

```

// Program 1A
// CIS-200-01
// Fall 2017
// Due: 9/25/2017
// By: C5503

// File: AddressSelectionForm.cs
// The AddressSelectionForm is a form that allows you to select an address to edit based
on the given names that are stored.

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Prog2
{
    public partial class AddressSelectionForm : Form
    {
        private List<Address> addressList; // List of addresses used to fill combo boxes
        ErrorProvider errorProvider1 = new ErrorProvider(); // Error provider created to
use in validation

        // Precondition: addresses.Count >= MIN_ADDRESSES
        // Postcondition: The form's GUI is prepared for display.
        public AddressSelectionForm(List<Address> addresses)
        {
            InitializeComponent();
            addressList = addresses;
        }

        public int addressIndex
        {
            // Precondition: User has selected from addressCbo1
            // Postcondition: The index of the selected origin address returned
            get
            {
                return addressCbo1.SelectedIndex;
            }
        }

        // Precondition: addressList.Count >= MIN_ADDRESSES
        // Postcondition: The list of addresses is used to populate the
        // addressCbo1 combobox
        private void AddressSelection_Load(object sender, EventArgs e)
        {
            foreach (Address a in addressList)
                addressCbo1.Items.Add(a.Name);
        }

        // Precondition: Focus shifting from one of the address combo boxes
        // sender is ComboBox
        // Postcondition: If no address selected, focus remains and error provider
        // highlights the field
    }
}

```

```

private void addressCbo_Validating(object sender, CancelEventArgs e)
{
    if(addressCbo1.SelectedIndex == -1)
    {
        e.Cancel = true;
        errorProvider1.SetError(addressCbo1, "Select an Address");
    }
}
// Precondition: Validating of sender not cancelled, so data OK
// sender is Control
// Postcondition: Error provider cleared and focus allowed to change
private void addressCbo_Validated(object sender, EventArgs e)
{
    errorProvider1.SetError(addressCbo1, "");
}
// Precondition: User pressed on cancelBtn
// Postcondition: Form closes
private void btnExit_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
}
// Precondition: User clicked on okBtn
// Postcondition: If invalid field on dialog, keep form open and give first
invalid // field the focus. Else return OK and close form.
private void btnOk_Click(object sender, EventArgs e)
{
    if (ValidateChildren())
        this.DialogResult = DialogResult.OK;
}
}
}

```