```
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-06-17
 * Assignment:  cView-P3
 * Source File: PublicFacility.cs
 * Language:    C#
 * Course:      CSCI-C 490, C# Programming, MoWe 08:00
 * Purpose:     Contains the PublicFacility class, derived from the Item abstract class, and
 *              supporting methods.
--------------------------------------------------------------------------------------------------*/

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ph3
{
    public class PublicFacility : Item
    {
        /*------------------------------------------------------------------------------------------
         * Type:    Helper Constants
         * Purpose:
        ------------------------------------------------------------------------------------------*/
        public new const FieldMenuHelper FieldMin = FieldMenuHelper.Name;
        public new const FieldMenuHelper FieldMax = FieldMenuHelper.Back;
        public const int FieldOffset = 0;

        /*------------------------------------------------------------------------------------------
         * Type:    Private Fields
        ------------------------------------------------------------------------------------------*/
        private string itemType = "publicfacility";

        /*------------------------------------------------------------------------------------------
         * Type:    Constructor
         * Purpose: Basic no-parameter constructor.
         * Input:   Nothing.
        ------------------------------------------------------------------------------------------*/
        public PublicFacility()
        {
            // Nothing else to do.
        }

        /*------------------------------------------------------------------------------------------
         * Type:    Constructor
         * Purpose: Copy constructor.
         * Input:   PublicFacility fromItem, reference to the other PublicFacility from which fields
         *          should be copied.
        ------------------------------------------------------------------------------------------*/
        public PublicFacility(PublicFacility fromItem)
            : base(fromItem)
        {
            itemType = fromItem.itemType;
        }

        /*------------------------------------------------------------------------------------------
         * Type:    Constructor
         * Purpose: Constructor that will fill all the properties except ItemID and ItemType.
         * Input:   string name, contains the desired Name for the object.
         * Input:   string type, contains the desired Type for the object.
         * Input:   string streetAddress, contains the desired StreetAddress for the object.
         * Input:   string city, contains the desired City for the object.
         * Input:   string state, contains the desired State for the object.
         * Input:   string zip, contains the desired Zip for the object.
         * Input:   string latitude, contains the desired Latitude for the object.
         * Input:   string longitude, contains the desired Longitude for the object.
         * Input:   string phone, contains the desired Phone for the object.
         * Output:  Nothing.
        ------------------------------------------------------------------------------------------*/
        public PublicFacility(string name, string type, string streetAddress, string city, string state,
            string zip, string latitude, string longitude, string phone)
            : base(name, type, streetAddress, city, state, zip, latitude, longitude, phone)
        {
            // Nothing else to do.
        }

        /*------------------------------------------------------------------------------------------
         * Type:    Auto-implemented Properties
        ------------------------------------------------------------------------------------------*/
        public override int ItemID { get; set; }            // Item ID

        public override string Name { get; set; }           // Facility Name
        public override string Type { get; set; }           // Facility Type
        public override string StreetAddress { get; set; }  // Street Address
        public override string City { get; set; }           // City
        public override string State { get; set; }          // State
        public override string Zip { get; set; }            // Zip Code
        public override string Latitude { get; set; }       // Latitude
        public override string Longitude { get; set; }      // Longitude
        public override string Phone { get; set; }          // Phone Number

        /*------------------------------------------------------------------------------------------
         * Name:    ItemType
         * Type:    Property
         * Purpose: Provides access to the itemType field.
        ------------------------------------------------------------------------------------------*/
        public override string ItemType
        {
            get
            {
                return itemType;
            }
            set
            {
                // Do nothing.
            }
        }

        /*------------------------------------------------------------------------------------------
         * Name:    ToString
         * Type:    Method
         * Purpose: Override of ToString() method. Formats the data contained in this object so it
         *          looks pretty.
         * Input:   Nothing.
         * Output:  string, containing serialized object data.
        ------------------------------------------------------------------------------------------*/
        public override string ToString()
        {
            // Returns a the base a string formatted as follows:
            //  Item ID (Item Type): <ItemID> (<ItemType>)
            // Facility Name (Type): <Name> (<Type>)
            //              Address: <StreetAddress>, <City>, <State> <Zip>
            //      GPS Coordinates: (<Latitude>, <Longitude>)
            //         Phone Number: <Phone>
            return string.Format(
                " Item ID (Item Type): {0} ({1})\n" +
                "Facility Name (Type): {2} ({3})\n" +
                "             Address: {4}, {5}, {6} {7}\n" +
                "     GPS Coordinates: ({8}, {9})\n" +
                "        Phone Number: {10}",
                ItemID, ItemType,
                Name, Type,
                StreetAddress, City, State, Zip,
                Latitude, Longitude,
                Phone);
        }

    }
}

```