Universidad Americana

Facultad de Ingeniería y Arquitectura

Ejercicios en C# con arreglos, métodos y paso de parámetros

Materia:

Metodología y programación estructurada

Elaborado por:

Jonathan Josué Rivera Guido

Diego Alejandro Rodriguez Luque

Docente

Silvia Gigdalia Ticay López

Fecha 09/23/2024

- Ejercicio #1(Ejercicios de evaluación)

El código permite al usuario ingresar los datos de hasta 10 personas y realizar varias operaciones sobre esa información, como mostrar todos los nombres, filtrar por edad o buscar personas por nombre.

Arreglo Persona[] personas = new Persona[10];: Se crea un arreglo para almacenar hasta 10 objetos del tipo Persona, que es una estructura definida más adelante.

int contador = 0;: El contador rastrea cuántas personas se han ingresado.

bool continuar = true;: Una variable booleana que controla si el bucle para ingresar datos sigue activo.

Primer bucle while (continuar && contador < 10)

Este bucle permite al usuario ingresar los datos de una persona en cada iteración, mientras el contador sea menor que 10 y el usuario desee continuar. Se llama al método LeerDatosPersona() que solicita y almacena la información de una persona. Después de ingresar los datos de una persona, el programa pregunta si el usuario desea continuar. Si la respuesta es "s", el bucle se repite, de lo contrario, se detiene.

Segundo bucle while (true)

Este es un bucle infinito que muestra un menú al usuario hasta que elija salir.

almacenarDatos.MostrarMenu();: Muestra el menú de opciones con varias operaciones que el usuario puede realizar sobre la lista de personas ingresadas. luego se captura la opción que el usuario elige.

2. Clase almacenarDatos

Esta clase contiene métodos relacionados con la entrada, almacenamiento y visualización de datos.

Estructura Persona

La estructura Persona contiene cuatro atributos:

- nombre: el nombre de la persona (tipo string).
- dirección: la dirección de la persona (tipo string).
- teléfono: el número de teléfono de la persona (tipo int).

- edad: la edad de la persona (tipo int).

Método LeerDatosPersona()

Este método solicita al usuario que ingrese los datos de una persona (nombre, dirección, teléfono y edad) y devuelve un objeto Persona con estos datos.

Método MostrarMenu()

Este método muestra un menú con cuatro opciones:

- Mostrar la lista de todos los nombres.
- Mostrar las personas de una cierta edad.
- Mostrar las personas que coincidan con un nombre.
- Salir del programa.

Método MostrarNombres()

Muestra todos los nombres de las personas ingresadas en el arreglo personas. Solo itera sobre los elementos del arreglo que contienen datos, controlado por el contador.

Método MostrarPersonasPorEdad()

Filtra y muestra las personas cuya edad coincida con la ingresada por el usuario. Usa un bucle para recorrer el arreglo y verifica si la edad de cada persona es igual a la edad dada. Si encuentra una coincidencia, imprime los detalles de esa persona.

Método MostrarPersonasPorNombre()

Busca y muestra las personas cuyo nombre coincida con el ingresado por el usuario, ignorando mayúsculas y minúsculas. Usa la comparación Equals con la opción OrdinalIgnoreCase para que la comparación no distinga entre mayúsculas y minúsculas.

3. Operaciones del menú

Dependiendo de la opción seleccionada en el menú, el programa realiza diferentes acciones:

Mostrar la lista de todos los nombres: Llama al método MostrarNombres. Mostrar personas de una cierta edad: Solicita una edad al usuario y llama al método MostrarPersonasPorEdad. Mostrar personas con un nombre específico: Solicita un nombre y llama al método MostrarPersonasPorNombre. Termina la ejecución con return.

- Ejercicio #4 (Ejercicios de evaluación)

El código permite calcular el factorial de un número, mostrar el resultado y preguntar si se desea hacer otro cálculo o finalizar;

bool continuar = true;: Se utiliza una variable booleana llamada continuar que controla si el ciclo de repetición (bucle) debe continuar ejecutándose. Empieza en true.

while (continuar): Se utiliza un bucle while que se ejecutará mientras la variable continuar sea verdadera. Esto implica que el programa seguirá pidiendo números al usuario hasta que decida detenerse.

El programa solicita al usuario que ingrese un número. El número que el usuario ingresa es leído como una cadena y luego convertido a un entero utilizando int.Parse(). Si el número ingresado es negativo, se muestra un mensaje indicando que no se puede calcular el factorial de números negativos. Si el número no es negativo, se calcula el factorial.

int[] arregloNumeros = new int[1] { numero };: Se crea un arreglo de enteros arregloNumeros con un solo elemento, que es el número ingresado por el usuario. int[] arregloFactoriales = Factorial.calculaFactorial(arregloNumeros);: Se llama al método estático calculaFactorial de la clase Factorial, que toma el arreglo de números y devuelve un arreglo con los factoriales calculados.

Por último, el resultado se muestra al usuario mediante la función MostrarResultado, que imprime el arreglo de números y sus respectivos factoriales; Se le pregunta al usuario si desea continuar. Si la respuesta es "s", la variable continuar seguirá siendo true, lo que hará que el bucle se repita. Si la respuesta es otra, el bucle terminará.

Método calculaFactorial(int[] arregloNumeros): Este método toma un arreglo de enteros como parámetro. Luego, para cada número en el arreglo, calcula su factorial utilizando un bucle anidado.

El factorial se inicializa en 1 y se multiplica por cada valor desde 1 hasta el número en cuestión.

Los resultados se almacenan en arregloFactoriales y se retornan.

Método MostrarResultado(int[] arregloNumeros, int[] arregloFactoriales): Este método imprime el contenido del arreglo de números ingresados y sus factoriales correspondientes usando un bucle foreach.

- Ejercicio #1 (Guía didáctica)

Crea un programa que solicite al usuario la base y la altura de un triángulo. El programa debe incluir una función llamada calcular_area_triangulo que reciba como parámetros la base y la altura, y devuelva el área del triángulo. La fórmula para calcular el área es: Área =(base * altura) /2

Explicación:

El programa interactúa con el usuario y le pide la base y la altura de un triángulo, para luego calcular el área utilizando una función matemática y finalmente muestra el resultado en la pantalla.

Primeramente, el programa imprime un mensaje en la consola indicando que permitirá calcular el área de un triángulo. Posteriormente, solicita al usuario que ingrese la base del triángulo y lee la entrada del usuario y la convierte a un tipo double. Este mismo proceso lo hace con la altura. Después, llama al método calcular_area_triangulo de la clase Calculodearea, pasando la base y la altura como argumentos. El método calcular_area_triangulo calcula el área del triángulo utilizando la fórmula (base * altura) / 2 y devuelve el resultado. Finalmente, se llama al método imprimir_area de la clase Calculodearea, pasando el área calculada como argumento y el método imprimir_area imprime el resultado en la consola.

- ## Ejercicio #5 (Ejercicios de evaluación)

Escribe el programa que tenga una función que reciba un arreglo de enteros y lo invierta (el primer elemento se convierte en el último, el segundo en el penúltimo, etc.). Muestra el arreglo original y el invertido. - No puedes utilizar métodos ya definidos en el lenguaje. - Implementa una función que determine en el arreglo invertido, cuántos valores impares existen y los imprima.

Explicación:

Primeramente, se define una clase interna llamada InvertirYcontar dentro del espacio de nombres ArregloEnteros. Esta clase contiene tres métodos estáticos. El método InvertirArreglo toma un arreglo de enteros como parámetro y devuelve un nuevo arreglo con los elementos en orden inverso, utilizando un bucle for para asignar los valores en posiciones invertidas. El método ImprimirArreglo recibe un arreglo de enteros y utiliza un bucle foreach para imprimir cada elemento del arreglo seguido de un espacio, terminando con una nueva línea. Finalmente, el método ContarImpares también recibe un arreglo de enteros y cuenta cuántos de sus elementos son impares, incrementando un contador cada vez que encuentra un número impar y devolviendo el valor del contador al final. Posteriormente, en el programa principal se imprime un mensaje pidiendo al usuario que ingrese los datos del arreglo. Después, convierte la cadena de entrada en un arreglo de enteros. Luego, imprime el arreglo original. Después, llama al método imprimir arreglo y le pasa el arreglo original. Luego, llama al metodo invertirarreglo de la clase invertir y contar para invertir el arreglo original y lo almacena en el array invertido. Finalmente, se muestra el arreglo invertido gracias al metodo imprimir arreglo establecido en la clase y Llama al método ContarImpares de la clase InvertirYcontar para contar los números impares en el arreglo invertido y almacena el resultado en la variable impares.