

## 12 КЛАСТАРДЫҢ КОМПОЗИЦИЯСЫ ЖӘНЕ КОЛЛЕКЦИЯСЫ

### 12.1 Кластардың композициясы және коллекциясы ұғымы

Егер класс өз өрістерінде басқа класс объекттерін пайдаланатын болса, онда кластардың осындай бірлестігі композиция деп аталады.

Класс композициясы бұл - бұрын жазылған қосымша кодын қайта пайдалану түрлерінің бірі. Мысалы, «дәріхана» класын «дәрілер» класының композициясы, яғни «дәрілер» класының түрлі объекттерінің массиві ретінде қарастыруға болады.

Композиция мысалы ретінде «гараж» класында «автомобиль» объекттерінің бірлестігін келтіруге болады.

Деректердің бір құбылымда бір типті объекттердің бірлестігі коллекция деп аталады.

Коллекция көптеген объекттерді өңдеу функцияларын қамтамасыз етуі керек, яғни объекттерді сақтау мен жою және объекттерді қосу, жаңарту бойынша операцияларды ұсыну.

Әдетте коллекция жеке класпен анықталады.

Бүкіл коллекцияны сипаттайтын класты коллекция класы деп атайды.

Кластар композициясы бойынша қарастырылған мысалдарда екінші класс бірінші кластың объекттер коллекциясы қызметінде болады. Мысалы, «гараж» класы «автомобиль» класының объекттер коллекциясы болып табылады. «Дәріхана» класы – «дәрілер» класы объекттерінің коллекциясы.

Барлық коллекциялар кластарын шартты түрде сызықты және сызықты емес деп бөлуге болады.

Сызықты коллекцияларды келесі коллекциялардың кластары ұсынады:

– индекстелген қол жеткізу, мысалы, сөздіктер мен анықтама кітаптары;

– тікелей қол жеткізу, мысалы, массивтер;

– ретті қол жеткізу, мысалы, стектер, кезектер, тізімдер. Көптеген «тізімдер» динамикалық массивтермен анықталған;

Сызықты емес коллекцияларды келесі коллекциялардың кластарын ұсынады:

– иерархиялық, мысалы, әр түрлі бұтақтар тәрізді құрылым. Классификацияның иерархиялық жүйесі немесе құжаттарды іздестіру массивін ұйымдастыру;

– топтық, мысалы, түрлі жиынтықтар, тораптық құрылымдар, графтар.

Кластар композициясы бұрын құрылған кластарды немесе қосымшалар үзінділерін қолданған кезде бағдарламалаудың қуатты құралы болып табылатының ерекше атап өту керек.

С# бағдарламалау ортасынада басқа кластардың бір типті объекттерін коллекциялау үшін көптеген түрлі кластар қолданылады, мысалы, тізімдер, стектер, кезектер, сөздіктер, бұтақтар, және көптеген басқа да коллекциялар.

## 12.2 Кластың композициясы мен коллекциясын қолдану мысалы

Сан алуан коллекциялар кластарының ішінен ең қарапайым кластар коллекциясын – стекті қарастырайық.

12.1-есеп. Стек элементіне КІТАП класының объектісі жатады. Кітаптарды жинағанда кітаптарды тек үстіңгі жақтан ғана алуға немесе қосуға болады.

Кластың деректерін ашық деп есептейік (кітап авторы, атауы, бағасы). Класс әдісі ретінде параметрлері берілген конструкторды ғана қолданамыз.

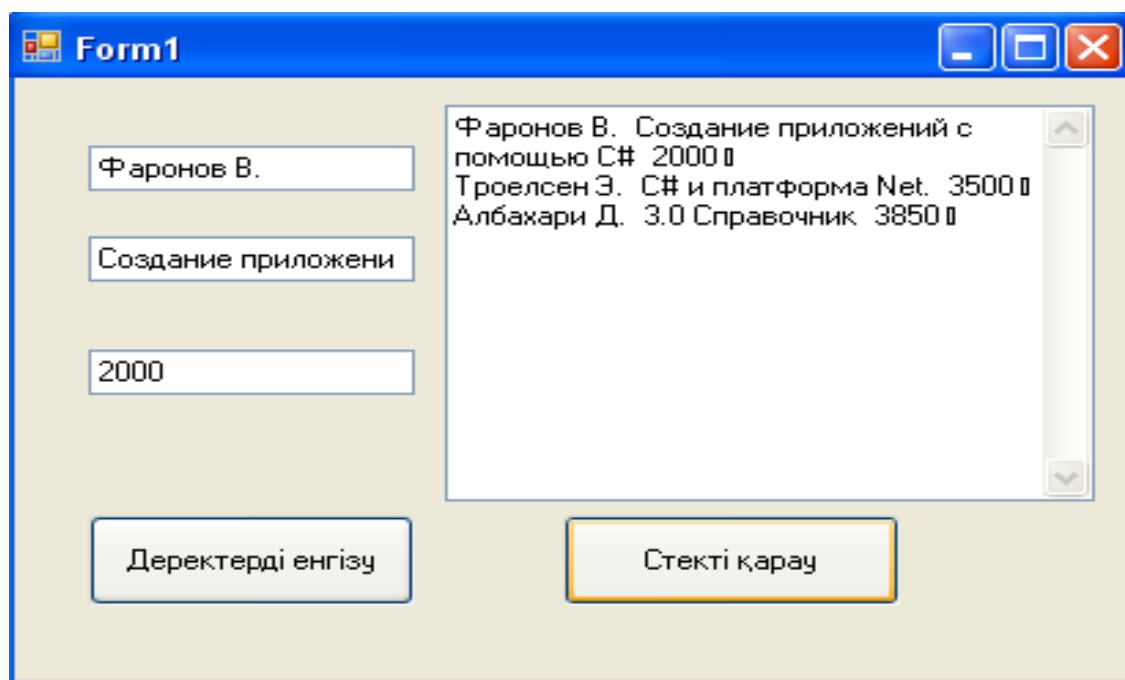
Тізімдік құрылым түрінде коллекцияны ұйымдастыру үшін стандартты Stack құрылымын қолданамыз.

Form1.cs файлының коды:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public class Kitap
        {
            public string Atayi;
            public string Avtor;
            public int Bagasi;
            public Kitap(string sa, string sb, int sc)
            {
                Avtor = sa; Atayi = sb; Bagasi = sc;
            }
        };
        public Stack<Kitap> vstek = new Stack<Kitap>();
        public string ss = "";
        private void button1_Click(object sender, EventArgs e)
        {
            string a, b;
            a = textBox1.Text;
            b = textBox2.Text;
            c = Convert.ToInt32(textBox3.Text);
            Kitap Tom = new Kitap(a, b, c);
            vstek.Push(Tom);
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Kitap Tom = new Kitap("", "", 0);
```

```
foreach (Kitap T in vstek )
{
    ss = T.Avtor + " " + T.Atayi+ " " +
    Convert.ToString(T.Bagasi) + " \n";
    textBox4.AppendText(ss);
}
}
}
```

Қосымша жұмысы 12.1-суретінде көрсетілген



12.1-сурет – Қосымшаның жұмысы

Айта кететін жәйт, С# тілінде объекттер коллекцияларын жүзеге асыратын бірнеше кластар бар, мысалы, массив, стектер, кезектер, т.б. Массивтер ғана С# тілінің конструкциясы болып келеді, ал басқа коллекциялар - .NET платформасының кластары (Framework кітапханалары). Сондықтан осы кітапхана коллекциясының құрамымен танысудың маңызы бар.

### 12.3 Framework-тың кейбір коллекциялары

Коллекциялар кластарын жүзеге асыру интерфейстер көмегімен орындалады. .NET платформасы, анығырақ айтқанда Framework кітапханасы коллекцияларды ұйымдастыру, индекс бойынша оның элементтеріне қол жеткізу, коллекцияда іздестіруді, т.б. орындау үшін бағдарламашыға қосымша интерфейстерді ұсынады.

ICollection интерфейсі объекттер коллекциялары үшін стандартты болып келеді. Ол IEnumerable, IEnumerator интерфейстерін қолданып өз

элементтерін бір бірден қарап шығуға; коллекция өлшемін анықтауға; күрделі өңдеу жұмыстары үшін коллекцияны массивке көшіруді қамтамасыз етеді.

IList интерфейсі массив типіндегі коллекцияларды құру үшін стандартты болып келеді. Ол IEnumerable, IEnumerator интерфейстерін қолданып өз элементтерін бір бірден қарап шығуды да қамтамасыз етеді. Сонымен қатар интерфейс қайта жүктелетін индексатордың көмегімен элементке тікелей қол жеткізуге мүмкіндік туғызады. IList интерфейсі индекс бойынша коллекция элементін қосуды, жоюды, редакциялауды қамтамасыз етеді.

Стандартты емес коллекциялармен жұмыс жасау үшін көптеген интерфейстер бар, мысалы, IDictionaryEnumerator интерфейсі сөздік типтегі коллекциялармен жұмыс жасауға көмекеседі.

Интерфейстерді қолдану Framework кітапханасында коллекциялардың бірнеше класын құруға мүмкіндік береді, мысалы, Array, ArrayList, LinkedList, Queue, Stack және басқалар.

Коллекцияны визуалды көрсету үшін C# тілінде бірнеше басқару элементтері (немесе коллекциялар кластары ) бар, мысалы, DataGridView, TreeView, т.б.

Сіздер коллекцияларды «Қолданбалы бағдарламалау» және «Деректер базалары» пәндерінде толық қарастыратын боласыңдар, ал осы пәнде ArrayList коллекциясының мысалында коллекциялар класымен жұмыс жасауды қарастырамыз

## 12.4 ArrayList коллекциясы

ArrayList Класс System.Collections.ArrayList атаулар кеңістігіне тиісті және кез келген типтегі объекттерді сақтау үшін қолданылады. ArrayList класының негізгі қасиеттері мен әдістері 12.1-кестеде көрсетілген.

12.1-кесте - ArrayList класының негізгі қасиеттері мен әдістері

Қасиеттер мен әдістер	Сипаттама
public static ArrayList (IList: List);	List коллекциясының негізінде ArrayList объектін құрады
public virtual int Add(Oblect Value);	Тізім соңына жаңа объектті қосады және индексті қайтарады.
public virtual void AddRange (ICollection coll)	Тізім соңына бірнеше объекттерді қосады.

12.1-кестенің жалғасы

Қасиеттер мен әдістер	Сипаттама
-----------------------	-----------

<code>public virtual int BinarySearch(Object Value);</code>	Value объектіні іздестіреді және оның индексін қайтарады, ал егер объект табылмаса, теріс санды шығарады.
<code>public virtual int Capacity {get;}</code>	Осы қасиет тек қана «оқу» үшін қолданылады, коллекцияның ағымдағы сыйымдылығын сақтайды.
<code>public virtual void Clear();</code>	Коллекцияның барлық элементтерін жояды.
<code>public virtual bool Contains (Object Value);</code>	Коллекцияның Value деген элементі бар болса, true мәні қайтарылады.
<code>public virtual int Count {get;}</code>	Осы қасиет тек қана «оқу» үшін қолданылады, коллекцияның ағымдағы ұзындығын сақтайды.
<code>public static ArrayList FixedSize(ArrayList AL);</code>	Әдіс объектіні қайтарады. Оның элементтерін өзгертуге болады, бірақ қосуға немесе жоюға болмайды.
<code>public virtual IEnumerator GetEnumerator();</code>	Объект үшін итераторды қайтарады.
<code>public virtual ArrayList GetRange(int Indx, int Count);</code>	Элементтер диапазонын қайтарады.
<code>public virtual int IndexOf(Object Value);</code>	Value деген мәні бар элемент индексін қайтарады.
<code>public virtual void Insert (int Indx, Object Value);</code>	Коллекцияның Indx керекті орнына Value элементін орналастырады.
<code>public virtual void InsertRange(int Indx, ICollection col);</code>	Элементтер диапазонын кіргізеді.
<code>public virtual bool IsFixedSize{get;}</code>	Егер объектінің өлшемі тұрақты болса, онда true мәнін қайтарады.
<code>public virtual bool IsReadOnly{get;}</code>	Егер объектінің «оқу» үшін ғана арналған элементтері болса, онда true мәні қайтарылады.
<code>public virtual Object this[int Indx]{set; get;}</code>	Бұл қасиет элементтерді индексі бойынша қолдануға мүмкіндік береді.

#### 12.1-кестенің жалғасы

<code>public virtual void RemoveRange(int Indx, int</code>	Индексі Indx тең элементтен бастап коллекциядан count элементті жояды.
--	--

count);	
public static ArrayList Repeat(Object Value, int count);	Value элементі count рет қайталанатын коллекцияны қайтарады.
public virtual void Reverse();	Элементтердің реттілігін кері бағытқа өзгертеді.
public virtual void SetRange (int Indx, ICollection col);	Indx индексінен бастап col коллекциясын кіргізеді.
public virtual void Sort();	Коллекцияны сұрыптайды
public virtual void TrimToSize();	Коллекцияның сиымдылығын орнатады. Сиымдылық элементтерінің санына тең.
public static ArrayList ReadOnly(ArrayList AL);	Коллекция элементтеріне тек қана «оқу» үшін режимін орнатады
public virtual void Remove (Object Value);	Коллекциядан мәні Value-тең бірінші элементті жояды.
public virtual void RemoveAt (int Indx);	Коллекциядан индексі Indx-тең элементті жояды.

Әдетте ArrayList коллекциясының сиымдылығы 16 элементке тең. Коллекция кеңейген кезде оның сиымдылығы екі еселенеді, 32, 64, 128 элементтерді құрайды. TrimToSize() әдісі қолданылмайтын элементтерді кесіп тастайды.

12.1-есеп. ArrayList класын «Автомобиль» объекттерін біріктіретін «Гараж» коллекциясын ұйымдастыру үшін қолдану. Коллекциямен жұмыс жасау үшін 12.1-кестесінің қасиеттері мен әдістерін қолдану керек. Бұл мысалда «Автомобиль» объектісінің өрістер саны тек екеу болады – үлгісі мен бағасы. Қосымша беттер жиынтығы – TabControl басқару элементі қолданылды, осы элемент арқылы бағдарлама менюі мен «қосымша формалар» қосылады.

Бағдарламада Panel басқару элементі – бағдарламаның нәтижесін шығару немесе жасыру үшін панелі қолданылады: «Қарап шығу» режимі, «Баға графигін қарап шығу» командасы.

ArrayList коллекциясының ерекшелігі – кез келген типтегі объекттерді сақтау үшін қолданылуында, сондықтан 12.1-кестесіндегі кейбір әдістер интерфейстік кластарды қосымша күйге келтіруді талап етеді.

Бағдарламаны тексеру барысында егер коллекция объекттер саны 4-тен аспаса, онда коллекция сиымдылығы 16-ға емес 4-ке тең болады. Коллекцияны кеңейткен кезде сиымдылық екі есе өседі – 8 (егер объекттер саны 4-тен үлкен, бірақ 9-кіші болса), 16 және т.б.

Form1.cs файлының коды:

```
using System;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Drawing;
```

```

using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            textBox3.Text = "";
            panel2.Visible = true;
            panel3.Visible = false;
            panel1.Visible = false;
        }
        public class Avto
        {
            public string Marka;
            public int Bagasi;
            public Avto(string sm, int sc)
            {
                Marka = sm; Bagasi = sc;
            }
        };
        public ArrayList Garaj = new ArrayList();

        private void button1_Click(object sender, EventArgs e)
        {
            panel2.Visible = true;
            panel3.Visible = false;
            panel1.Visible = false;
            string n;
            int c;
            n = textBox1.Text;
            c = Convert.ToInt32(textBox2.Text);
            Avto at = new Avto(n, c);
            Garaj.Add(at);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            panel2.Visible = false;
            panel3.Visible = false;
            panel1.Visible = true;
            int i = 0;
            dataGridView1.Rows.Clear();
            foreach (Avto at in Garaj)
            {
                dataGridView1.Rows.Add();
                dataGridView1.Rows[i].Cells[0].Value = at.Marka;
                dataGridView1.Rows[i].Cells[1].Value = at.Bagasi;
                i++;
            }
        }
    }
}

```

```

    }
}

private void button3_Click(object sender, EventArgs e)
{
    panel2.Visible = false;
    panel3.Visible = false;
    panel1.Visible = false;

    this.Invalidate();
}

private void button4_Click(object sender, EventArgs e)
{
    panel2.Visible = false;
    panel3.Visible = true;
    panel1.Visible = false;
    int i = Garaj.Capacity;
    int j = Garaj.Count;
    textBox3.AppendText("Коллекция " + j.ToString() + "
элементтерден тұрады" + "\r\n");
    textBox3.AppendText("Ағымдағы коллекцияның өлшемі = " +
i.ToString() + " элемент" + "\r\n");
}

private void button5_Click(object sender, EventArgs e)
{
    panel2.Visible = false;
    panel3.Visible = true;
    panel1.Visible = false;
    Avto at1 = new Avto("", 0);
    Avto at2 = new Avto("", 0);
    int i=0;
    foreach (Avto at in Garaj)
    {
        if (i==0) at1=at;
        at2 = at;
        i++;
    }
    int j = Garaj.Count;
    Garaj.RemoveAt(j-1);
    Garaj.RemoveAt(0);
    Garaj.Insert(0, at2);
    Garaj.Insert(j-1, at1);
    textBox3.AppendText("Алмастыру аяқталды" + "\r\n");
}

private void button6_Click(object sender, EventArgs e)
{
    Garaj.RemoveAt(0);
    panel2.Visible = false;
    panel3.Visible = true;
    panel1.Visible = false;
}

```

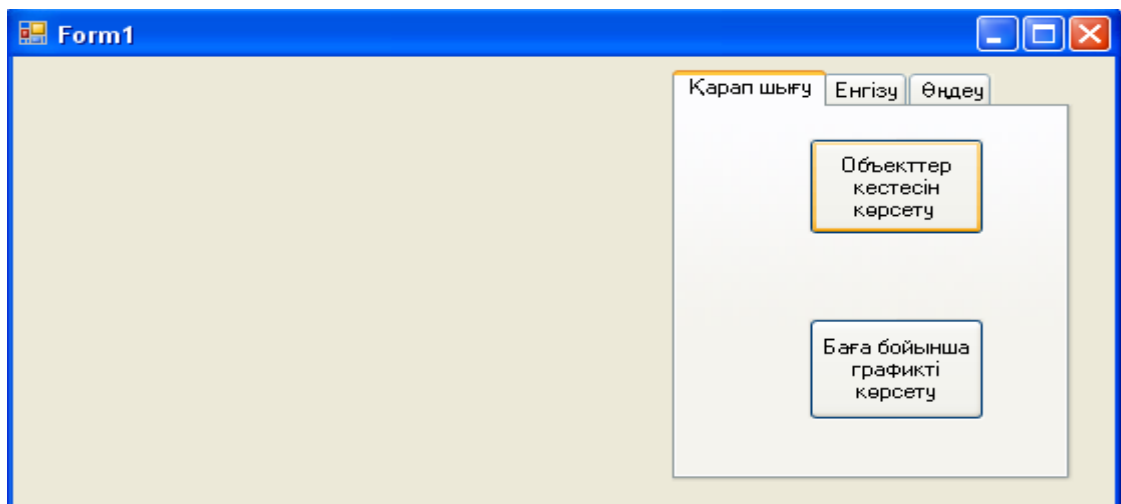


```

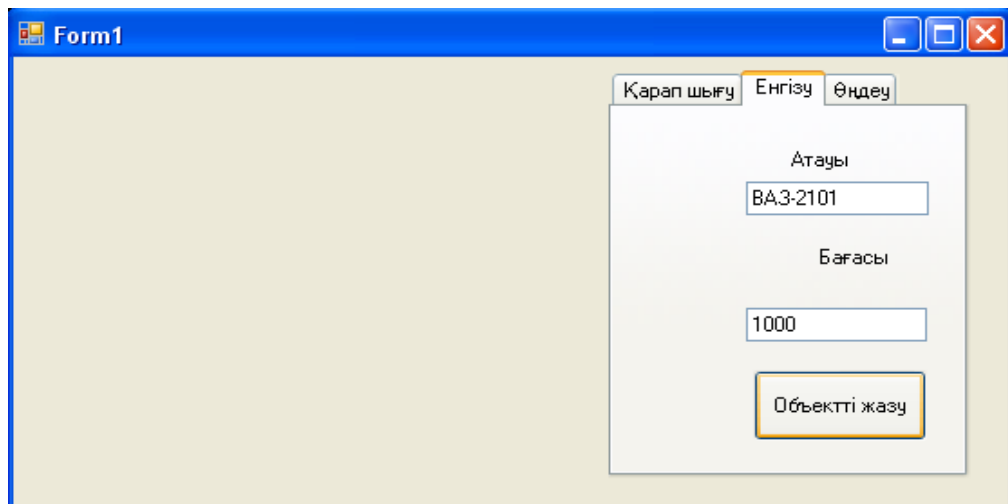
        textBox3.AppendText("Коллекциядан 0 элемент жойылды" +
"\r\n");
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        string ss;
        Graphics g = e.Graphics;
        Pen myPen = new Pen(Color.Red, 2);
        g.DrawLine(myPen, 50, 150, 50, 50);
        g.DrawLine(myPen, 50, 150, 250, 150);
        int h,i = 0;
        foreach (Avto at in Garaj)
        {
            h = at.Bagasi/100;
            if (i < 8)
            {
                g.FillRectangle(Brushes.Blue, i * 25 + 55, 150 - h, 20,
h);
                ss = i.ToString();
                g.DrawString(ss, new Font("10_IC_1", 12), Brushes.Black,
(i + 1) * 25 + 30, 160);
            }
            i++;
        }
    }
}

```

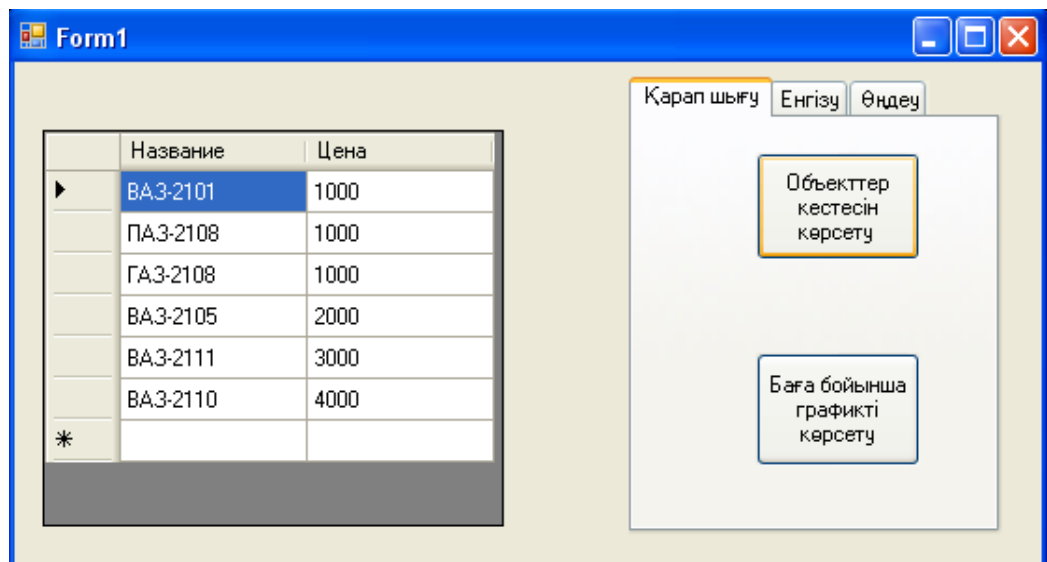
Қосымша жұмысы 12.2 –12.6 суреттерінде көрсетілген.



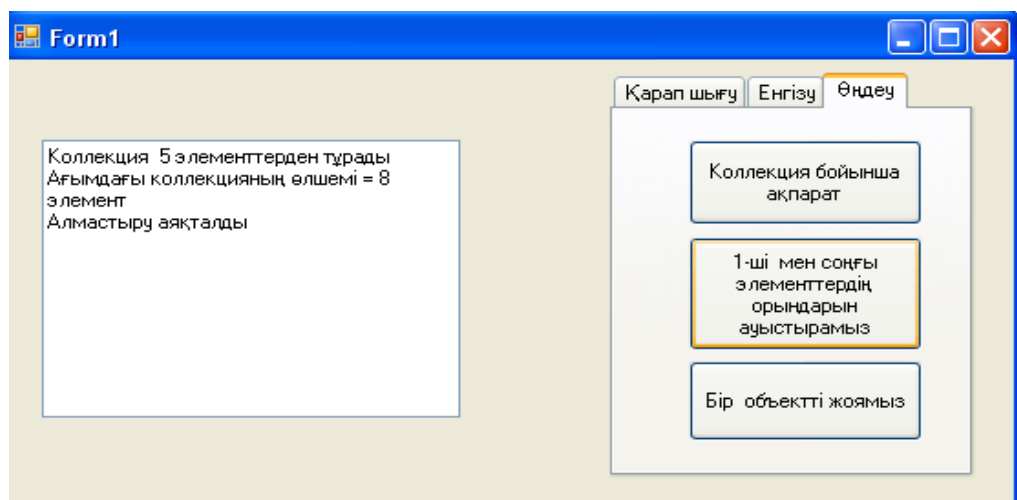
12.2-сурет – Суреті бар қосымша жұмысының терезесі



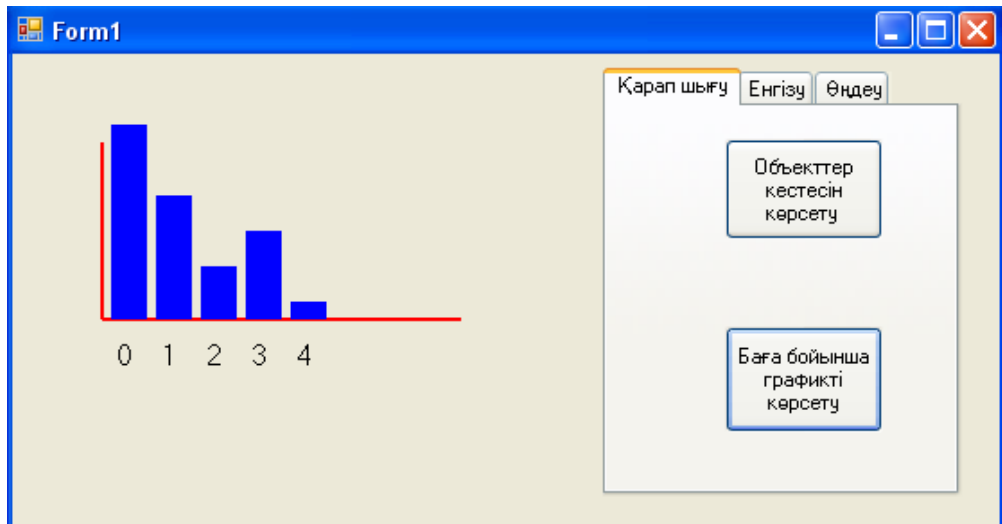
12.3-сурет – Коллекцияның деректерін енгізу режимі



12.4-сурет – Коллекция объекттерінің кестелерін қарау режимі



12.5-сурет – Коллекция туралы ақпаратты қарау режимі



12.6-сурет – Коллекция объекттерінің баға графигін көрсету

Осы мысалда негізгі назар жаңа басқару элементтерін және кластарды қолдану технологиясына аударылған.

## 12.5 Өзін-өзі тексеру сұрақтары

- 1 Класс композициясы ұғымы.
- 2 Класс композициясының негізгі мақсаты неде?
- 3 Класс коллекцияның ұғымы.
- 4 Бір типтегі объекттерді коллекциялауда әдетте қандай біріктіретін құрылым қолданылады?
- 5 Қандай коллекциялар тікелей қол жеткізу коллекцияларына жатады?
- 6 Қандай коллекциялар тізбекті қол жеткізу коллекцияларына жатады?
- 7 Қандай коллекциялар иерархиялық коллекцияларға жатады?
- 8 Қандай типтегі объекттер ArrayList класының коллекциясына қосыла алады?
- 9 ArrayList класының қандай қасиеті коллекцияның ағымдағы сиымдылығын сақтайды?
- 10 ArrayList класының қандай қасиеті коллекцияның ағымдағы ұзындығын сақтайды?
- 11 ArrayList класының қандай әдісі объекті коллекцияның керекті орнына орналастыруға мүмкіндік береді?

