

Universidad Tecnológica Metropolitana

Maestra: Ruth Betsaida Martinez Dominguez

Alumno: Jesus Alejandro Cauich Sosa

4 – B

Desarrollo de Software Multiplataforma

Estructura de Datos Aplicados

Actividad 3 y 4

Actividad 3

Declaración de la Clase Alumno:

La clase Alumno es una clase interna (internal), lo que significa que solo es accesible dentro del mismo ensamblado (assembly).

Propiedades:

La clase Alumno tiene dos propiedades:

Nombre: Representa el nombre del alumno.

Calificacion: Representa la calificación del alumno.

Nota: Las propiedades no son accesibles desde fuera de la clase porque no se especifica un modificador de acceso, por lo que, de manera predeterminada, son private. Para hacer que estas propiedades sean accesibles desde fuera de la clase, debes especificar public antes de string y double en las declaraciones de propiedades.

Constructor de Alumno:

El constructor Alumno(string nombre, double calificacion) se utiliza para crear una instancia de la clase Alumno y establece las propiedades Nombre y Calificacion con los valores proporcionados como argumentos. Método ToString:

Se ha sobrescrito el método ToString en la clase Alumno. El método ToString se utiliza para proporcionar una representación de cadena de un objeto.

En este caso, el método ToString devuelve una cadena que combina el nombre y la calificación del alumno en el formato "Nombre: Calificacion".A screen shot of a computer program

Description automatically generated

Listas de Alumnos:

Se definen dos listas de objetos Alumno:

Aprobados: Esta lista almacenará a los alumnos que han aprobado.

Reprobados: Esta lista almacenará a los alumnos que han reprobado.

Constructor de Form1:

El constructor Form1 inicializa la ventana del formulario llamando a InitializeComponent() y configura las listas de Aprobados y Reprobados.

Evento btnAgregar\_Click:

Este evento se dispara cuando se hace clic en un botón llamado btnAgregar.

Obtención de Datos:

Se obtiene el nombre ingresado en txtNombre (un cuadro de texto) y se almacena en la variable nombre.

Se verifica si el cuadro de texto txtCalificacion contiene una calificación válida (un número decimal). Si es válido, se almacena en la variable calificacion.

Creación del Objeto Alumno:

Se crea un objeto Alumno llamado alumno utilizando el nombre y la calificación proporcionados.

Clasificación de Alumnos:

Se verifica si la calificación del alumno es mayor o igual a 7. Si es así, se agrega a la lista de Aprobados y se muestra en el lstAprobados (un control de lista).

Si la calificación es menor a 7, el alumno se agrega a la lista de Reprobados y se muestra en el lstReprobados.

Manejo de Errores:

Se muestran mensajes de error si el nombre ingresado es nulo o vacío, o si la calificación no es un número válido.

Limpiar Cuadros de Texto:

Después de procesar los datos, los cuadros de texto se borran para permitir la entrada de nuevos datos. A screen shot of a computer program

Description automatically generatedA computer screen with colorful text

Description automatically generated

En resumen, este código es parte de una aplicación de Windows Forms que permite al usuario ingresar información de estudiantes (nombre y calificación) y clasificarlos como "Aprobados" o "Reprobados" según su calificación. Los alumnos se almacenan en las listas Aprobados o Reprobados y se muestran en listas visibles en la interfaz de usuario.

Actividad 4

Declaración de la Clase Producto:

La clase Producto es una clase interna (internal), lo que significa que solo es accesible desde dentro del mismo ensamblado (assembly).

Propiedades de la Clase:

La clase Producto tiene tres propiedades públicas que representan las características del producto:

Nombre: Representa el nombre del producto.

Precio: Representa el precio del producto, generalmente se utiliza el tipo de dato decimal para representar valores monetarios debido a su precisión.

ID: Representa el identificador único del producto, que generalmente es un número entero (tipo int).

Propiedades Auto-implementadas:

Las propiedades en esta clase están utilizando una sintaxis de propiedad auto-implementada. Esto significa que no hay lógica adicional en los métodos get y set, y el valor de la propiedad se almacena automáticamente en el fondo. A computer screen shot of a program code

Description automatically generated

Campos y Propiedades:

List<Producto> listaProductos: Una lista que almacena objetos de tipo Producto, representando la lista de productos en la aplicación.

Producto producto: Un objeto de tipo Producto que parece no ser utilizado directamente en el código.

Constructor de Form1:

El constructor Form1 inicializa la ventana del formulario llamando a InitializeComponent() y configura las listas de productos.

Evento btnAgregar\_Click:

Este evento se dispara cuando se hace clic en un botón llamado btnAgregar.

Agregar Producto:

Se genera un ID aleatorio entre 1 y 100 para el nuevo producto.

Se obtiene el nombre y el precio ingresados en los cuadros de texto (txtNombre y txtPrecio).

Se verifica si el precio es un número decimal válido. Si es válido, se crea un nuevo objeto Producto y se agrega a la lista listaProductos.

Luego, se llama al método ActualizarListaProductos() para actualizar la interfaz de usuario y se limpian los cuadros de texto.

Evento btnEliminar\_Click:

Este evento se dispara cuando se hace clic en un botón llamado btnEliminar.

Eliminar Producto:

Se obtiene la clave de eliminación ingresada en el cuadro de texto txtClaveEliminar.

Se utiliza el método RemoveAll para eliminar productos de la lista listaProductos cuyo ID coincida con la clave de eliminación ingresada.

Luego, se llama al método ActualizarListaProductos() para actualizar la interfaz de usuario y se limpia el cuadro de texto.

Evento btnOrdenar\_Click:

Este evento se dispara cuando se hace clic en un botón llamado btnOrdenar.

Ordenar Productos:

Se ordena la lista de productos (listaProductos) en orden alfabético basado en el nombre del producto utilizando el método OrderBy.

Método ActualizarListaProductos:

Este método se encarga de actualizar el contenido del control listBoxProductos con los productos actuales, calcular el costo total de los productos y mostrarlo en labelCostoTotal.

Interfaz de Usuario:

En la interfaz, se utilizan controles como botones, cuadros de texto, listas y etiquetas para interactuar con el usuario y mostrar información.

A computer screen shot of a program code

Description automatically generatedA screen shot of a computer program

Description automatically generatedEn resumen, esta aplicación de Windows Forms permite al usuario administrar una lista de productos, agregar nuevos productos, eliminar productos por clave y ordenar los productos por nombre. La información de los productos se muestra en una lista, y el costo total de los productos se calcula y se muestra en la interfaz de usuario.