

Curso de Java a distancia

Clase 6: Estructura de datos tipo vector.

Hemos empleado variables de distinto tipo para el almacenamiento de datos (variables int, float, String) En esta sección veremos otros tipos de variables que permiten almacenar un conjunto de datos en una única variable.

Un vector es una estructura de datos que permite almacenar un CONJUNTO de datos del MISMO tipo.

Con un único nombre se define un vector y por medio de un subíndice hacemos referencia a cada elemento del mismo (componente)

Problema 1:

Se desea guardar los sueldos de 5 operarios.

Según lo conocido deberíamos definir 5 variables si queremos tener en un cierto momento los 5 sueldos almacenados en memoria.

Empleando un vector solo se requiere definir un único nombre y accedemos a cada elemento por medio del subíndice.

sueldos				
1200	750	820	550	490
sueldos[0]	sueldos[1]	sueldos[2]	sueldos[3]	sueldos[4]

Programa:

```
import java.util.Scanner;
public class PruebaVector1 {
    private Scanner teclado;
    private int[] sueldos;

    public void cargar()
    {
        teclado=new Scanner(System.in);
        sueldos=new int[5];
        for(int f=0;f<5;f++) {
```

```

        System.out.print("Ingrese valor de la componente:");
        sueldos[f]=teclado.nextInt();
    }
}

public void imprimir() {
    for(int f=0;f<5;f++) {
        System.out.println(sueldos[f]);
    }
}

public static void main(String[] ar) {
    PruebaVector1 pv=new PruebaVector1();
    pv.cargar();
    pv.imprimir();
}
}

```

Para la declaración de un vector le antecedemos al nombre los corchetes abiertos y cerrados:

```
private int[] sueldos;
```

Lo definimos como atributo de la clase ya que lo utilizaremos en los dos métodos.

En el método de cargar lo primero que hacemos es crear el vector (en java los vectores son objetos por lo que es necesario proceder a su creación mediante el operador new):

```
sueldos=new int[5];
```

Cuando creamos el vector indicamos entre corchetes la cantidad de elementos que se pueden almacenar posteriormente en el mismo.

Para cargar cada componente debemos indicar entre corchetes que elemento del vector estamos accediendo:

```

for(int f=0;f<5;f++) {
    System.out.print("Ingrese valor de la componente:");
    sueldos[f]=teclado.nextInt();
}

```

La estructura de programación que más se adapta para cargar en forma completa las componentes de un vector es un for, ya que sabemos de antemano la cantidad de valores a cargar.

Cuando f vale cero estamos accediendo a la primer componente del vector (en nuestro caso sería):

```
sueldos[0]=teclado.nextInt();
```

Lo mas común es utilizar una estructura repetitiva for para recorrer cada componente del vector.

Utilizar el for nos reduce la cantidad de código, si no utilizo un for debería en forma secuencial implementar el siguiente código:

```

System.out.print("Ingrese valor de la componente:");
sueldos[0]=teclado.nextInt();
System.out.print("Ingrese valor de la componente:");
sueldos[1]=teclado.nextInt();
System.out.print("Ingrese valor de la componente:");
sueldos[2]=teclado.nextInt();
System.out.print("Ingrese valor de la componente:");

```

```
suealdos[3]=teclado.nextInt();
System.out.print("Ingrese valor de la componente:");
suealdos[4]=teclado.nextInt();
```

La impresión de las componentes del vector lo hacemos en el otro método:

```
public void imprimir() {
    for(int f=0;f<5;f++) {
        System.out.println(suealdos[f]);
    }
}
```

Siempre que queremos acceder a una componente del vector debemos indicar entre corchetes la componente, dicho valor comienza a numerarse en cero y continua hasta un número menos del tamaño del vector, en nuestro caso creamos el vector con 5 elementos:

```
suealdos=new int[5];
```

Por último en este programa creamos un objeto en la main y llamamos a lo métodos de cargar e imprimir el vector:

```
public static void main(String[] ar) {
    PruebaVector1 pv=new PruebaVector1();
    pv.cargar();
    pv.imprimir();
}
```

Problema 2:

Definir un vector de 5 componentes de tipo float que representen las alturas de 5 personas. Obtener el promedio de las mismas. Contar cuántas personas son más altas que el promedio y cuántas más bajas.

Programa:

```
import java.util.Scanner;
public class PruebaVector2 {
    private Scanner teclado;
    private float[] alturas;
    private float promedio;

    public void cargar() {
        teclado=new Scanner(System.in);
        alturas=new float[5];
        for(int f=0;f<5;f++) {
            System.out.print("Ingrese la altura de la persona:");
            alturas[f]=teclado.nextFloat();
        }
    }

    public void calcularPromedio() {
        float suma;
        suma=0;
        for(int f=0;f<5;f++) {
            suma=suma+alturas[f];
        }
        promedio=suma/5;
        System.out.println("Promedio de alturas:"+promedio);
    }

    public void mayoresMenores() {
        int may,men;
```

```

        may=0;
        men=0;
        for(int f=0;f<5;f++) {
            if (alturas[f]>promedio) {
                may++;
            } else {
                if (alturas[f]<promedio) {
                    men++;
                }
            }
        }
    }
    System.out.println("Cantidad de personas mayores al promedio:"+may);
    System.out.println("Cantidad de personas menores al promedio:"+men);
}

public static void main(String[] ar) {
    PruebaVector2 pv2=new PruebaVector2();
    pv2.cargar();
    pv2.calcularPromedio();
    pv2.mayoresMenores();
}
}

```

Definimos como atributo un vector donde almacenaremos las alturas:

```
private float[] alturas;
```

En la carga creamos el vector indicando que reserve espacio para 5 componentes:

```
alturas=new float[5];
```

Procedemos seguidamente a cargar todos sus elementos:

```

for(int f=0;f<5;f++) {
    System.out.print("Ingrese la altura de la persona:");
    alturas[f]=teclado.nextFloat();
}

```

En otro método procedemos a sumar todas sus componentes y obtener el promedio. El promedio lo almacenamos en un atributo de la clase ya que lo necesitamos en otro método:

```

public void calcularPromedio() {
    float suma;
    suma=0;
    for(int f=0;f<5;f++) {
        suma=suma+alturas[f];
    }
    promedio=suma/5;
    System.out.println("Promedio de alturas:"+promedio);
}

```

Por último en un tercer método comparamos cada componente del vector con el atributo promedio, si el valor almacenado supera al promedio incrementamos un contador en caso que sea menor al promedio incrementamos otro contador:

```

public void mayoresMenores() {
    int may,men;
    may=0;
    men=0;
    for(int f=0;f<5;f++) {
        if (alturas[f]>promedio) {

```

```

        may++;
    } else {
        if (alturas[f]<promedio) {
            men++;
        }
    }
}
}
System.out.println("Cantidad de personas mayores al promedio:"+may);
System.out.println("Cantidad de personas menores al promedio:"+men);
}

```

Importante:

En este problema podemos observar una ventaja de tener almacenadas todas las alturas de las personas. Si no conociéramos los vectores tenemos que cargar otra vez las alturas por teclado para compararlas con el promedio.

Mientras el programa está en ejecución tenemos el vector alturas a nuestra disposición. Es importante tener en cuenta que cuando finaliza la ejecución del programa se pierde el contenido de todas las variables (simples y vectores)

Problema 3:

Una empresa tiene dos turnos (mañana y tarde) en los que trabajan 8 empleados (4 por la mañana y 4 por la tarde)

Confeccionar un programa que permita almacenar los sueldos de los empleados agrupados por turno.

Imprimir los gastos en sueldos de cada turno.

Programa:

```

import java.util.Scanner;
public class PruebaVector3 {
    private Scanner teclado;
    private float[] turnoMan;
    private float[] turnoTar;

    public void cargar() {
        teclado=new Scanner(System.in);
        turnoMan=new float[4];
        turnoTar=new float[4];
        System.out.println("Sueldos de empleados del turno de la mañana.");
        for(int f=0;f<4;f++) {
            System.out.print("Ingrese sueldo:");
            turnoMan[f]=teclado.nextFloat();
        }
        System.out.println("Sueldos de empleados del turno de la tarde.");
        for(int f=0;f<4;f++) {
            System.out.print("Ingrese sueldo:");
            turnoTar[f]=teclado.nextFloat();
        }
    }

    public void calcularGastos() {
        float man=0;
        float tar=0;
        for(int f=0;f<4;f++){
            man=man+turnoMan[f];
            tar=tar+turnoTar[f];
        }
        System.out.println("Total de gastos del turno de la mañana:"+man);
        System.out.println("Total de gastos del turno de la tarde:"+tar);
    }
}

```

```

    }

    public static void main(String[] ar){
        PruebaVector3 pv=new PruebaVector3();
        pv.cargar();
        pv.calcularGastos();
    }
}

```

Definimos dos atributos de tipo vector donde almacenaremos los sueldos de los empleados de cada turno:

```

private float[] turnoMan;
private float[] turnoTar;

```

Creamos los vectores con cuatro elementos cada uno:

```

    turnoMan=new float[4];
    turnoTar=new float[4];

```

Mediante dos estructuras repetitivas procedemos a cargar cada vector:

```

    System.out.println("Sueldos de empleados del turno de la mañana.");
    for(int f=0;f<4;f++) {
        System.out.print("Ingrese sueldo:");
        turnoMan[f]=teclado.nextFloat();
    }
    System.out.println("Sueldos de empleados del turno de la tarde.");
    for(int f=0;f<4;f++) {
        System.out.print("Ingrese sueldo:");
        turnoTar[f]=teclado.nextFloat();
    }
}

```

En otro método procedemos a sumar las componentes de cada vector y mostrar dichos acumuladores:

```

    float man=0;
    float tar=0;
    for(int f=0;f<4;f++){
        man=man+turnoMan[f];
        tar=tar+turnoTar[f];
    }
    System.out.println("Total de gastos del turno de la mañana:"+man);
    System.out.println("Total de gastos del turno de la tarde:"+tar);

```

Problemas propuestos

1. Desarrollar un programa que permita ingresar un vector de 8 elementos, e informe:
El valor acumulado de todos los elementos del vector.
El valor acumulado de los elementos del vector que sean mayores a 36.
Cantidad de valores mayores a 50.
2. Realizar un programa que pida la carga de dos vectores numéricos enteros de 4 elementos. Obtener la suma de los dos vectores, dicho resultado guardarlo en un tercer vector del mismo tamaño. Sumar componente a componente.
3. Se tienen las notas del primer parcial de los alumnos de dos cursos, el curso A y el curso B, cada curso cuenta con 5 alumnos.
Realizar un programa que muestre el curso que obtuvo el mayor promedio general.
4. Cargar un vector de 10 elementos y verificar posteriormente si el mismo está ordenado de menor a mayor.

Solución

```
import java.util.Scanner;
public class PruebaVector4 {
    private Scanner teclado;
    private int[] vec;

    public void cargar() {
        teclado=new Scanner(System.in);
        vec=new int[8];
        for(int f=0;f<8;f++) {
            System.out.print("Ingrese elemento:");
            vec[f]=teclado.nextInt();
        }
    }

    public void acumularElementos() {
        int suma=0;
        for(int f=0;f<8;f++) {
            suma=suma+vec[f];
        }
        System.out.println("La suma de los 8 elementos es:"+suma);
    }

    public void acumularMayores36() {
        int suma=0;
        for(int f=0;f<8;f++) {
            if (vec[f]>36) {
                suma=suma+vec[f];
            }
        }
        System.out.println("La suma de los elementos mayores a 36 es:"+suma);
    }

    public void cantidadMayores50() {
        int cant=0;
        for(int f=0;f<8;f++) {
            if (vec[f]>50) {
                cant++;
            }
        }
        System.out.println("La cantidad de valores mayores a 50 es:"+cant);
    }

    public static void main(String[] ar) {
        PruebaVector4 pv=new PruebaVector4();
        pv.cargar();
        pv.acumularElementos();
        pv.acumularMayores36();
        pv.cantidadMayores50();
    }
}
```

```
import java.util.Scanner;
public class PruebaVector5 {
    private Scanner teclado;
    private int[] vec1;
```

```

private int[] vec2;
private int[] vecSuma;

public void cargar() {
    teclado=new Scanner(System.in);
    vec1=new int[4];
    vec2=new int[4];
    System.out.println("Carga del primer vector.");
    for(int f=0;f<4;f++) {
        System.out.print("Ingrese elemento:");
        vec1[f]=teclado.nextInt();
    }
    System.out.println("Carga del segundo vector.");
    for(int f=0;f<4;f++) {
        System.out.print("Ingrese elemento:");
        vec2[f]=teclado.nextInt();
    }
}

public void sumarizarVectores() {
    vecSuma=new int[4];
    for(int f=0;f<4;f++) {
        vecSuma[f]=vec1[f]+vec2[f];
    }
    System.out.println("Vector resultante.");
    for(int f=0;f<4;f++) {
        System.out.println(vecSuma[f]);
    }
}

public static void main(String[] ar) {
    PruebaVector5 pv=new PruebaVector5();
    pv.cargar();
    pv.sumarizarVectores();
}
}

```

```

import java.util.Scanner;
public class PruebaVector6 {
    private Scanner teclado;
    private int[] cursoa;
    private int[] cursob;
    private int[] vecSuma;

    public void cargar() {
        teclado=new Scanner(System.in);
        cursoa=new int[5];
        cursob=new int[5];
        System.out.println("Carga de notas del curso A");
        for(int f=0;f<5;f++) {
            System.out.print("Ingrese nota:");
            cursoa[f]=teclado.nextInt();
        }
        System.out.println("Carga de notas del curso B");
        for(int f=0;f<5;f++) {
            System.out.print("Ingrese nota:");
            cursob[f]=teclado.nextInt();
        }
    }
}

```



```

    }
}

public void calcularPromedios() {
    int suma1=0;
    int suma2=0;
    for(int f=0;f<5;f++) {
        suma1=suma1+cursoa[f];
        suma2=suma2+cursob[f];
    }
    int promedioa=suma1/5;
    int promediob=suma2/5;
    if (promedioa>promediob) {
        System.out.println("El curso A tiene un promedio mayor.");
    } else {
        System.out.println("El curso B tiene un promedio mayor.");
    }
}

public static void main(String[] ar) {
    PruebaVector6 pv=new PruebaVector6();
    pv.cargar();
    pv.calcularPromedios();
}
}

```

```

import java.util.Scanner;
public class PruebaVector7 {
    private Scanner teclado;
    private int[] vec;

    public void cargar() {
        teclado=new Scanner(System.in);
        vec=new int[10];
        for(int f=0;f<10;f++) {
            System.out.print("Ingrese elemento.");
            vec[f]=teclado.nextInt();
        }
    }

    public void verificarOrdenado() {
        int orden=1;
        for(int f=0;f<9;f++) {
            if (vec[f+1]<vec[f]) {
                orden=0;
            }
        }
        if (orden==1) {
            System.out.println("Esta ordenado de menor a mayor");
        } else {
            System.out.println("No esta ordenado de menor a mayor");
        }
    }

    public static void main(String[] ar) {
        PruebaVector7 pv=new PruebaVector7();
        pv.cargar();
    }
}

```

```

        pv.verificarOrdenado();
    }
}

```

Vector (Tamaño de un vector)

Como hemos visto cuando se crea un vector indicamos entre corchetes su tamaño:

```
suealdos=new int[5];
```

Luego cuando tenemos que recorrer dicho vector disponemos una estructura repetitiva for:

```

for(int f=0;f<5;f++) {
    System.out.print("Ingrese valor de la componente:");
    suealdos[f]=teclado.nextInt();
}

```

Como vemos el for se repite mientras el contador f vale menos de 5. Esta estructura repetitiva es idéntica cada vez que recorremos el vector.

Que pasa ahora si cambiamos el tamaño del vector cuando lo creamos:

```
suealdos=new int[7];
```

Con esto tenemos que cambiar todos los for que recorren dicho vector. Ahora veremos que un vector al ser un objeto tiene un atributo llamado length que almacena su tamaño. Luego podemos modificar todos los for con la siguiente sintaxis:

```

for(int f=0;f<suealdos.length;f++) {
    System.out.print("Ingrese valor de la componente:");
    suealdos[f]=teclado.nextInt();
}

```

También podemos pedir al usuario que indique el tamaño del vector en tiempo de ejecución, en estos casos se hace imprescindible el empleo del atributo length.

Problema 1:

Se desea almacenar los sueldos de operarios. Cuando se ejecuta el programa se debe pedir la cantidad de sueldos a ingresar. Luego crear un vector con dicho tamaño.

Programa:

```

import java.util.Scanner;
public class PruebaVector8 {
    private Scanner teclado;
    private int[] suealdos;

    public void cargar()
    {
        teclado=new Scanner(System.in);
        System.out.print("Cuántos sueldos cargará:");
        int cant;
        cant=teclado.nextInt();
        suealdos=new int[cant];
        for(int f=0;f<suealdos.length;f++) {
            System.out.print("Ingrese sueldo:");

```

```

        sueldos[f]=teclado.nextInt();
    }
}

public void imprimir() {
    for(int f=0;f<sueldos.length;f++) {
        System.out.println(sueldos[f]);
    }
}

public static void main(String[] ar) {
    PruebaVector8 pv=new PruebaVector8();
    pv.cargar();
    pv.imprimir();
}
}

```

La definición del vector no varía:

```
private int[] sueldos;
```

Luego para la creación del mismo ingresamos una variable entera y la utilizamos como subíndice en el momento de la creación del vector:

```

System.out.print("Cuántos sueldos cargará:");
int cant;
cant=teclado.nextInt();
sueldos=new int[cant];

```

Luego las estructuras repetitivas las acotamos accediendo al atributo length del vector:

```

for(int f=0;f<sueldos.length;f++) {
    System.out.print("Ingrese sueldo:");
    sueldos[f]=teclado.nextInt();
}

```

Problemas propuestos

1. Desarrollar un programa que permita ingresar un vector de n elementos, ingresar n por teclado. Luego imprimir la suma de todos sus elementos

Solución

```
import java.util.Scanner;
public class PruebaVector9 {
    private Scanner teclado;
    private int[] vec;

    public void cargar() {
        teclado=new Scanner(System.in);
        System.out.print("Cuantos elementos tiene el vector:");
        int n;
        n=teclado.nextInt();
        vec=new int[n];
        for(int f=0;f<vec.length;f++) {
            System.out.print("Ingrese elemento:");
            vec[f]=teclado.nextInt();
        }
    }

    public void acumularElementos() {
        int suma=0;
        for(int f=0;f<vec.length;f++) {
            suma=suma+vec[f];
        }
        System.out.println("La suma de los elementos es:"+suma);
    }

    public static void main(String[] ar) {
        PruebaVector9 pv=new PruebaVector9();
        pv.cargar();
        pv.acumularElementos();
    }
}
```

Vectores paralelos

Este concepto se da cuando hay una relación entre las componentes de igual subíndice (misma posición) de un vector y otro.

<i>nombres</i>	Juan	Ana	Marcos	Pablo	Laura
<i>edades</i>	12	21	27	14	21

Si tenemos dos vectores de 5 elementos cada uno. En uno se almacenan los nombres de personas en el otro las edades de dichas personas.

Decimos que el vector nombres es paralelo al vector edades si en la componente 0 de cada vector se almacena información relacionada a una persona (Juan - 12 años)

Es decir hay una relación entre cada componente de los dos vectores.

Esta relación la conoce únicamente el programador y se hace para facilitar el desarrollo de algoritmos que procesen los datos almacenados en las estructuras de datos.

Problema 1:

Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años)

Programa:

```
import java.util.Scanner;
public class PruebaVector10 {
    private Scanner teclado;
    private String[] nombres;
    private int[] edades;

    public void cargar() {
        teclado=new Scanner(System.in);
        nombres=new String[5];
        edades=new int[5];
        for(int f=0;f<nombres.length;f++) {
            System.out.print("Ingrese nombre:");
            nombres[f]=teclado.next();
            System.out.print("Ingrese edad:");
            edades[f]=teclado.nextInt();
        }
    }

    public void mayoresEdad() {
        System.out.println("Personas mayores de edad.");
        for(int f=0;f<nombres.length;f++) {
            if (edades[f]>=18) {
                System.out.println(nombres[f]);
            }
        }
    }

    public static void main(String[] ar) {
        PruebaVector10 pv=new PruebaVector10();
        pv.cargar();
        pv.mayoresEdad();
    }
}
```

Definimos los dos vectores:

```
private String[] nombres;
private int[] edades;
```

Creamos los dos vectores con 5 elementos cada uno:

```
nombres=new String[5];
edades=new int[5];
```

Mediante un for procedemos a la carga de los elementos de los vectores:

```
for(int f=0;f<nombres.length;f++) {
    System.out.print("Ingrese nombre:");
    nombres[f]=teclado.next();
    System.out.print("Ingrese edad:");
    edades[f]=teclado.nextInt();
}
```

Podemos utilizar el length de cualquiera de los dos vectores, ya que tienen el mismo tamaño.

Para imprimir los nombres de las personas mayores de edad verificamos cada componente del vector de edades, en caso que sea igual o mayor o 18 procedemos a mostrar el elemento de la misma posición del otro vector:

```
for(int f=0;f<nombres.length;f++) {  
    if (edades[f]>=18) {  
        System.out.println(nombres[f]);  
    }  
}
```

Muchas gracias hasta la próxima clase.

Alsina 16 [B1642FNB] San Isidro | Pcia. De Buenos Aires |Argentina |

TEL.: [011] 4742-1532 o [011] 4742-1665 |

www.institutosanisidro.com.ar info@institutosanisidro.com.ar