```
/*-------------------------------------------------------------------------------------------
 * Name:        Dan Cassidy
 * Date:        2015-06-02
 * Assignment:  cView-P1
 * Source File: CViewDataSet.cs
 * Class:       CSCI-C 490, C# Programming, MoWe 08:00
 * Purpose:     Builds a List-based class for collections of CViewData objects and contains related
 *              methods and properties.
-------------------------------------------------------------------------------------------*/

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

namespace cView_P1_DanCassidy
{
    class CViewDataSet
    {
        //Basic property of the class.
        private List<CViewData> dataSet = new List<CViewData>();

        //Enable read-only access to the Count property.
        public int Count
        {
            get
            {
                return dataSet.Count;
            }
        }

        //Enable read-only access to the Header property. Uses the header from the CViewData class
        //so if needs to be changed, it only needs to be changed in one place.
        public string Header
        {
            get
            {
                return CViewData.Header;
            }
        }

        /*-------------------------------------------------------------------------------------
         * Method:  this[]
         * Purpose: Access the objects in this dataset via index number.
         * Input:   int objectNum, the index of the object that will be accessed.
         * Output:  CViewData object of the referenced object at the index.
        -------------------------------------------------------------------------------------*/
        public CViewData this[int objectNum]
        {
            get
            {
                //Try to simply return the object at index objectNum.
                try
                {
                    return dataSet[objectNum];
                }
                catch (ArgumentOutOfRangeException)
                {
                    //If this exception is caught, let the user know and return a null.
                    Console.WriteLine("Index [{0}] is out of range.", objectNum);
                    return null;
                }
            }
            set
            {
```

```
67                //Try to set the object at index objectNum.
68                try
69                {
70                    dataSet[objectNum] = value;
71                }
72                catch (ArgumentOutOfRangeException)
73                {
74                    //If this exception is caught, do nothing further and let the user know.
75                    Console.WriteLine("Index [{0}] is out of range.", objectNum);
76                }
77            }
78        }
79
80        /*-------------------------------------------------------------------------------------------
81         * Method:  Add
82         * Purpose: Add a data object to the dataset.
83         * Input:   CViewData toAdd, this is the object that will get added to the dataset.
84         * Output:  Nothing.
85        -------------------------------------------------------------------------------------------*/
86        public void Add(CViewData toAdd)
87        {
88            //Add object using List Add method.
89            dataSet.Add(toAdd);
90        }
91
92        /*-------------------------------------------------------------------------------------------
93         * Method:  Delete
94         * Purpose: Delete an object at the given index from the dataset.
95         * Input:   int indexToRemove, the index of the object to be removed from the dataset.
96         * Output:  Nothing.
97        -------------------------------------------------------------------------------------------*/
98        public void Delete(int indexToRemove)
99        {
100           //Delete object at specified index by using List RemoveAt method.
101           dataSet.RemoveAt(indexToRemove);
102       }
103
104       /*-------------------------------------------------------------------------------------------
105        * Method:  SortByName
106        * Purpose: Sort the dataset by the Name property of the objects, with a secondary sort by
107        *          the Address property.
108        * Input:   Nothing.
109        * Output:  Nothing.
110       -------------------------------------------------------------------------------------------*/
111       public void SortByName()
112       {
113           //Idea from Stack Overflow: http://stackoverflow.com/a/3309230
114           //Yay lambda expressions!
115           dataSet = dataSet.OrderBy(data => data.Name).OrderBy(data => data.Address).ToList();
116       }
117
118       /*-------------------------------------------------------------------------------------------
119        * Method:  Search
120        * Purpose: Search for a given string in this dataset.
121        * Input:   string toSearchFor, this is the string that will be searched for.
122        * Output:  CViewDataSet object, containing all (if any) objects found.
123       -------------------------------------------------------------------------------------------*/
124       public CViewDataSet Search(string toSearchFor)
125       {
126           //Create a new dataset to hold the found objects.
127           CViewDataSet foundDataSet = new CViewDataSet();
128
129           //Iterate through the objects and add them to foundDataSet if applicable.
130           foreach (CViewData data in dataSet)
131               if (data.Contains(toSearchFor))
132                   foundDataSet.Add(data);
```

```
133
134             //Return the dataset containing the found objects.
135             return foundDataSet;
136         }
137
138         /*-------------------------------------------------------------------------------------------
139          * Method:  ToString
140          * Purpose: Override of the ToString() method. Formats the return value so it looks pretty.
141          * Input:   Nothing.
142          * Output:  String object containing serialized collection data.
143         -------------------------------------------------------------------------------------------*/
144         public override string ToString()
145         {
146             //Declare the string.
147             string toReturn = "";
148
149             //Build the string.
150             foreach (CViewData item in dataSet)
151                 toReturn += item.ToString() + "\n";
152
153             //Return the string.
154             return toReturn;
155         }
156     }
157 }
158
```