namespace Membership\_Maintenance\_Form

{

/// <summary>

/// a list class to store a member object

/// </summary>

class MembershipList

{

private List<Member> members; //list attribute to store

/// <summary>

/// delegate a declaration and

/// declare the event

/// </summary>

/// <param name="members"></param>

public delegate void ChangeHandler(MembershipList members);

public event ChangeHandler Changed;

.

.

.

//indexer properties

public Member this[int i]

{

get

{

return members[i];

}

set

{

members[i] = value;

Changed(this);

}

}

/// <summary>

/// each new Member object is added to the indexer

/// </summary>

/// <param name="i"></param>

/// <returns></returns>

public Member GetMemberByIndex(int i) => members[i];

/// <summary>

/// Fill the list with membership data from the Membership data

/// from a file using the GetMemberships method of the

/// MembershipData class

/// </summary>

public void Fill() => MembershipData.GetMemberships(this);

/// <summary>

/// Saves the memberships to a file using the SaveMemberships

/// method of the MembershipData class

/// </summary>

public void Save() => MembershipData.SaveMemberships(members);

//Adds a specified Member object to the list

public void Add(Member m)

{

members.Add(m);

Changed(this);

}

//Removes the specified Member object from the list

public void Remove(Member m)

{

members.Remove(m);

Changed(this);

}

/// <summary>

/// overloading operators

/// </summary>

/// <param name="m1"></param>

/// <param name="m2"></param>

/// <returns></returns>

//overload + with add method

public static MembershipList operator +(MembershipList m1, Member m2)

{

m1.Add(m2);

return m1;

}

//overload - with remove method

public static MembershipList operator -(MembershipList m1, Member m2)

{

m1.Remove(m2);

return m1;

}

}

}

namespace Membership\_Maintenance\_Form

{

public partial class MMForm : Form

{

.

.

.

private MembershipList members = new MembershipList();

private void MMForm\_Load(object sender, EventArgs e)

{

members.Changed += new MembershipList.ChangeHandler(HandleChange);

members.Fill();

FillMemberListBox();

}

private void FillMemberListBox()

{

Member m;

ListBoxLabel.Items.Clear();

for(int i = 0; i < members.Count; i++)

{

m = members[i];

ListBoxLabel.Items.Add(m.GetDisplayText());

}

}

private void HandleChange(MembershipList members)

{

members.Save();

FillMemberListBox();

}

private void AddButton\_Click(object sender, EventArgs e)

{

AddMForm AddMember = new AddMForm();

Member member = AddMember.GetNewMember();

if (member != null)

{

members += member;

}

}

private void DeleteButton\_Click(object sender, EventArgs e)

{

int MemberNum = ListBoxLabel.SelectedIndex;

if (MemberNum != -1)

{

Member member = members[MemberNum];

string message = "Are you sure you want to delete " + member.firstName + " " + member.lastName + "?";

DialogResult button = MessageBox.Show(message, "Confirm Delete", MessageBoxButtons.YesNo);

if (button == DialogResult.Yes)

{

members -= member;

}

}

}

}

}

namespace Membership\_Maintenance\_Form

{

public partial class AddMForm : Form

{

private Member member;

.

.

.

internal Member GetNewMember()

{

this.ShowDialog();

return member;

}

}

}