

Universidad Tecnológica Metropolitana

Maestra: Ruth Betsaida Martinez Dominguez

Alumno: Jesus Alejandro Cauich Sosa

4 – B

Desarrollo de Software Multiplataforma

Estructura de Datos Aplicados

Actividad 11, 12 y 13

Matrices

Documentación de la Practica Numero 11

Introducción:

Este código C# es una aplicación de Windows Forms que genera una matriz 2D de 5x10 elementos con números aleatorios únicos y luego calcula la suma y el promedio de cada fila y columna de la matriz. Los resultados se muestran en la interfaz gráfica.

Desarrollo:

El código comienza por declarar variables de clase para la matriz y los arreglos que almacenarán la suma y el promedio de filas y columnas. Luego, en el constructor `Form1()`, se inicializan estas variables.

El evento `button1\_Click` se activa cuando se hace clic en un botón en la interfaz. En este evento, se genera la matriz aleatoria asegurándose de que los números sean únicos utilizando la función `NumeroRepetidoEnMatriz`. Luego, se calculan la suma y el promedio de cada fila y columna.

Finalmente, los resultados se muestran en etiquetas en la interfaz gráfica. La matriz se muestra en `label1`, la suma y el promedio de filas se muestran en `label2`, y la suma y el promedio de columnas se muestran en `label3`.

La función `NumeroRepetidoEnMatriz` se utiliza para verificar si un número ya existe en la matriz, evitando repeticiones.

Conclusión:

Este código es un ejemplo simple de cómo generar y procesar una matriz 2D en C# utilizando Windows Forms. Proporciona una forma de calcular la suma y el promedio de filas y columnas de la matriz y muestra los resultados en la interfaz gráfica. Este tipo de aplicaciones puede ser útil en diversos escenarios, como estadísticas o análisis de datos.

Desarrollo de la practica 12

Introducción:

Este código C# es una aplicación de Windows Forms que genera una tabla de ventas con números aleatorios y luego calcula diversas estadísticas sobre las ventas, como la venta total, la menor venta, la mayor venta y las ventas por día de la semana. Los resultados se muestran en la interfaz gráfica.

Desarrollo:

El código comienza por definir una clase llamada `Venta`, que se utiliza para realizar los cálculos de las estadísticas de ventas.

En la clase `forms`, se inicializa la interfaz de usuario en el constructor `forms()`. Se ocultan algunos campos de texto que se mostrarán más tarde cuando se realicen los cálculos.

El evento `button1\_Click` se activa cuando se hace clic en el botón "Generar Ventas". En este evento, se crea una tabla de ventas con 12 filas (representando semanas) y 7 columnas (representando días de la semana). Se generan números aleatorios únicos y se muestran en la tabla de ventas en la interfaz gráfica.

El evento `button2\_Click` se activa cuando se hace clic en el botón "Calcular Estadísticas". En este evento, se crea una instancia de la clase `Venta` y se llama al método `CalcularVentas` para calcular las estadísticas de ventas. Luego, se muestran las estadísticas en los campos de texto de la interfaz gráfica.

La clase `Venta` contiene el método `CalcularVentas`, que recorre la tabla de ventas y realiza los cálculos necesarios para obtener la menor venta, la mayor venta, la venta total y las ventas por día de la semana. Estos resultados se actualizan en los campos de texto de la interfaz gráfica.

Conclusión:

Este código es un ejemplo de una aplicación de Windows Forms en C# que genera y procesa datos de ventas. Proporciona una forma de calcular estadísticas importantes relacionadas con las ventas y mostrarlas en la interfaz de usuario. Este tipo de aplicaciones puede ser útil en diversas aplicaciones comerciales y financieras para el análisis de datos.

Desarrollo de la practica 13

Introducción:

Este código en C# es una aplicación de Windows Forms que genera calificaciones aleatorias para un número específico de alumnos y parciales. Luego, calcula varias estadísticas sobre estas calificaciones, como promedios de alumnos, promedio máximo, promedio mínimo, cantidad de reprobados y distribución de calificaciones.

Desarrollo:

- En la clase `Form1`, se inicializa la interfaz de usuario y se manejan los eventos de los botones.

- El método `btnCalcular\_Click` se activa al hacer clic en el botón "Calcular". En este método, se crea una instancia de la clase `Calificaciones` con el número de alumnos y parciales ingresados en la interfaz de usuario. Luego, se generan calificaciones aleatorias, se calculan promedios por alumno, el promedio máximo, el promedio mínimo, la cantidad de reprobados y la distribución de calificaciones. Los resultados se muestran en un `Label` en la interfaz de usuario.

- La clase `Calificaciones` se encarga de gestionar las calificaciones y realizar los cálculos. Contiene métodos para generar calificaciones aleatorias, calcular promedios, obtener el promedio máximo, el promedio mínimo, contar reprobados y calcular la distribución de calificaciones.

- Se utiliza un `HashSet` llamado `calificacionesGeneradas` para asegurarse de que las calificaciones generadas sean únicas.

- Las calificaciones aleatorias se generan entre 5.0 y 10.0 con un decimal.

- Los promedios se calculan correctamente dividiendo la suma de las calificaciones entre el número de parciales.

- La distribución de calificaciones se calcula en base a los valores enteros de los promedios.

Conclusión:

Este código es un ejemplo de una aplicación de Windows Forms en C# que genera y procesa calificaciones de estudiantes, calculando estadísticas útiles. Puede ser utilizado para análisis académicos y proporciona una forma efectiva de calcular y presentar información relacionada con las calificaciones de los alumnos.