LABORATORIO N°11
PROGRAMACIÓN DE COMPUTADORES

Objetivos del Laboratorio:

Al término de la clase los alumnos serán capaces de:

- Diseñar algoritmos no secuenciales definiendo las entradas, el proceso y la salida que dará solución a un determinado problema
- Comprender el concepto de arreglo y su implementación en Java
- Probar algoritmos usando arreglos en programación Java

Actividades:

Conceptos: Arreglos

En general, un arreglo es una colección de datos del mismo tipo, que se almacenan en posiciones consecutivas o contiguas de memoria y reciben un nombre común. La colección de datos es finita, homogénea y ordenada.

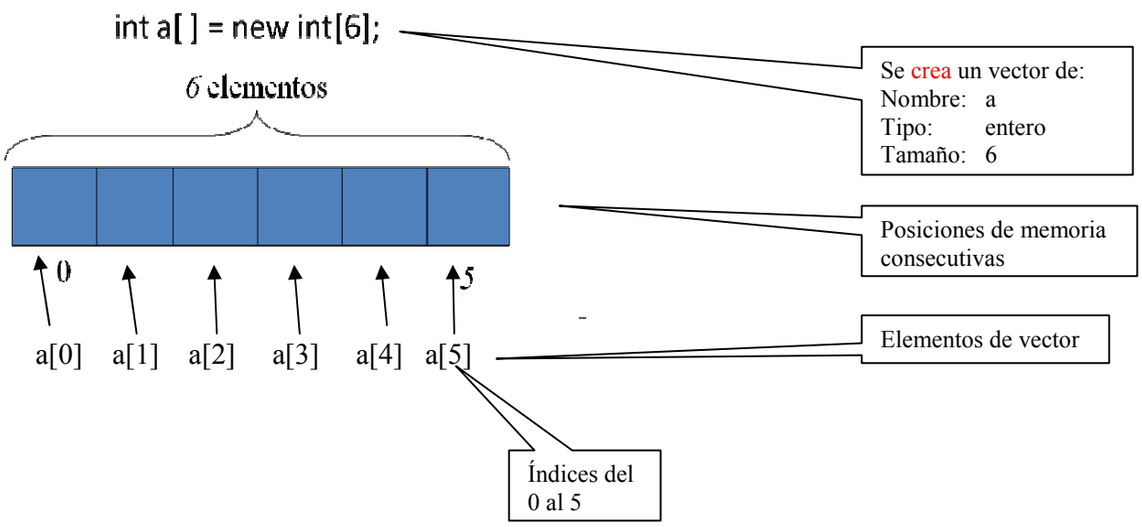
Finita: Todo arreglo tiene un límite; el número máximo de elementos que formarán parte del arreglo.

Homogénea: Todos los elementos del arreglo deben ser del mismo tipo.

Ordenada: Se sabe cuál es el primer elemento, el segundo, el tercero,.... y el n-ésimo el elemento.

Existen arreglos unidimensionales (vectores), bidimensionales (matrices o tablas) y multidimensionales (tres o más dimensiones)

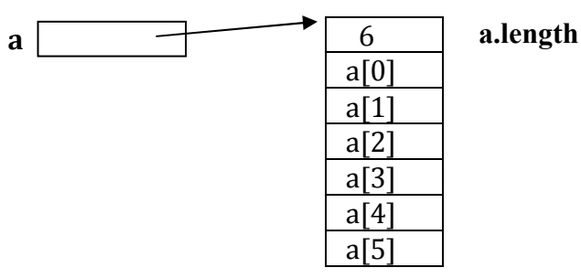
Ejemplo:



En Java, cada elemento de un vector es de un tipo primitivo, o bien una referencia a un objeto; y cada elemento de una matriz multidimensional es, a su vez, una referencia a otra matriz.

Representación de un vector en memoria:

int a[] = new int[6]



En la dirección **a** hay un puntero que apunta a una posición de memoria en donde está el tamaño del vector (32 bit cada posición = 4 byte) seguido de los elementos del vector, cada una de 32 bit.

La longitud de un vector se puede obtener con el método length perteneciente a la clase String que devuelve un valor entero que indica el número de elementos del vector.

Para crear y utilizar una vector hay que realizar 3 operaciones:

- Declararlo
- Crearlo
- Iniciarlo

Declarar un vector, hay dos alternativas:

```
tipo[] nombre;
tipo nombre[];
```

tipo: tipo de elementos del vector. Puede ser de cualquier tipo primitivo o referenciado;
nombre: es el identificador que nombra al vector.

Ejemplo:

```
int[] suma;
float[] temperatura;
int notas[];
double promedios[];
```

Las declaraciones no especifican el tamaño del vector. El tamaño será especificado cuando se cree el vector, operación que se hará durante la ejecución del programa.

Crear un vector:

Consiste en reservar la cantidad de memoria necesaria para contener todos sus elementos y asignar al nombre del vector una referencia a ese bloque.

```
nombre = new tipo[tamaño];
```

tipo: tipo de elementos del vector.
nombre: es el identificador que nombra al vector.
tamaño: entero que indica el número de posiciones disponible para el vector.
new: implementa el vector como un nuevo objeto

Ejemplo:

```
suma = new int[10];
temperatura = new float[31];
notas = new int[12];
promedios = new double[12];
```

Declarar y crear un vector:

Es usual declarar y crear un vector en una misma línea. Es una combinación de las anteriores.

```
tipo[] nombre = new tipo[tamaño];
tipo nombre[]= new tipo[tamaño];
```

Ejemplo:

```
int[] suma = new int[10];
float[] temperatura = new float[31];
int notas[] = new int[12];
double promedios[] = new double[12];
```

Iniciar un vector:

Al ser creado un vector numérico, sus elementos son iniciados con el valor cero, automáticamente. Sin embargo, existe la opción de asignarles un valor distinto al inicio.

Ejemplo: suma[0] = 1; suma[1] = 3; suma[2] = 4;…….; suma[9] = 15;

Declarar, crear e iniciar un vector:

Un vector se puede declarar, crear e inicializar en una misma línea.

Ejemplo:

```
int[] datos = {1,2,3,4,5,6,7,8,9};
```

El tamaño del vector está determinado por la cantidad de elementos.

Ejercicio 1: Confeccionar un programa que almacene 6 notas enteras en un vector y después las muestre por pantalla.

Para el ejercicio se pide:

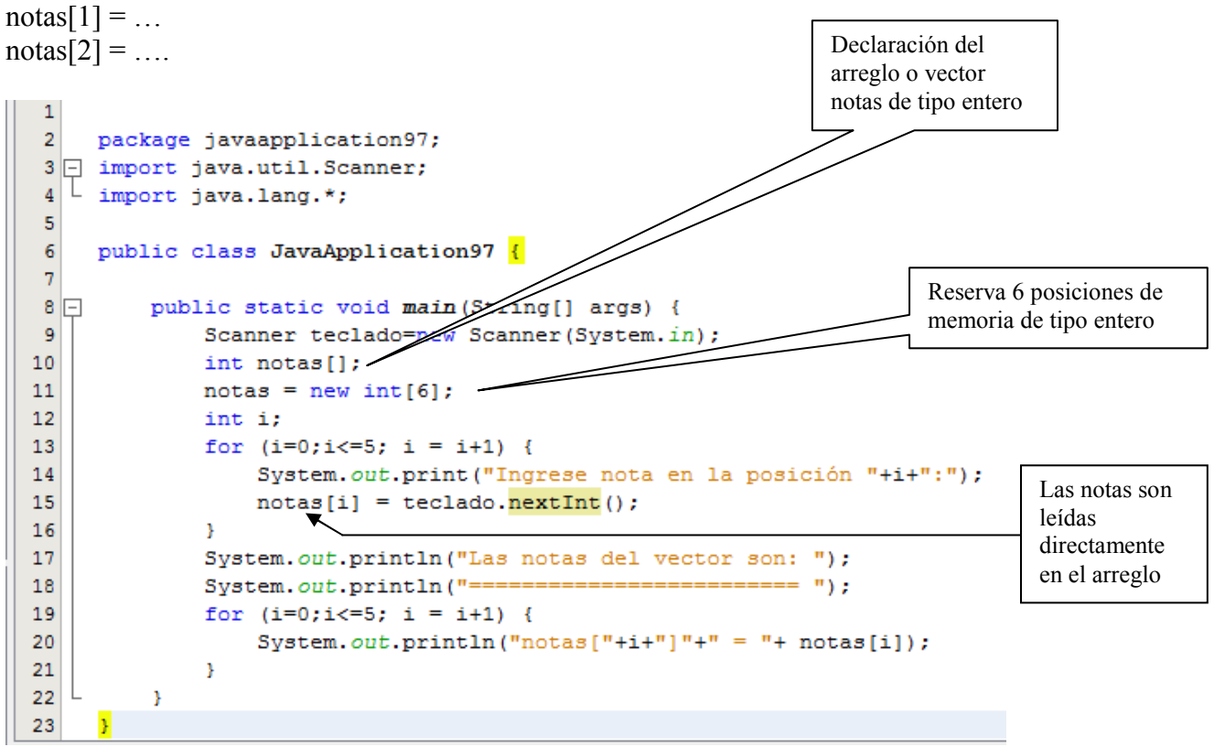
a. Describir las Entradas, Proceso y Salida
b. Diseñar el algoritmo/programa en Java
c. Ejecutar y probar el algoritmo

Entradas: Las seis notas que ingresa el usuario

Proceso: Declaramos un arreglo o vector (notas) para almacenar las notas. Usamos un ciclo for que comienzan en 0 y termina en 5. Para mostrar las notas usamos otro ciclo for. Para recorrer el arreglo desde 0 hasta 5.

Salida: Las 6 notas con el siguiente formato:

notas[0] = …
notas[1] = …
notas[2] = ….




```
1
2   package javaapplication97;
3   import java.util.Scanner;
4   import java.lang.*;
5
6   public class JavaApplication97 {
7
8       public static void main(String[] args) {
9           Scanner teclado=new Scanner(System.in);
10          int notas[];
11          notas = new int[6];
12          int i;
13          for (i=0;i<=5; i = i+1) {
14              System.out.print("Ingrese nota en la posición "+i+":");
15              notas[i] = teclado.nextInt();
16          }
17          System.out.println("Las notas del vector son: ");
18          System.out.println("========================== ");
19          for (i=0;i<=5; i = i+1) {
20              System.out.println("notas["+i+"]"+" = "+ notas[i]);
21          }
22      }
23  }
```

Ejecución del Ejercicio 1:

```
run:
Ingrese nota en la posición 0:1
Ingrese nota en la posición 1:2
Ingrese nota en la posición 2:3
Ingrese nota en la posición 3:4
Ingrese nota en la posición 4:5
Ingrese nota en la posición 5:6
Las notas del vector son:
==========================
notas[0] = 1
notas[1] = 2
notas[2] = 3
notas[3] = 4
notas[4] = 5
notas[5] = 6
BUILD SUCCESSFUL (total time: 12 seconds)
```

Ejercicio 2: Confeccionar un programa que almacene 6 notas en un vector y las muestre por pantalla. Seguidamente, sume a cada nota las anteriores de tal manera que la última posición almacene la suma de todas las notas anteriores.

Para el ejercicio se pide:

a. Describir las Entradas, Proceso y Salida
b. Diseñar el algoritmo/programa en Java
c. Ejecutar y probar el algoritmo

Entradas: Las seis notas que ingresa el usuario

Proceso: Declaramos un arreglo o vector (notas) para almacenar las notas. Usamos un ciclo for que comienzan en 0 y termina en 5. Para mostrar las notas usamos otro ciclo for.

La siguiente asignación nos permite sumar a cada posición las notas anteriores:

```
notas[i] = notas[i] + notas[i-1];  //con i variando de 1 a 5
```

Salida: El contenido o los elementos del arreglo.

```java
package javaapplication98;
import java.util.Scanner;
import java.lang.*;

public class JavaApplication98 {
   public static void main(String[] args) {
        Scanner teclado=new Scanner(System.in);
        int notas[];
        notas = new int[6];
        int i;
        for (i=0;i<=5; i = i+1) {
            System.out.print("Ingrese nota en la posición "+i+":");
            notas[i] = teclado.nextInt();
        }
        System.out.println("Las notas del vector son: ");
        System.out.println("========================= ");
        for (i=0;i<=5; i = i+1) {
            System.out.print("notas["+i+"]"+" = "+ notas[i]+" - ");
        }
        System.out.println("");
        for (i=1;i<=5; i = i+1) {
            notas[i] = notas[i] + notas[i-1];
        }
        System.out.println("Las notas sumadas del vector son: ");
        System.out.println("================================= ");
        for (i=0;i< notas.length; i = i+1) {
            System.out.print("notas["+i+"]"+" = "+ notas[i]+" - ");
        }
        System.out.println("");
    }
```

Notar el uso del método length para lo cual se importó java.lang.*

Ejecución Ejercicio 2:

```
run:
Ingrese nota en la posición 0:1
Ingrese nota en la posición 1:2
Ingrese nota en la posición 2:3
Ingrese nota en la posición 3:4
Ingrese nota en la posición 4:5
Ingrese nota en la posición 5:6
Las notas del vector son:
=========================
notas[0] = 1 - notas[1] = 2 - notas[2] = 3 - notas[3] = 4 - notas[4] = 5 - notas[5] = 6 -
Las notas sumadas del vector son:
=================================
notas[0] = 1 - notas[1] = 3 - notas[2] = 6 - notas[3] = 10 - notas[4] = 15 - notas[5] = 21 -
BUILD SUCCESSFUL (total time: 8 seconds)
```

Ejercicio 3: Suponga que tiene un vector con 6 números enteros. Escriba un algoritmo/programa que encuentre el mayor de los seis números e indique la posición en la que se encuentra.
Para el ejercicio se pide:

a. Describir las Entradas, Proceso y Salida
b. Diseñar el algoritmo/programa en Java
c. Ejecutar y probar el algoritmo
d. Confeccione la Tabla de Seguimiento de programa

Ejercicio 4: Que debe modificar en el programa del Ejercicio 3 para obtener el menor de los números y su posición.
Ejercicio 5: Suponga que tiene un arreglo o vector con los valores netos de 10 productos de una tienda. Se pide diseñar un programa que a partir de estos valores cree un arreglo contenga los precios de ventas de los productos. Estos precios deben incluir el IVA y una utilidad del 30%. Al final muestre la siguiente lista:

Producto	Valor Neto	Precio Venta
1	1000	1547
2	1500	2320

…..
Ejercicio 6: Diseñe un programa que genere los 10 primeros números de Fibonacci, los almacene en un vector y los muestre en pantalla. FIB(0) = 0, FIB(1) = 1 y en general FIB (n) = FIB(n-1)+FIB(n-2)
Ejercicio 7: Diseñar un programa que lea 10 números enteros, los almacene en un vector y determine las posiciones en la que se encuentran los números que terminan en 5.
Ejercicio 8: Diseñar un programa que lea 10 números enteros, los almacene en un vector y determine cuántos de ellos son menores que el promedio.

Ejercicio 9: Diseñar un programa que lea 10 números enteros, los almacene en un vector y muestre en pantalla todos los enteros comprendidos entre 1 y cada uno de los números almacenados en el vector.

Ejercicio 10: Diseñar un programa que lea 10 números enteros, los almacene en un vector y determine la posición del número cuya suma de dígitos sea la mayor.