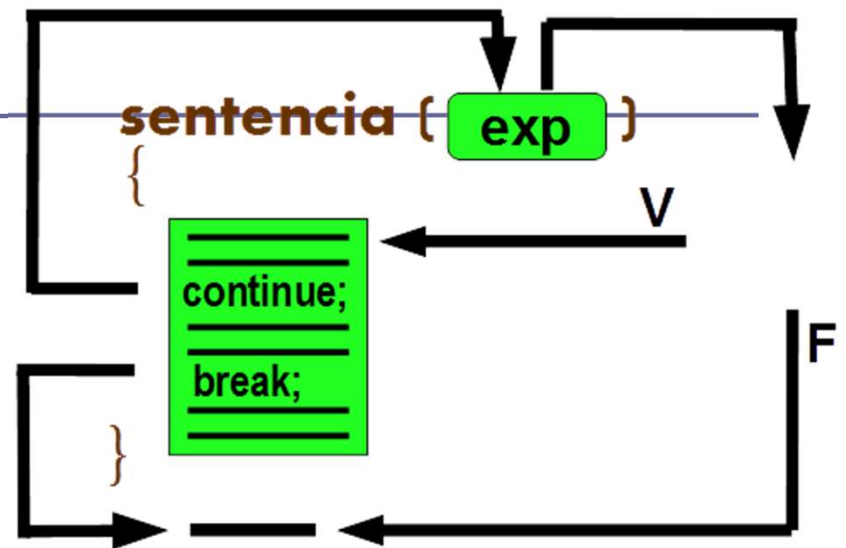
Sentencias break y continue

□ break

- Sale del bloque.

□ continue

- En sentencias de repetición, lleva a evaluar la expresión, sin ejecutar las sentencias posteriores.



```
for(int i=0; i<=100; i++) {  
    if ( i%3==0 && i%7==0)  
        continue;  
    System.out.print(i) ;  
}
```