Лабораторная работа

**«Реализация интерфейса в классе»**

**Тема**: реализация интерфейса IEnumerable<T> на основе класса.

Структура работы:

1. в VS создать WindowsForm проект для C# и установить на форму панель (экземпляр класса Panel),
2. определить в модуле Form1.cs класс MiniList<T>, реализующий упрощённый список узлов, содержащих в виде поля объекты типа T,
3. создать на форме Form1 интерфейс для тестирования класса MiniList<T> при подстановке типа CheckBox вместо параметра T и провести тестирование,
4. разработать на основе класса MiniList<T> класс

MiniList<T: IEnumerable<T>

Реализующий интерфейсы:

**public interface IEnumerable<out T> : IEnumerable**

{

IEnumerator<T> GetEnumerator()

}

и

**public interface IEnumerator<out T> : IDisposable, IEnumerator**

{

T Current { get; }

void Dispose()

bool MoveNext()

void Reset()

}

1. тестировать реализацию универсальных интерфейсов IEnumerable<T> и IEnumerator<T>.
2. тестировать реализацию интерфейсов IEnumerable и IEnumerator.

Указания:

1.1 Установить на форме главное меню с командами:

**Создать список**

**Добавить элемент**

**Перебор кнопок**

**Перебор объектов**

1.2 Занести на форму экземпляр класса Panel (панель **panel1**) и установить в нём свойства:

AutoScroll = true;

1.3 Занести на форму экземпляр **statusStrip1** класса StatusStrip и с помощью мастера ввести в него метку **toolStripStatusLabel1.**

Очистить поле Text метки.

2.1 Определить в модуле Form1.cs класс

class MiniList<T>

{

static int num = 0;

public Node Top;

public class Node

{

public T content;

public Node Next = null;

}

public int Append(T s)

{

Node p = new Node();

p.content = s;

if (Top != null) p.Next = Top;

Top = p;

return num++;

}

}

Этот класс обеспечивает построение однонаправленного цепного списка узлов, содержащих данные произвольного типа, заданного параметром T.

3.1 Определить в классе Form1 поле

MiniList<CheckBox> v;

3.2 Вести в класс Form1 и подключить к пункту меню **Создать список** обработчик события Click:

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

{

v = new MiniList<CheckBox>();

toolStripStatusLabel1.Text = "Список создан.";

}

3.3 Вести в класс Form1 и подключить к пункту меню **Добавить элемент** обработчик события Click:

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

{

if (v != null)

{

CheckBox vcb = new CheckBox();

vcb.Appearance = Appearance.Button;

vcb.FlatStyle = FlatStyle.Standard;

vcb.Top = 50;

vcb.Width = 30;

vcb.Left = 40 \* v.Append(vcb);

vcb.Text = Convert.ToString(vcb.Left);

this.panel1.Controls.Add(vcb);

toolStripStatusLabel1.Text = "";

} else toolStripStatusLabel1.Text = "Список не создан.";

}

Обработчик создаёт кнопку, определяет её размеры и местоположение на панели и добавляет на панель. Смещение от левой границы панели определяется номером кнопки, который формируется при создании узла в методе Append.

3.5 Создать список командой меню **Создать список** и удостовериться, что повторением команды меню **Добавить элемент** можно ввести в список практически неограниченное количество узлов, содержимое которых (кнопки) отображается на панель.

4.1 Удостовериться, что к модулю Form1.cs подключены оба пространства имён

System.Collections и System.Collections.Generic.

и ввести в модуль Form1.cs класс (потомок класса Exception):

class ListIsEmpty : InvalidOperationException { }

4.2 Вести в класс MiniList<T> класс Numerator:

public class Numerator: IEnumerator<T>

{

bool active = false;

MiniList<T> lst;

MiniList<T>.Node current = null;

public Numerator(MiniList<T> vl)

{

lst = vl;

}

T IEnumerator<T>.Current

{

get {

if (current == null||!active) throw new InvalidOperationException();

else return current.content;

}

}

object IEnumerator.Current

{

get

{

if (current == null||!active) throw new InvalidOperationException();

else return current.content;

}

}

public bool MoveNext()

{

if (active)

if(current.Next == null)return false;

else current = current.Next;

else

if (lst.Top != null)

{

active = true;

current = lst.Top;

}

else throw new ListIsEmpty();

return true;

}

public void Reset()

{

active = false;

current = lst.Top;

}

public void Dispose() { current = null; }

}

}

4.3 Вести в класс MiniList<T> методы GetEnumerator() и IEnumerable.GetEnumerator():

public IEnumerator<T> GetEnumerator()

{

return new Numerator(this);

}

IEnumerator IEnumerable.GetEnumerator()

{

return new Numerator(this);

}

а также дополнить определение класса MiniList<T>:

class MiniList<T> : IEnumerable<T>

5.1 Вести в класс Form1 поле:

IEnumerator<CheckBox> ven;

5.2 Установить подпункты пункта меню **Перебор кнопок**:

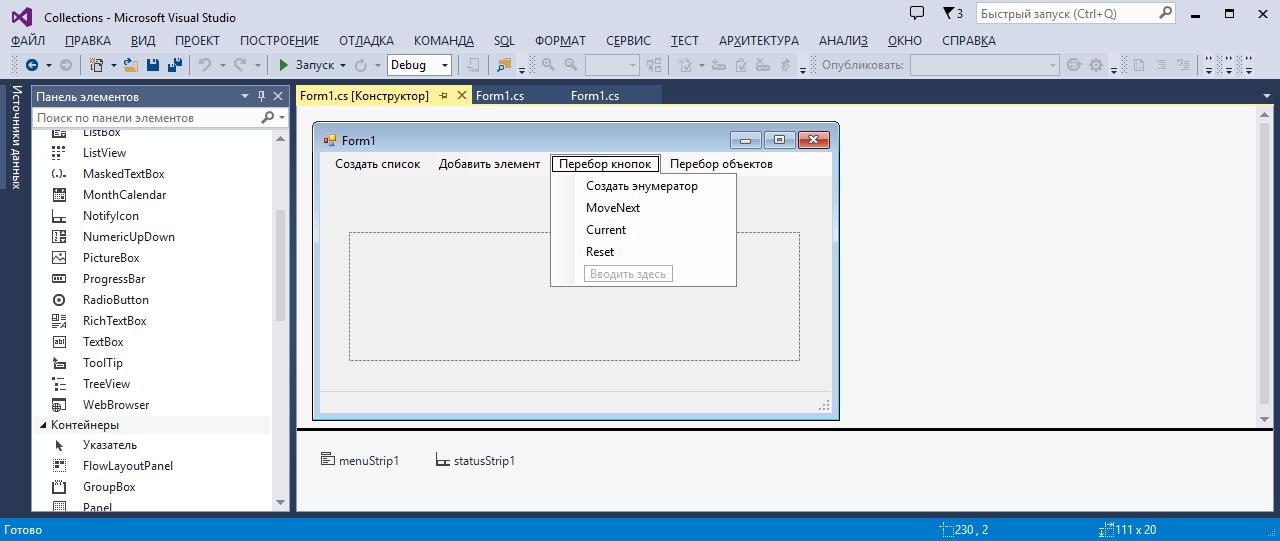
**Создать энумератор**

**MoveNext**

**Current**

**Reset**

Ниже приведён вид главного окна приложения после определения меню.



5.3 Вести в класс Form1 и подключить к пункту меню **Создать энумератор**

обработчик события Click:

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

{

ven = v.GetEnumerator();

toolStripStatusLabel1.Text = "Энумератор создан.";

}

5.4 Вести в класс Form1 и подключить к пункту меню **MoveNext** обработчик события Click:

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

{

try

{

if (ven != null)

if (ven.MoveNext())

{

ven.Current.CheckState = CheckState.Checked;

toolStripStatusLabel1.Text = "O.K.";

}

else toolStripStatusLabel1.Text = "Конец списка!";

else toolStripStatusLabel1.Text = "Энумератор не создан!";

}

catch (ListIsEmpty) {toolStripStatusLabel1.Text = "Список пуст!";}

}

Обратить внимание, что кнопка устанавливается в нажатое (checked) состояние.

5.5 Вести в класс Form1 и подключить к пункту меню **Current** обработчик события Click:

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

{

try

{

if (ven != null)

if (ven.Current.Checked)

ven.Current.CheckState = CheckState.Unchecked;

else ven.Current.CheckState = CheckState.Checked;

else toolStripStatusLabel1.Text = "Энумератор не создан!";

}

catch (InvalidOperationException) { toolStripStatusLabel1.Text = "Перебор не начат!"; }

}

Обратить внимание, что кнопка устанавливается в ненажатое (unchecked) состояние.

5.6 Вести в класс Form1 и подключить к пункту меню **Reset** обработчик события Click:

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

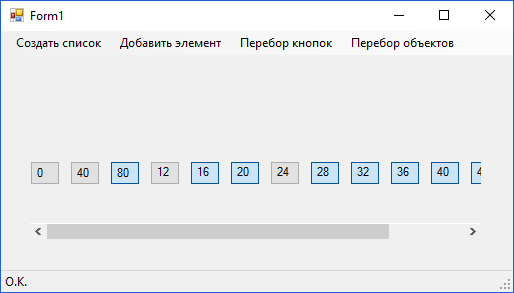
{

toolStripStatusLabel1.Text = "";

if (ven != null) { ven.Reset(); toolStripStatusLabel1.Text = "Энумератор обновлён!"; }

else toolStripStatusLabel1.Text = "Энумератор не создан!"; }

5.7 Убедиться, пользуясь введёнными в меню командами, что class MiniList<T> : IEnumerable<T> обеспечивает создание энумератора и перебор списка в соответствии с интерфейсами IEnumerable<T> и IEnumerator.

Пример тестирования показан ниже.

6.1 Вести в класс Form1 поле:

IEnumerator<CheckBox> vn;

6.2 Установить подпункты пункта меню Перебор кнопок:

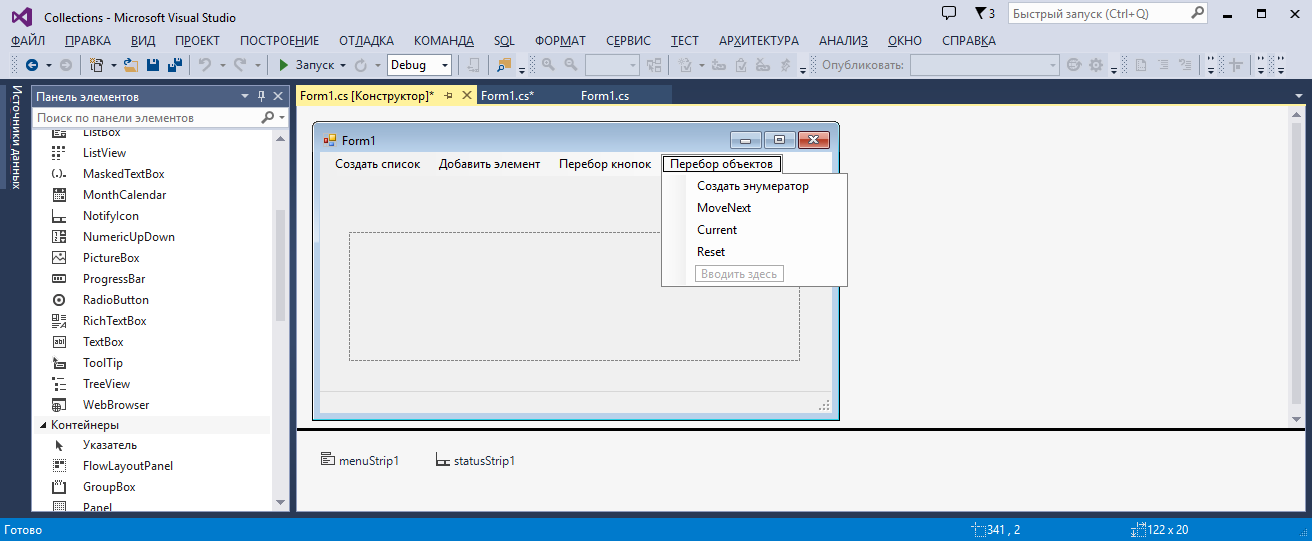
**Создать энумератор**

**MoveNext**

**Current**

**Reset**

Ниже приведён вид главного окна приложения после определения меню.



6.3 Вести в класс Form1 и подключить к пункту меню **Создать энумератор**

обработчик события Click:

private void создатьЭнумераторToolStripMenuItem1\_Click(object sender, EventArgs e)

{

vn = v.GetEnumerator();

toolStripStatusLabel1.Text = "Энумератор создан.";

}

6.4 Вести в класс Form1 и подключить к пункту меню **MoveNext** обработчик события Click:

private void moveNextToolStripMenuItem1\_Click(object sender, EventArgs e)

{

try

{

if (vn != null)

if (vn.MoveNext())

{

(vn.Current as CheckBox).CheckState = CheckState.Checked;

toolStripStatusLabel1.Text = "O.K.";

}

else toolStripStatusLabel1.Text = "Конец списка!";

else toolStripStatusLabel1.Text = "Энумератор объектов не создан!";

}

catch (ListIsEmpty) { toolStripStatusLabel1.Text = "Список пуст!"; }

}

Обратить внимание на отличия этого обработчика от обработчика для универсального списка.

6.5 Вести в класс Form1 и подключить к пункту меню **Current** обработчик события Click:

private void currentToolStripMenuItem1\_Click(object sender, EventArgs e)

{

try

{

if (vn != null)

if ((vn.Current as CheckBox).Checked)

(vn.Current as CheckBox).CheckState = CheckState.Unchecked;

else (vn.Current as CheckBox).CheckState = CheckState.Checked;

else toolStripStatusLabel1.Text = "Энумератор объектов не создан!";

}

catch (InvalidOperationException) { toolStripStatusLabel1.Text = "Перебор не начат!"; }

}

Обратить внимание на отличия этого обработчика от обработчика для универсального списка.

6.6 Вести в класс Form1 и подключить к пункту меню **Reset** обработчик события Click:

private void resetToolStripMenuItem1\_Click(object sender, EventArgs e)

{

toolStripStatusLabel1.Text = "";

if (vn != null) { vn.Reset(); toolStripStatusLabel1.Text = "Энумератор объектов обновлён!"; }

else toolStripStatusLabel1.Text = "Энумератор объектов не создан!";

}

6.7 Убедиться, пользуясь введёнными в меню командами, что class MiniList<T> : IEnumerable<T> обеспечивает создание энумератора и перебор списка в соответствии с интерфейсами IEnumerable и IEnumerator.