

Programación II

Carla TARAMASCO

Profesora e Investigadora

DECOM & CNRS

mail : alumnos_uv@yahoo.cl

14 mars 2013

Plan Arreglos, condiciones y ciclos

■ Arreglos

- Declarar variables de arreglo
- Crear objetos de arreglo
- Acceder a los elementos de un arreglo
- Cambiar elementos de un arreglo

■ Arreglos multidimensionales

■ Condicionales

- if
- switch

■ Ciclos

- for
- while y do

■ Ejercicio de Arreglos, condicionales y ciclos

Declarar variables de arreglo

Las variables de arreglo indican el tipo de objeto que el arreglo contendrá y el nombre del arreglo. Los corchetes vacíos pueden ponerse después del tipo de dato o después del nombre del arreglo indistintamente.

- String palabras[] ;
- Point hist[] ;
- int temps[] ;
- float [] promedio ;
- String [] palabras ;
- Point [] hist ;

Crear objetos de arreglo

- 5/40

Acceso a los elementos del arreglo

```
String [] arr = new String [10];
```

```
int len = arr.length;
```

En java no puede asignar un valor a una casilla del arreglo fuera de las fronteras de este. Los subindices se inician en 0.

Cambiar elementos del arreglo

```
arr [1]=10;  
promedio [9] = 9.7;  
texto [4] = "test";  
cadenas [10] = cadenas[1];
```

Al igual que con los objetos, un arreglo de objetos en java consiste en un arreglo de referencias a dichos objetos. Cuando asigna un valor a una casilla en un arreglo, crea una referencia a ese objeto. Cuando desplaza valores dentro de un arreglo solo se reasigna la referencia, **no** se copia el valor de una casilla a otra. En cambio los arreglos de tipos primitivos **si** copian los valores de una casilla a otra.

Arreglos Multidimensionales

Java no soporta los arreglos multidimensionales. Sin embargo, se pueden declarar un arreglo de arreglos y acceder a el de la siguiente manera :

```
int coods [][]= new int [12][12];
coods [0][0]=2;
```


if

El condicional **if** permite ejecutar partes del código basándose en una simple prueba. Contiene la palabra clave **if** seguida de una prueba booleana y de un enunciado a ejecutar si la prueba es verdadera. Una palabra opcional **else** ofrece el enunciado a ejecutarse si la prueba es falsa.

```
if (estado == true)
    System.out.println("El estado es verdadero");
else {
    System.out.println("cambiando el estado...");
    if (aceptaCambio == 1)
        estado = true;
    System.out.println("Cambio el estado y ahora es verdadero");
    else System.out.println("No se acepto el cambio de estado");
}
```

Operador condicional

?

El operador condicional **?** es un operador ternario solo tiene tres términos.

Es útil para condicionales cortos o sencillos.

test ? trueResult : falseResult ;

int smaller = x < y ? x : y ;

Condicional switch

Para evitar **if** muy largos se usa el **switch** o **case** que permite agrupar pruebas y acciones en un solo enunciado.

```
switch (test){  
    case valorUno:  
        resultadoUno;  
        break;  
    case valorDos:  
        resultadoDos;  
        break;  
    case valorTres:  
        resultadoTres;  
        break;  
    default : resultadoPorDefecto;  
}
```

Si no existen coincidencias en ninguno de los casos y el *default* no existe el *switch* se completa sin hacer nada.

Condicional switch

Limitaciones y break

Una limitación del condicional **switch** es que las pruebas y los valores solo pueden ser de tipos primitivos específicamente **int**. No puede usar tipos primitivos mas grandes (**long, float**), cadenas u objetos, tampoco puede probar otra relación que la igualdad. Si se encuentra una coincidencia se ejecuta el enunciado de esta y todos los inferiores hasta encontrar un **break** o hasta el final del **switch**. Si solo se quiere ejecutar el enunciado de la prueba se debe poner un **break** después de cada linea.

```
switch (x){  
    case 2:  
    case 3:  
    case 6:  
    case 8:  
        System.out.println ("x esta en uno de los casos");  
        break;  
    default: System.out.println ("x no esta en los casos");  
}
```

for

```
for (inicialización ; test ; incremento){
enunciados }
```

- 13/40

Ejercicio

- Escriba un programa que construya un arreglo de tamaño 10 y lo llene con números enteros entre 0 y 100 generados aleatoriamente.
- Recorra el arreglo y sume su contenido. Imprima cada valor y la suma total en pantalla. Para generar números aleatorios use el metodo random de la clase Math que esta en el paquete `java.lang.Math`

Ejercicio resuelto

```
//import java.lang.Math;
public class EjArregloSum{

    public static int sumar (int[] tab)
    {
        int sum = 0,r;
        for (int i = 0; i < tab.length; i++)
        {
            r= (int) (Math.random()*100);
            tab[i]=r;
            System.out.println ("el valor numero:" +i+"es"+r);
            sum += tab[i];
        }
        return sum;
    }

    public static void main (String args[]){
        int tab[]=new int [10];
        System.out.println ("la suma es : " + sumar(tab));
    }
}
```

Ejemplo discutido en clases

```
public int ejCaracterAleatoreo () {  
    int rand_num = (int) (Math.random ()*(122 - 97))+97;  
    System.out.println (" Letra : " + (char)rand_num );  
    return rand_num ;  
}
```


Ejercicios

Mostrar las notas de un alumno

- Cree una clase "estudiante", declare un arreglo de notas de tipo int y una constante con la cantidad de notas totales del estudiante (*static final int NB_NOTAS = 10 ;*)
- Cree un método "mostrar" que imprima en pantalla las notas de un alumno.
- Cree un método "llenar" que llene el arreglo con notas aleatorias entre 0 y 10.
- Cree un método "promedio" que calcule el promedio de notas del alumno.
- En el método principal : Cree un objeto del tipo "estudiante", cree el arreglo declarado como atributo de la clase y (usando el objeto recién creado) llame a los metodos llenar, mostrar y promedio.

Ejercicio resuelto

```
public class Estudiante {  
    // Numero de notas  
    static final int NB_NOTAS = 10;  
    // Arreglo de notas  
    int notas[] ;  
  
    void mostrar(){  
        // Mostrar las notas del estudiante  
        for (int i = 0 ; i < notas.length ; i++)  
            System.out.println (notas [i]) ;  
    }  
    void llenar(){  
        // Llenar el arreglo  
        for (int i = 0 ; i < NB_NOTAS ; i++) {  
            notas[i]=(int)(Math.random()*10);  
            //System.out.println (notas [i]) ;  
        } }  
}
```

Ejercicio resuelto continuación

```
void promedio(){
// Promedio de notas
float promedio=0;
for (int i = 0 ; i < NB.NOTAS ; i++) {
promedio+=notas[i];}
promedio/=NB.NOTAS;
System.out.println ("El promedio es :"+promedio) ;
}
public static void main (String arg []) {
// Crear un objeto del tipo Estudiante
Estudiante duran = new Estudiante() ;

// Crear el arreglo de notas
duran.notas = new int [NB.NOTAS] ;
duran.llenar();
duran.mostrar();
duran.promedio();
}}
```

Lectura argumentos

Usando arreglos

Como podemos leer una lista de argumentos ingresados al momento de ejecutar el código ?

```
public class LecturaArgumentos {  
    public static void main( String [] arg) {  
        for (int i=0 ; i < arg.length ; i++) {  
        }  
    }  
}
```

Lectura argumentos

Usando arreglos

- Desarrolle un programa que lea un conjunto de notas ingresadas como argumentos al ejecutar el programa. Use el metodo `parseInt` de la clase `Integer` para pasar de caracter a enteros y `parseFloat` de la clase `Float` para flotantes.
- Cree un método que ingrese las notas a un arreglo.
- Cree un metodo que calcule el promedio de notas y lo imprima en pantalla.

Ejercicio resuelto

```

public class AdjuntarNotas {
    int notas[] ;

    void promedio(String [] arg){
        // Promedio de notas
        float promedio=0;
        for (int i = 0 ; i < arg.length ; i++) {
            promedio+=notas[i];}
        promedio/=arg.length;
        System.out.println ("El promedio es :"+promedio) ;
    }

    void llenar(String [] arg){
        // Llenar el arreglo
        for (int i = 0 ; i < arg.length ; i++) {
            notas[i]= Integer.parseInt(arg[i]);
        }
    }

    public static void main( String [] arg) {

        AdjuntarNotas duran = new AdjuntarNotas() ;

        // Crear el arreglo de notas
        duran.notas = new int [arg.length] ;
        duran.llenar(arg);
        duran.promedio(arg);} }

```

Lectura de datos

Usando la clase Scanner

Como podemos leer datos desde el teclado ?

```
import java.util.Scanner;
public class LecturaScanner {
    static final int TOT=4;
    public static void main( String [] arg) {
        int nota[]=new int [TOT];
        //creamos un objeto kyb que leera desde teclado
        Scanner kyb = new Scanner(System.in);
        System.out.println ("Deberas ingresar un total de "+ TOT + " valores");
        for (int i=0 ; i < TOT ; i++) {
            System.out.println ("Ingresa el valor n :"+i);
            //el metodo nextInt permite leer el entero ingresado por el usuario
            nota[i] =kyb.nextInt();
            //System.out.println (nota[i]);
        }
    }
}
```

Lectura de datos

Usando la clase Scanner

Para leer datos float, long, short, double, String, debemos usar : .

- nextFloat()
- nextShort()
- nextDouble()
- next() y nextLine()

Ejercicios

Trabajar con arreglos y metodos

- Cree una clase "EjArreglos"
- Construya 4 arreglos de tamaño $N=10$ y uno de tamaño definido por el usuario.
- Elija si va a trabajar con metodos static o creara objetos.
- Metodos para llenar cada arreglo :
 - ➊ Arr1 : con elementos enteros en orden creciente entre $[0, n[$.
 - ➋ Arr2 : con elementos enteros en orden creciente a partir de 5.
 - ➌ Arr3 : con elementos enteros en orden decreciente de $[n, 1]$.
 - ➍ Arr4 : con elementos enteros generados aleatoriamente entre $[0, 10]$.
 - ➎ Arr5 : con elementos enteros ingresados por el usuario como argumentos o por teclado.

Ejercicios continuación

Trabajar con arreglos y metodos

■ Métodos a aplicar a cada arreglo :

- ➊ crear método "mostrar" que imprima en pantalla los elementos del arreglo en pantalla.
- ➋ crear método "mostrarMayorMenor" que el numero de elementos mayor que 4 o menor que 2.
- ➌ crear método "mostrarNueve" que calcule e imprima en pantalla el numero de elementos de valor 9 del arreglo.
- ➍ crear método "mostrarSuma" que calcule e imprima la suma de todos los elementos del arreglo.
- ➎ crear método "mostrarMayor" que imprima el mayor valor del arreglo.
- ➏ crear método "mostrarPromedio" que calcule e que imprima el promedio de los elementos del arreglo.
- ➐ Genere un método llamado "operaciones" que llame a los 6 métodos auxiliares creados. entregando el arreglo como parametro.

Ejercicio resuelto

```
import java.util.Scanner;
class EjArreglos {

    static final int N_TAMANO =10;

    /*Este metodo retorna un arreglo int [] de n elementos
       inicializados en orden creciente de 0 a N_TAMANO-1 */

    static int[] remplir1(int nb) {
        int[] tab = new int[nb];
        for (int i = 0; i < nb; i++) {
            tab[i] = i;}
        return tab;    }

    /*Este metodo retorna un arreglo int [] de N_TAMANO elementos
       inicializados en orden creciente a partir de 5 */
    static int[] remplir2(int nb) {
        int[] tab = new int[nb];
        for (int i = 0; i < nb; i++) {
            tab[i] = i + 5;}
        return tab;}
}
```

Ejercicio resuelto continuación

```
/*Este metodo retorna un arreglo int [] de N.TAMANO elementos
  inicializados en orden decreciente de N.TAMANO a 1 */
static int[] remplir3(int nb) {
int[] tab = new int[nb];
for (int i = 0, j = nb; i < nb; i++, j--) {
tab[i] = j;}
return tab;    }
```

```
/*Este metodo retorna un arreglo int [] de N.TAMANO elementos inicializados aleatoreamente*/
static int[] remplir4(int nb) {
int r;
int [] tab = new int[nb];
for (int i = 0; i < nb; i++){
r= (int) (Math.random()*10);
tab[i]=r;}
return tab;}
```

```
/*Este metodo retorna un arreglo int [] de arg.lentgh elementos ingresados por teclado*/
static int[] remplir5() {
int r;
Scanner kyb=new Scanner(System.in);
System.out.println("Cuantos valores desea ingresar? ");
int cantValores = kyb.nextInt();
int [] tab = new int[cantValores];
for (int i = 0; i < cantValores; i++){
System.out.print("Ingrese el valor entero numero "+i+" : ");
tab[i]= kyb.nextInt();}
return tab;}
```

Ejercicio resuelto continuación

```
static void operations(int[] tab) {  
    afficher(tab);  
    operation1(tab);  
    operation2(tab);  
    operation3(tab);  
    operation4(tab);  
    operation5(tab);}
```

/*Este metodo imprime todos los elementos del arreglo en la misma linea los elementos estan separados por 2 espacios. */

```
static void afficher(int[] tab) {  
    for (int i = 0; i < tab.length; i++) {  
        System.out.print(tab[i]);  
        System.out.print(" ");  
        System.out.println();  
    }
```

/*Este metodo calcula e imprime el numero de elementos de valor 9. La variable nb se usa para contar los elementos encontrados, se inicializa en 0 y se incrementa en el ciclo cada vez que encuentra un 9*/

```
static void operation1(int[] tab) {  
    int nb = 0;  
    for (int i = 0; i < tab.length; i++) {  
        if (tab[i] == 9)  
            nb++;  
    }  
    System.out.println("Hay " + nb + " elementos de valor 9 en el arreglo");}
```

Ejercicio resuelto continuación

```
/*Este metodo calcula e imprime el numero de elementos que son
mayores que 4 o menores que 2 */
static void operation2(int[] tab) {
    int nb = 0;
    for (int i = 0; i < tab.length; i++) {
        if ((tab[i] > 4) || (tab[i] < 2))
            nb++;
    }
    System.out.println("Hay " + nb + " elementos mayores que 4 o menores que 2");
}

/*Este metodo calcula e imprime la suma de todos los elementos del arreglo*/
static void operation3(int[] tab) {
    int somme = 0;
    for (int i = 0; i < tab.length; i++) {
        somme = somme + tab[i];
    }
    System.out.println("La suma de los elementos es " + somme);
}

/*Este metodo calcula e imprime el promedio de todos los elementos del arreglo*/
static void operation4(int[] tab) {
    double somme = 0.0;
    for (int i = 0; i < tab.length; i++) {
        somme = somme + tab[i];
    }
    double moyenne = somme / tab.length;
    System.out.println("El promedio de los elementos es " + moyenne);
}
```

Ejercicio resuelto continuación

```
/*Este metodo calcula e imprime el mayor elemento del arreglo*/
static void operation5(int[] tab) {
    int max = tab[0];
    for (int i = 1; i < tab.length; i++) {
        if (tab[i] > max) {
            max = tab[i];
        }
    }
    System.out.println("El mayor elemento es : " + max);
}

public static void main(String args[]) {
    // Construccion de los arreglos
    int[] tab1 = remplir1(N.TAMANO);
    operations(tab1);

    int[] tab2 = remplir2(N.TAMANO);
    operations(tab2);

    int[] tab3 = remplir3(N.TAMANO);
    operations(tab3);

    int[] tab4 = remplir4(N.TAMANO);
    operations(tab4);

    int[] tab5 = remplir5();
    operations(tab5);
}}
```

Ciclos

While y do

El ciclo **while y do** permiten ejecutar un bloque de código de manera repetida hasta encontrar una condición específica.

- Se ejecuta el bloque de código hasta que la condición sea verdadera. Si la condición es falsa el ciclo nunca se ejecutará.

```
while (condicion){  
  instrucciones a ejecutar;  
}
```

- Este ciclo prueba la condición después de haberse ejecutado una vez.

```
do{  
  instrucciones a ejecutar;  
}while (condicion);
```


Ciclos

Como salir de los ciclos ?

Todos los ciclos se terminan cuando la condición se cumple, si desea salir de manera anticipada y detener por completo el ciclo actual debe usar **break**.

```
public class Break1 {
    public static void main (String args []){
        for (int i=1; i<=10;i++){
            System.out.println("Inicio del ciclo" + i);
            if (i==3) break;
            System.out.println("fin del ciclo"+ i);
        }System.out.println("Despues del ciclo");
    }
}
```

Inserte este código dentro de una clase y pruebe el resultado.

```
int cont = 0;
while (cont < arr.length){
    if (arr[cont]==0){
        break;}
    arr1[cont]=(float)arr[cont++];}
```

Ciclos

Como salir de los ciclos ?

Todos los ciclos se terminan cuando la condición se cumple, si desea salir de manera anticipada e iniciar la siguiente iteración (reiniciar el ciclo) debe usar **continue**. Para los ciclos **while** y **do** significa que la ejecución del bloque se inicia de nuevo. Para el ciclo **for** el incremento se evalúa y después el bloque se ejecuta.

```
public class Conti {  
    public static void main (String args []){  
        for (int i=1; i<=10;i++){  
            System.out.println("Inicio del ciclo" + i);  
            if (i<4) continue;  
            System.out.println("fin del ciclo"+ i);  
        }System.out.println("Despues del ciclo");  
    }  
}
```

Inserte este código dentro de una clase y pruebe el resultado.

```
int cont = 0;  
int cont1 = 0;  
while (cont < arr.length){  
    if (arr[cont]==0){  
        continue;}  
    arr1 [cont1 ++]=(float) arr [cont ++];}
```

Ciclos

Como salir de los ciclos ?

Se pueden usar etiquetas para salir de ciclos anidados o salir de más de un ciclo al mismo tiempo.

```
out:
for (int i=1;i<=5;i++){
    for (int j=1;j<=3;j++){
        if ((i+j)>4)
            break out;}}
System.out.println("salida de ciclos")
```

Ejercicios lectura, condicionales y ciclos

- Escriba un programa que calcule la raíz cuadrada de valores ingresados por el usuario. Pregunte al usuario cuantos valores ingresará. Solo debe considerar valores positivos.
- Escriba un programa de facturación con descuento. El usuario debe ingresar el precio sin impuestos, y el programa calculará los impuestos (fijo a 19%) y hará el descuento si corresponde.
 - 0% si el monto es inferior a 1000
 - 1% si el monto es superior o igual a 1000 e inferior a 3000
 - 3% si el monto es superior o igual a 3000 e inferior a 5000
 - 5% si el monto es superior o igual a 5000
- Escriba un programa que calcule el área de un rectángulo. El usuario debe ingresar los valores de x_1, x_2, y_1, y_2 .
- Escriba un programa que identifique si un caracter es vocal o consonante. Cree 20 caracteres a partir de números aleatorios. Imprima en pantalla el caracter y su tipo.

Ejercicios lectura, condicionales y ciclos

```

import java.util.Scanner;
public class Raices {

    static void raizCal (float a){
        // calcular la raiz cuadrada
        float raiz = (float)Math.sqrt(a);
        System.out.println ("La raiz de " +a + " es :"+raiz); }

    static void leerDato(int nveces){
        // Datos del usuario
        int i=1;
        Scanner kyb = new Scanner(System.in);
        do {
            System.out.println ("Ingrese el valor para calcular");
            float dato =kyb.nextFloat();
            if (dato<0) {
                System.out.println ("El valor ingresado debe ser mayor que 0");
                break;}
            raizCal(dato);
            i++;
            System.out.println ("Usted a ingresado"+i+"de"+nveces+"valores a ingresar")

        } while (i < nveces);}

    public static void main( String [] arg) {
        System.out.println ("Cuantos valores quieres calcular?");
        Scanner kyb = new Scanner(System.in);
        int nveces =kyb.nextInt();
        leerDato(nveces);} }

```

Ejercicios lectura, condicionales y ciclos

```
import java.util.Scanner;
public class Factura {
    final static double IMPUESTO=18.7;

    static void descuentoCal (double preciomp){
        // calcular el valor - descuento
        double des,precioFinal;
        if (preciomp < 1000.) des=0;
        else if (preciomp < 3000.) des=1.;
        else if (preciomp < 5000.) des=3.;
        else des=5.;

        precioFinal= preciomp - (preciomp * (des/100.));
        System.out.println ("El precio final es : " +precioFinal );
    }

    static void impuestoCal(){
        // Calcular precio + impuesto
        Scanner kyb = new Scanner(System.in);
        double precio =kyb.nextFloat();
        double preciomp = precio + (precio * (IMPUESTO/100.));
        System.out.println ("El precio es sin impuesto es : " +precio );
        System.out.println ("El precio con impuesto es: " +preciomp );
        descuentoCal(preciomp);
    }

    public static void main( String [] arg) {
        System.out.println ("Ingrese el precio");
        impuestoCal();}
}
```

Ejercicios lectura, condicionales y ciclos

```
import java.util.Scanner;
public class Rectangulo {
    float x1, x2, y1, y2;

    void rectangulo ()

{
    Scanner sc=new Scanner(System.in);
    System.out.println("Ingresa datos para x e y:");
    x1=sc.nextInt();
    x2=sc.nextInt();
    y1=sc.nextInt();
    y2=sc.nextInt();
}

    float area(){
        return (x2-x1)*(y2-y1);
    }
    public static void main(String[] args)
    {
        Rectangulo p1=new Rectangulo();
        p1.rectangulo();
        System.out.println(p1.area());
    }
}
```

Ejercicios lectura, condicionales y ciclos

```
public class Consonante{
    public static void main(String[] args){
        for(int i = 0; i < 20; i++){
            char c = (char)((Math.random()*26) + 97);
            System.out.print(c+ ": ");
            switch(c){
                case 'a':
                case 'e':
                case 'i':
                case 'o':
                case 'u':
                    System.out.println("vocal");
                    break;
                default:
                    System.out.println("consonante");
            }
        }
    }
}
```