Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente namespace clase1

{

    public class PedidoPostre

    {

        public string nombre { get; set; }

        public int precio { get; set; }

        public int cantidadPedido { get; set; }

    }

    public class PedidoBebida

    {

        public string nombre { get; set; }

        public int cantidad { get; set; }

        public decimal precio { get; set; }

        public string marca { get; set; }

        public string tamaño { get; set; }

    }

    public class Pedidoropa

    {

        public string talla { get; set; }

        public string nombre { get; set; }

        public decimal precio { get; set; }

        public int cantidad { get; set; }

        public string marca { get; set; }

        public string color { get; set; }

        public string tiporopa { get; set; }

    }

    //creo un interfaz cola elegi una cola ya que los pedidos el primero en la cola sera el primero en salir tomando ese principio elegi la solucion de una cola

    public interface Icola<T>

    {

        void agregarcola(T data);

        bool estanull();

        T delete();

        T search(Predicate<T> filtrobusqueda);

    }

    public class Nodo<T>

    {

        public T Data { get; set; }

        public Nodo<T> Siguiente { get; set; }

        public Nodo(T data)

        {

            Data = data;

            Siguiente = null;

        }

    }

    public class Cola<T>: Icola<T>

    {

        private Nodo<T> firstnodo;

        private Nodo<T> lastnodo;

        public T search(Predicate<T> filtrobusqueda)

        {

            Nodo<T> actual = firstnodo;

            while (actual != null)

            {

                if (filtrobusqueda(actual.Data))

                {

                    return actual.Data;

                }

                actual = actual.Siguiente;

            }

            return default(T);

        }

        public void agregarcola(T data)

        {

            Nodo<T> newnodo = new Nodo<T>(data);

            if (estanull())

            {

                firstnodo = newnodo;

                lastnodo = newnodo;

            }

            else

            {

                lastnodo.Siguiente = newnodo;

                lastnodo = newnodo;

            }

        }

        public T delete()

        {

            if (estanull())

            {

                throw new InvalidOperationException("la cola se encuentra vacia");

            }

            T datadelete = firstnodo.Data;

            if(firstnodo == lastnodo)

            {

                firstnodo = null;

                lastnodo=null;

            }

            else

            {

                firstnodo = firstnodo.Siguiente;

            }

            return datadelete;

        }

        public bool estanull()

        {

            return firstnodo == null;

        }

    }

    public class PedidoPostres

    {

        private Cola<PedidoPostres> colapedidospostres = new Cola<PedidoPostres>();

        public void agregarpedido(PedidoPostres pedido)

        {

            colapedidospostres.agregarcola(pedido);

        }

        public PedidoPostre atenderpedido()

        {

            return colapedidospostres.delete();

        }

        public PedidoPostre buscarpedidoporcantidad(string Nombre)

        {

            return colapedidospostres.search(x => x.nombre == Nomsbre)

        }

    }

    public class PedidoBebidas

    {

        private Cola<PedidoBebidas> colapedidobebida = new Cola<PedidoBebidas> ();

        public void agregarpedido(PedidoBebidas pedido)

        {

            colapedidobebida.agregarcola(pedido);

        }

        public PedidoBebida atenderpedido()

        {

            return colapedidobebida.delete();

        }

        public PedidoPostre buscarpedidopornombre(string Cantidad)

        {

            return colapedidobebida.search(x => x.cantidad == Cantidad)

        }

    }

    public class PedidoRopas

    {

        private Cola<Pedidoropas> colapedidoropa = new Cola<Pedidoropas> ();

        public void agregarpedido(Pedidoropas pedido)

        {

            colapedidoropa.agregarcola(pedido);

        }

        public Pedidoropa atenderpedido()

        {

            return colapedidoropa.delete();

        }

        public Pedidoropa buscarpedidoportalla(string Talla)

        {

            return colapedidoropa.search(x => x.talla == Talla)

        }

    }

}

SEGUNDO PARCIAL FUE PRESENTACION DE ARBOLES AVL Y COLAS DE PRIORIDAD EN GRUPOS