```
/*-----------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-06-17
 * Assignment:  cView-P3
 * Source File: Business.cs
 * Language:    C#
 * Course:      CSCI-C 490, C# Programming, MoWe 08:00
 * Purpose:     Contains the Business 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 Business : Item
    {
        /*-------------------------------------------------------------------------------------------
         * Type:    Helper Constants
        -------------------------------------------------------------------------------------------*/
        public new const FieldMenuHelper FieldMin = FieldMenuHelper.LicenseFiscalYear;
        public new const FieldMenuHelper FieldMax = FieldMenuHelper.BackBusiness;
        public const int FieldOffset = 1;

        /*-------------------------------------------------------------------------------------------
         * Type:    Private Fields
        -------------------------------------------------------------------------------------------*/
        private string itemType = "business";
        private int licenseFiscalYear;
        private int licenseNumber;

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

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

            LicenseFiscalYear = fromItem.LicenseFiscalYear;
            LicenseNumber = fromItem.LicenseNumber;
            LicenseIssueDate = fromItem.LicenseIssueDate;
            LicenseExpirDate = fromItem.LicenseExpirDate;
            LicenseStatus = fromItem.LicenseStatus;
            CouncilDistrict = fromItem.CouncilDistrict;
        }

        /*-------------------------------------------------------------------------------------------
         * Type:    Constructor
```

```
67          * Purpose: Constructor that will fill all the properties except ItemID and ItemType.
68          * Input:   string name, contains the desired Name for the object.
69          * Input:   string type, contains the desired Type for the object.
70          * Input:   string streetAddress, contains the desired StreetAddress for the object.
71          * Input:   string city, contains the desired City for the object.
72          * Input:   string state, contains the desired State for the object.
73          * Input:   string zip, contains the desired Zip for the object.
74          * Input:   string latitude, contains the desired Latitude for the object.
75          * Input:   string longitude, contains the desired Longitude for the object.
76          * Input:   string phone, contains the desired Phone for the object.
77          * Input:   int licenseFiscalYear, contains the desired LicenseFiscalYear for the object.
78          * Input:   int licenseNumber, contains the desired LicenseNumber for the object.
79          * Input:   DateTime licenseIssueDate, contains the desired LicenseIssueDate for the object.
80          * Input:   DateTime licenseExpirDate, contains the desired LicenseExpirDate for the object.
81          * Input:   string licenseStatus, contains the desired LicenseStatus for the object.
82          * Input:   string councilDistrict, contains the desired CouncilDistrict for the object.
83          * Output:  Nothing.
84         ----------------------------------------------------------------------------------------*/
85         public Business(string name, string type, string streetAddress, string city, string state,
86             string zip, string latitude, string longitude, string phone, int licenseFiscalYear,
87             int licenseNumber, DateTime licenseIssueDate, DateTime licenseExpirDate,
88             string licenseStatus, string councilDistrict)
89             : base(name, type, streetAddress, city, state, zip, latitude, longitude, phone)
90         {
91             LicenseFiscalYear = licenseFiscalYear;
92             LicenseNumber = licenseNumber;
93             LicenseIssueDate = licenseIssueDate;
94             LicenseExpirDate = licenseExpirDate;
95             LicenseStatus = licenseStatus;
96             CouncilDistrict = councilDistrict;
97         }
98
99         /*----------------------------------------------------------------------------------------
100         * Type:    Auto-implemented Properties
101        ----------------------------------------------------------------------------------------*/
102        public override int ItemID { get; set; }            // Item ID
103
104        public override string Name { get; set; }           // Business Name
105        public override string Type { get; set; }           // Business Classification
106        public override string StreetAddress { get; set; }  // Street Address
107        public override string City { get; set; }           // City
108        public override string State { get; set; }          // State
109        public override string Zip { get; set; }            // Zip Code
110        public override string Latitude { get; set; }       // Latitude
111        public override string Longitude { get; set; }      // Longitude
112        public override string Phone { get; set; }          // Phone Number
113
114        public DateTime LicenseIssueDate { get; set; }      // Business License Issue Date
115        public DateTime LicenseExpirDate { get; set; }      // Business License Expiration Date
116        public string LicenseStatus { get; set; }           // Business License Status
117        public string CouncilDistrict { get; set; }         // Council District
118
119        /*----------------------------------------------------------------------------------------
120         * Name:    ItemType
121         * Type:    Property
122         * Purpose: Provides access to the itemType field.
123        ----------------------------------------------------------------------------------------*/
124        public override string ItemType
125        {
126            get
127            {
128                return itemType;
129            }
130            set
131            {
132                // Do nothing.
```

```
133             }
134         }
135 
136         /*-----------------------------------------------------------------------------------------
137          * Name:    LicenseFiscalYear
138          * Type:    Property
139          * Purpose: Provides access to the licenseFiscalYear field, and validation for the same.
140         -----------------------------------------------------------------------------------------*/
141         public int LicenseFiscalYear
142         {
143             get
144             {
145                 return licenseFiscalYear;
146             }
147             set
148             {
149                 if (value >= 0)
150                     licenseFiscalYear = value;
151             }
152         }
153 
154         /*-----------------------------------------------------------------------------------------
155          * Name:    LicenseNumber
156          * Type:    Property
157          * Purpose: Provides access to the licenseNumber field, and validation for the same.
158         -----------------------------------------------------------------------------------------*/
159         public int LicenseNumber
160         {
161             get
162             {
163                 return licenseNumber;
164             }
165             set
166             {
167                 if (value >= 0)
168                     licenseNumber = value;
169             }
170         }
171 
172         /*-----------------------------------------------------------------------------------------
173          * Name:    this[]
174          * Type:    Indexer
175          * Purpose: Provides easy access to the properties of the class.
176          * Input:   FieldMenuHelper fiendNum, represents the desired property.
177          * Output:  object, contains whichever property was desired, or 0 if the property was not
178          *          found.
179         -----------------------------------------------------------------------------------------*/
180         public override object this[FieldMenuHelper fieldNum]
181         {
182             get
183             {
184                 switch (fieldNum)
185                 {
186                     case FieldMenuHelper.LicenseFiscalYear:
187                         return LicenseFiscalYear;
188                     case FieldMenuHelper.LicenseNumber:
189                         return LicenseNumber;
190                     case FieldMenuHelper.LicenseIssueDate:
191                         return LicenseIssueDate;
192                     case FieldMenuHelper.LicenseExpirDate:
193                         return LicenseExpirDate;
194                     case FieldMenuHelper.LicenseStatus:
195                         return LicenseStatus;
196                     case FieldMenuHelper.CouncilDistrict:
197                         return CouncilDistrict;
198                     default:
```

```
199                         return base[fieldNum];
200                 }
201             }
202         }
203 
204         /*---------------------------------------------------------------------------------------
205          * Name:    ToStringCSV
206          * Type:    Method
207          * Purpose: Serializes the data contained in the object into a comma-separated value string.
208          * Input:   Nothing.
209          * Output:  string, representing the data of this object as serialized to a CSV string.
210         ---------------------------------------------------------------------------------------*/
211         public override string ToStringCSV()
212         {
213             char separator = ',';
214             return base.ToStringCSV() + separator + LicenseFiscalYear + '-' + LicenseNumber +
215                 separator + LicenseIssueDate.ToShortDateString() + separator +
216                 LicenseExpirDate.ToShortDateString() + separator + LicenseStatus + separator +
217                 CouncilDistrict;
218         }
219 
220         /*---------------------------------------------------------------------------------------
221          * Name:    ToString
222          * Type:    Method
223          * Purpose: Override of ToString() method. Formats the data contained in this object so it
224          *          looks pretty.
225          * Input:   Nothing.
226          * Output:  string, containing serialized object data.
227         ---------------------------------------------------------------------------------------*/
228         public override string ToString()
229         {
230             // Returns a string formatted as follows:
231             //  Item ID (Item Type): <ItemID> (<ItemType>)
232             // Business Name (Type): <Name> (<Type>)
233             //              Address: <StreetAddress>, <City>, <State> <Zip>
234             //      GPS Coordinates: (<Latitude>, <Longitude>)
235             //         Phone Number: <Phone>
236             //       License Number: <LicenseFiscalYear>-<LicenseNumber>
237             //                Valid: From <LicenseIssueDate> to <LicenseExpirDate>
238             //               Status: <LicenseStatus>
239             //     Council District: <CouncilDistrict>
240             return string.Format(
241                 " Item ID (Item Type): {0} ({1})\n" +
242                 "Business Name (Type): {2} ({3})\n" +
243                 "             Address: {4}, {5}, {6} {7}\n" +
244                 "     GPS Coordinates: ({8}, {9})\n" +
245                 "        Phone Number: {10}\n" +
246                 "      License Number: {11}-{12}\n" +
247                 "               Valid: From {13} to {14}\n" +
248                 "              Status: {15}\n" +
249                 "    Council District: {16}",
250                 ItemID, ItemType,
251                 Name, Type,
252                 StreetAddress, City, State, Zip,
253                 Latitude, Longitude,
254                 Phone,
255                 LicenseFiscalYear, LicenseNumber,
256                 LicenseIssueDate.ToShortDateString(), LicenseExpirDate.ToShortDateString(),
257                 LicenseStatus,
258                 CouncilDistrict);
259         }
260     }
261 }
262 
```