

Arreglos

Contenido

Introducción

Arreglos

Declarando y asignando memoria para los arreglos

Ejemplos usando arreglos

Asignar memoria para un arreglo e inicializar sus elementos

Total de los elementos de un arreglo

Usando histogramas para desplegar gráficamente los datos

Usando los elementos de un arreglo como contadores

Usando arreglos para analizar resultados de encuestas



Introducción

- Estructuras de datos
 - Están constituidos por elementos del mismo tipo
 - Estáticos: permanecen del mismo tamaño una vez que fueron creados



Arreglos (1)

- Un grupo de localidades de memoria consecutivas
 - Con el mismo nombre
 - Del mismo tipo
- Para referirse a un elemento particular del arreglo se hace por el *número de posición*
- Es posible referirse a cualquier elemento dando el nombre del arreglo seguido del número de posición del elemento dentro de paréntesis cuadrados ([])
- El primer elemento es el *elemento zero*

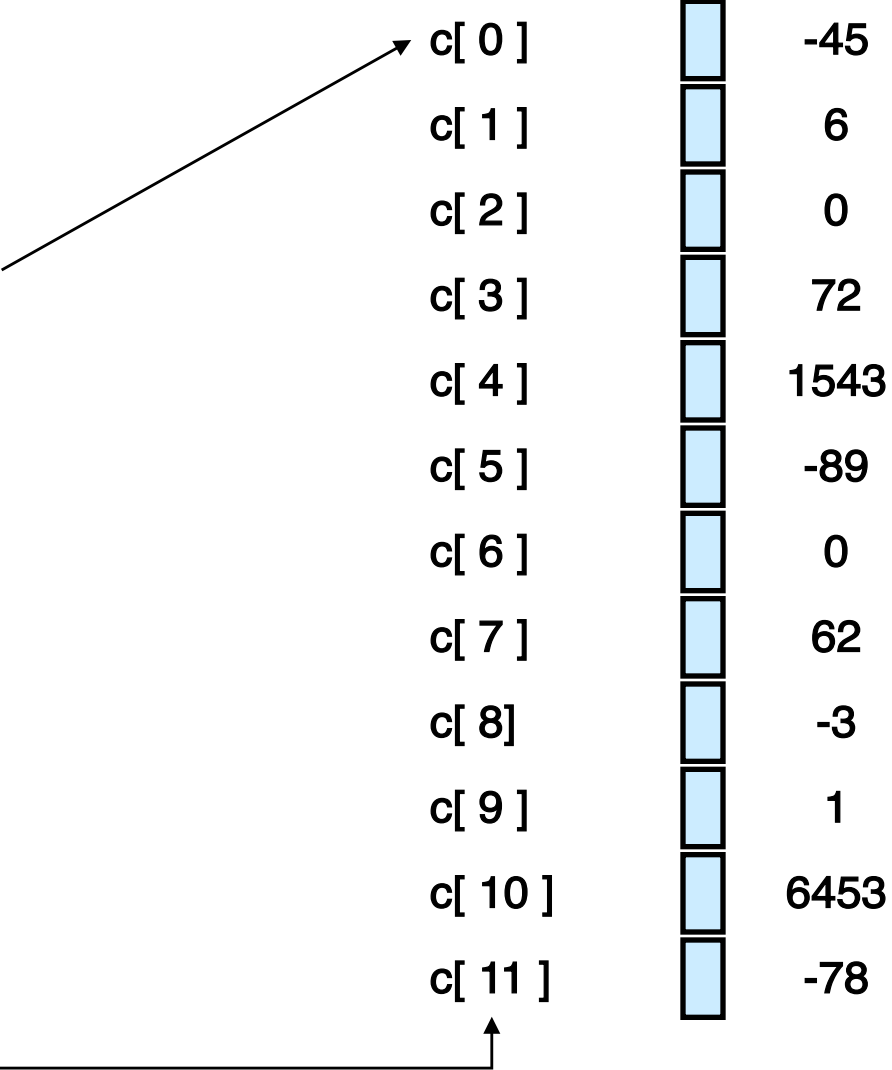


Arreglos (2)

Nombre del Arreglo
(Nota que todos los
elementos del arreglo
tienen el mismo
nombre, c)

Número de posición
(índice) del elemento
dentro del arreglo c

Un arreglo de 12 elementos



c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6453
c[11]	-78



Declarando y asignando memoria para los arreglos

- El programador especifica el tipo de elementos de un arreglo
- El operador **new** es usado para asignar memoria dinámicamente para el número de elementos en el arreglo
- Si se desea inicializar un arreglo al declararlo, esto se debe hacer en una misma sentencia.
- En arreglos de valor, cada elemento contiene un valor del tipo declarado
- En arreglos de referencia, cada elemento del arreglo es una referencia a un objeto del tipo de dato del arreglo



Asignar memoria para un arreglo e inicializar sus elementos

- Se puede asignar memoria a los arreglos usando la palabra clave **new** para especificar cuantos elementos debería contener el arreglo
- Los arreglos pueden ser inicializados con una *lista de inicializadores*
 - Asignar memoria para el arreglo: el número de elementos en la lista de inicializadores determina el tamaño del arreglo
 - Los elementos en el arreglo son inicializados con los valores en la lista de inicializadores





Outline

Diferentes formas de inicializar arreglos

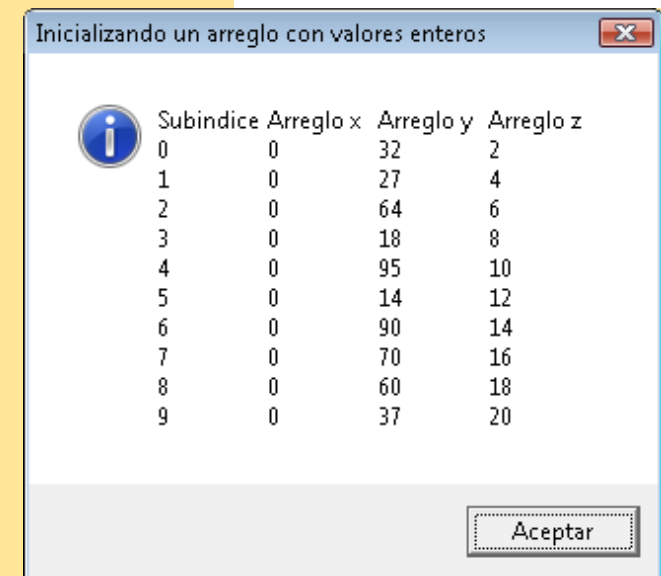
```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
namespace Arreglos
{
    class InicializaArreglos
    {
        static void Main( string[] args)
        {
            string salida = "";
            int[] x;      // Declara la referencia a un arreglo
            x = new int[10]; // Reserva memoria y establece no de elementos

            // Lista de inicializadores, especifica el numero de elementos
            // y el valor de cada elemento
            int[] y = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
            const int TAMANO_ARREGLO = 10; // constante con nombre
            int[] z;      // referencia a un arreglo

            // reserva memoria para TAMANO_ARREGLO(ej., 10) elementos
            z = new int[ TAMANO_ARREGLO ];

            // pone valores en el arreglo
            for ( int i = 0; i < z.Length; i++ )
                z[ i ] = 2 + 2 * i;
            salida += "Subindice\tArreglo x\tArreglo y\tArreglo z\n";

            // salida de los valores para cada arreglo
            for (int i = 0; i < TAMANO_ARREGLO; i++)
                salida += i + "\t" + x[i] + "\t" + y[i] +
                    "\t" + z[i] + "\n";
            MessageBox.Show(salida,
                "Iniciando un arreglo con valores enteros",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```





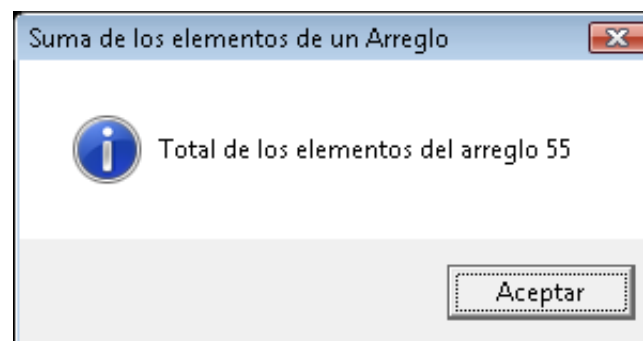
Outline

Suma de los elementos de un arreglo

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
namespace Arreglos
{
    class SumaArreglo
    {
        static void Main(string[] args)
        {
            int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
            int total = 0;

            for (int i = 0; i < a.Length; i++)
                total += a[i];

            MessageBox.Show("Total de elementos del arreglo " + total,
                "Suma de los elementos de un Arreglo",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

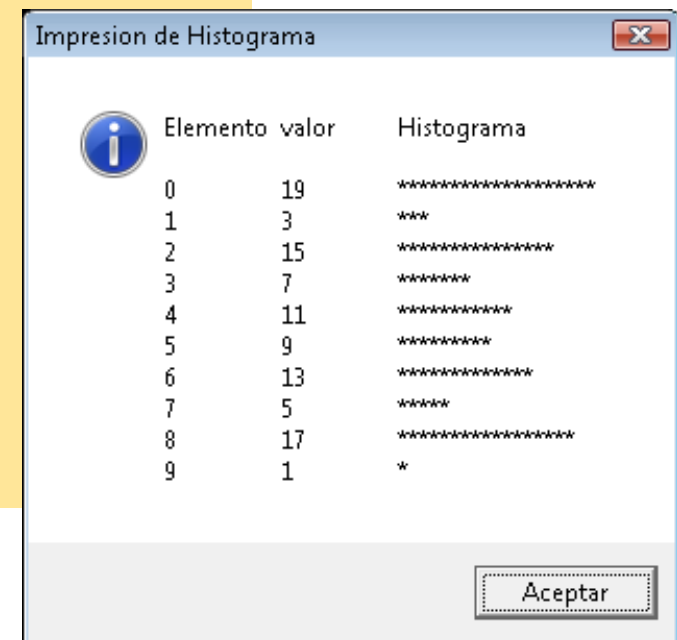




Outline

Usar datos para
crear un histograma

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
namespace Arreglos
{
    class Histograma
    {
        static void Main(string[] args)
        {
            int[] n = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
            string salida = "";
            salida += "Elemento\tvalor\tHistograma\n";
            // construye salida
            for (int i = 0; i < n.Length; i++)
            {
                salida += "\n" + i + "\t" + n[i] + "\t";
                for (int j = 1; j <= n[i]; j++) // imprime barra
                    salida += "*";
            }
            MessageBox.Show(salida, "Impresion de Histograma",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
        }
    }
}
```



Usando los elementos de un arreglo como contadores

- Usar los elementos de un arreglo para mantener el contador del número de ocurrencias
- Ejemplo – Programa Lanzar Dados
 - Usa el valor del dado lanzado como índice del arreglo
 - Incrementa el elemento del arreglo correspondiente cuando un dado es lanzado





Outline

Lanzar Dados (1)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
namespace Arreglos02
{
    public partial class Form1 : Form
    {
        Random numeroAleatorio = new Random(); // Crea un objeto aleatorio
        int[] frecuencia = new int[7];
        public Form1()
        {
            InitializeComponent();
        }
        private void btnLanzar_Click(object sender, EventArgs e)
        {
            // pasa las etiquetas a un metodo que
            // aleatoriamente asigna la cara al dado
            DespliegaDado(lblDado1);
            DespliegaDado(lblDado2);
            DespliegaDado(lblDado3);
            DespliegaDado(lblDado4);
            DespliegaDado(lblDado5);
            DespliegaDado(lblDado6);
            DespliegaDado(lblDado7);
            DespliegaDado(lblDado8);
            DespliegaDado(lblDado9);
            DespliegaDado(lblDado10);
            DespliegaDado(lblDado11);
            DespliegaDado(lblDado12);

            double total = 0;
            for (int i = 1; i < 7; i++)
                total += frecuencia[i];

            txtDespliega.Text = "Cara\tFrecuencia\tProcentaje\n";
        }
    }
}
```



Outline

Lanzar Dados (2)

```
// imprime los valores de las frecuencias
for (int x = 1; x < frecuencia.Length; x++)
{
    txtDespliega.Text += x + "\t" +
        frecuencia[x] + "\t\t" + String.Format("{0:N}",
        frecuencia[x] / total * 100) + "%\n";
}
}
public void DespliegaDado(Label lblDado)
{
    int cara = numeroAleatorio.Next(1, 7);
    lblDado.Image = Image.FromFile(
        Directory.GetCurrentDirectory() +
        @"\imagenes\dado" + cara + ".gif");
    frecuencia[cara]++;
}
}
```



Outline

Encuesta Estudiantes

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
namespace Arreglos
{
    class EncuestaEstudiantes
    {
        static void Main(string[] args)
        {
            int[] respuestas = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10, 1,
                                6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7,
                                5, 6, 6, 5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };

            int[] frecuencia = new int[11];
            string salida = "";

            // incrementa la frecuencia para cada respuesta
            for (int respuesta = 0; respuesta < respuestas.Length; respuesta++)
                ++frecuencia[respuestas[respuesta]];

            salida += "tasa\tfrecuencia\n";

            // salida de los resultados
            for (int tasa = 1; tasa < frecuencia.Length; tasa++)
                salida += tasa + "\t" + frecuencia[tasa] + "\n";
            MessageBox.Show(salida, "Encuesta de estudiantes",
                            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

