

2 БАСҚАРУ ЭЛЕМЕНТТЕРІ

2.1 Формаларды визуалды жобалау технологиясы

Визуалды бағдарламалау ортасында Windows.Forms.Designer (форма құрастырушысы немесе дизайнері) – интерактивті режимде басқару элементтерін орналастыра отырып форманы визуалды жобалауға мүмкіндік беретін құрал. Windows.Forms.Designer артықшылығы мынада: сіз формада басқару элементтерін керекті орынға орналастыра аласыз, сонымен қоса оның қасиеттерінің нақты мәндерін анықтаудың қажеті жоқ, мысалы, өлшемнің, орналасудың сандық мәні. Көптеген қасиеттердің мәндерін автоматты түрде форма құрастырушысы арқылы беруге болады, бірақ форма құрастырушысы Сізге белгілі дәрежеде ғана көмектесе алады.

Windows қосымшасында (консольді қосымшаға қарағанда) форма құрастырушысы автоматты түрде бірнеше .cs кеңейтуі бар кластарды құрайды, мысалы, Form1.cs және Program.cs.

C#-та кластар синтакстік тұрғыдан бірнеше бөліктен тұра алады, олардың әрқайсысы “partial” (бөліктік) кілттік сөзінен басталады. Бір кластың сипаттамасын бөліктерге бөлу мүмкіндігі үлкен класпен жұмыс істеуді жеңілдетеді. Кластың әрбір бөлігі жеке файлдарда, өз атауымен сақталады. Мысалы, алдыңғы бөлімдегі мысалда автоматты түрде Form1-дың .cs кеңейтуі бар екі файл – Form1.cs және Form1.Designer.cs құрылады. "Form1.cs" файлында сақталатын, әзірлеушіге арналған Form1 класының бірінші бөлігінде автоматты түрде құрылатын басқару элементтерінің оқиғалар өңдеушісі орналасады. Формалармен жұмыс істеуге негізделген бағдарламалаудың осындай технологиясы визуалды, оқиғалармен басқарылатын бағдарламалау технологиясы деп аталады. Мысалы, алдыңғы бөлімде қарастырылған Form1.cs файлынан үзінді:

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            int a,b,c,p;
            a = Convert.ToInt32(textBox1.Text);
            b = Convert.ToInt32(textBox2.Text);
            c = Convert.ToInt32(textBox3.Text);
            p = a + b + c;
            . . .
        }
    }
}
```

Form1 класының екінші бөлігі "Form1.Designer.cs" файлында орналасады. Кластың осы бөлігін автоматты түрде форма құрастырушысы толтырады. Визуалды жобалау барысында формаға әр түрлі басқару

элементтерін орналастырып, олардың қасиеттерін өзгертіп, оқиғалар өңдеушілерін анықтағанда форма құрастырушысы осы әрекеттерді сәйкес кластағы объекттер әрекеттеріне трансляциялайды, сәйкес кодты құрайды және осы кодты Form1 класына орналастырады.

Форма құрастырушысының жұмысына араласудың, Form1 **класының** коды түзеуді қажет етпейдә жоқ. Алайда оның жұмысы мен оның құрған коды туралы білу керек.

Төменде алдыңғы бөлімде қарастырылған Form1.Designer.cs файлынан үзінді көрсетілген:

```
namespace WindowsFormsApplication1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        . . .
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.button1 = new System.Windows.Forms.Button();

            . . .
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(12, 9);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(174, 13);
            this.label1.TabIndex = 0;
            this.label1.Text = "А үшбұрышының қабырғаларын енгізіңіз";
            . . . и т.б.
        }
    }
}
```

Жобада автоматты түрде құрылатын Program.cs класында статикалық Main() әдісі болады. Қосымша іске қосылғанда Windows жүйесі Main() әдісін іздейді және ондағы нұсқауларды орындай бастайды. Main() әдісін әдетте бағдарламаға кіру нүктесі деп атайды.

Төменде алдыңғы бөлімде қарастырылған мысалдың Program.cs файлы көрсетілген:

```
namespace WindowsFormsApplication1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
    }
}
```

```

static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
}
}

```

Консольді қосымшаларда Main() әдісінің денесі бос болады және онда кодты жоба әзірлеушісі жазу керек. Ал Windows қосымшаларында Main() әдісі керекті нұсқаулармен толықтырылып қойылады және әдетте әзірлеуші оны өзгертпейді. Автоматты құрылған Main() әдісі нені орындайды? Ол FCL кітапханасына тиісті Application класының үш статикалық әдістерін кезек-кезек шақыра отырып жұмыс істейді.

Application.EnableVisualStyles(); әдісі Windows XP стилінде құрылатын қосымша компоненттерін ұсынады.

Application.SetCompatibleTextRenderingDefault(false); әдісі true – GDI+, false – GDI мәтінін экранға шығару механизмін анықтайды.

Application.Run(new Form1()); әдісі Program класының негізгі әдісі және Main() әдісінде ол жалғыз болып келеді.

Негізгі жұмысты Run әдісі орындайды, оны шақыру процесінде Form1 класының объектісі құрылады және форма – объектің визуалды көрінісі ашылады. Формада орналасқан барлық басқару элементтерінің инициализациясы аяқталғаннан кейін Windows хабарларын өңдеу циклі қосылады, қосымша Windows-тан хабарларды күтеді. Ал пайдаланушы болса, мысалы, мына әрекеттерді орындай алады: форма өрістеріне деректерді енгізу, батырманы басу.

Визуалды бағдарламалауды орындау үшін Toolbox панелінде орналасқан басқару элементтерінің қызметін қысқаша қарастырып өту керек.

2.2 Toolbox панеліндегі басқару элементтері

Басқару элементтер немесе компоненттер формаға ToolBox панелінен орналастырылады (View ► ToolBox). Оқу құралының осы бөлімінде ToolBox панеліндегі қарапайым басқару элементтері қысқаша сипатталған.

Басқару элементтерімен танысуды формада үнемі қолданылатын Label элементінен бастайық.

2.2.1 Label – таңба (метка). Таңба формада мәтінді оналастыру үшін керек, мәтін элементтің Text қасиетінде сақталады. Мәтіннің қарпін (Font қасиеті), түсін (BackColor қасиеті), !!!(ForeColor) мәтінді туралауды (TextAlign) анықтауға болады. Таңба өз өлшемін автоматты түрде өзгерте алады (AutoSize = True қасиеті). Онда суретті (Image қасиеті) орналастыруға, оның мөлдірлігін (BackColor қасиетіне Color.Transparent мәнін беру керек) анықтауға болады .

Label басқару элементі енгізу фокусын қабылдамайды (тышқан пернесін, пернетақтаны басқанда оқиға өңдеушісін құрай алмайды).

2.2.2 Button – батырма. Button басқару элементі енгізу фокусын қабылдай алады, сонымен қатар тышқанды шерту – негізгі оқиға болып есептеледі (Click).

Егер форманың Accept Button қасиетіне батырма атауын көрсетсек, онда Enter пернесін басқанда Click оқиғасы шақырылады, ондай батырманың жиектемесі болады. Ал егер форманың Cancel Button қасиетіне батырма атауын көрсетсек, онда Esc пернесін басқанда батырманың Click оқиғасы шақырылады. Батырма мәтінінің өлшемін өзгертуге, фонның түсі мен суретін анықтауға болады. Батырмада суретті Image немесе ImageList, ImageIndex қасиеттері анықтауға өзгертуге болады.

Батырмалар көбінесе диалогтық терезелерде қолданылады. Диалогтық терезенің модальдық қасиеті болады, яғни осы терезе жабылғанша қосымшамен ары қарай жұмыс жасау тоқтатылады. Ақпаратты ОК (Yes) батырмасын басып растауға немесе терезені жабу батырмасы арқылы оның күшін жоюға болады, нәтижесінде терезе жабылады. Терезенің жабылғаны туралы мәліметті сақтау үшін батырмада DialogResult қасиеті анықталады, бұл қасиет стандартты мәндерді System.Windows.Forms кеңістігінде анықталған DialogResult тізімінен қабылдай алады. DialogResult тізімінің мәндері 2.1-кестеде көрсетілген.

2.1-кесте – DialogResult тізімінің мәндері

Мәні	Сипаттамасы	Мәні	Сипаттамасы
None	Терезе жабылмайды	Ignore	Ignore батырмасы басылды
ОК	ОК батырмасы басылды	Yes	Yes батырмасы басылды
Cancel	Cancel батырмасы басылды	No	No батырмасы басылды
Abort	Abort батырмасы басылды	Retry	Retry батырмасы басылды

2.2.3 TextBox енгізу өрісі. TextBox Компоненті мәтінді енгізуге, редакциялауға мүмкіндік береді, мәтін Text қасиетінде сақталады. Жолдардың санына шек қойылмайды (шамамен 32 000 символға дейін), осы компонент арқылы масканы анықтап парольді енгізуге болады (PasswordChar қасиеті).

Text қасиеті тек бір жолды ғана енгізуге, ал Lines қасиеті бірнеше жолды енгізуге мүмкіндік береді. Соңғы қасиет бойынша элементтергі жолдар массив түрінде сақталады, сондықтан оларға индекс түрінде қол жеткізуді ұйымдастыруға болады.

Бірнеше жолды енгізу және шығару үшін Multiline, ScrollBars, WordWrap қасиеттері қолданылады. ReadOnly қасиеті тек оқу рұқсатын орнатады.

Элементтің тазарту (Clear), ерекшелену (Select), буферге көшіру (Copy), буферден кірістіру (Paste), т.б. әдістері бар. Компонент көптеген оқиғаларды өңдей алады, олардың негізгілері - KeyPress және KeyDown.

2.2.4 ListBox — тізім. ListBox компоненті бір немесе бірнеше пункттерді таңдау мүмкіндігін ұсынады. SelectMode қасиетінде келесі бірнеше мәндер болуы мүмкін: None — пункттерді таңдауға рұқсат жоқ; One — бір пунктті ғана таңдауға болады; MultiSimple — бірнеше пункттерді таңдауға болады; MultiExtended — Shift және Ctrl пернелерін басып тұрып бірнеше пункттерді таңдауға болады: егер Shift пернесі басылып тұрса, пункттердің шексіз диапазоны таңдалып алынады; егер Ctrl пернесі басылып тұрса, пункттердің еркін (үздіксіз болуы шарт емес) диапазоны таңдалып алынады. Егер MultiColumn қасиетіне True мәні меншіктелсе, тізім пункттері бірнеше бағанада орналаса алады, ал ColumnWidth қасиеті бағана жалпақтығын анықтайды. Егер бағана компоненттің ені бойынша шығып кететін болса, онда автоматты түрде көлденең айналдыру жолағы (полоса прокрутки) қойылады.

2.2.5 RadioButton ауыстырып-қосқышы. Ауыстырып-қосқыш пайдаланушыға бірнеше нұсқалардың ішінен бірін таңдауға мүмкіндік береді, сондықтан әдетте ауыстырып-қосқыштар топқа біріктіріледі. Егер олардың бірі белгіленсе (Checked қасиеті), қалғандары автоматты түрде босатылады. Бағдарламашы мәтіннің стилін, түсін өзгерте алады. Ауыстырып-қосқышы үшін фон түсі мен фонның суретін анықтауға болады. Ауыстырып-қосқыштарды тікелей формаға орналастыруға болады. Егер формада ауыстырып-қосқыштардың бірнеше тобын орналастыру керек болса, онда оларды Group немесе Panel компонентінің ішіне орналастырады. Appearance қасиеті ауыстырып-қосқыштың бейнесін анықтайды: батырма (Button) немесе қалыпты (Normal) түрде бейнеленеді.

2.3 Басқару элементтерін қолдану мысалы

Осы бөлім үшін синус, косинус және тангенс тригонометриялық функцияларын есептеу мысалы таңдап алынды. Айнымалылардың мәндері (бұрыш радианмен берілген) бағдарламаның диалог режимінде беріледі. Сонымен қатар диалог режимінде есептелетін функцияның аты және монитор экранына шығару форматының разрядтар саны – есептеу дәлдігі беріледі. Осы есепті дайындау үшін жобада келесі басқару элементтері қолданылған: Label, Button, Panel, RadioButton, ListBox және TextBox.

Form1.cs файлының коды:

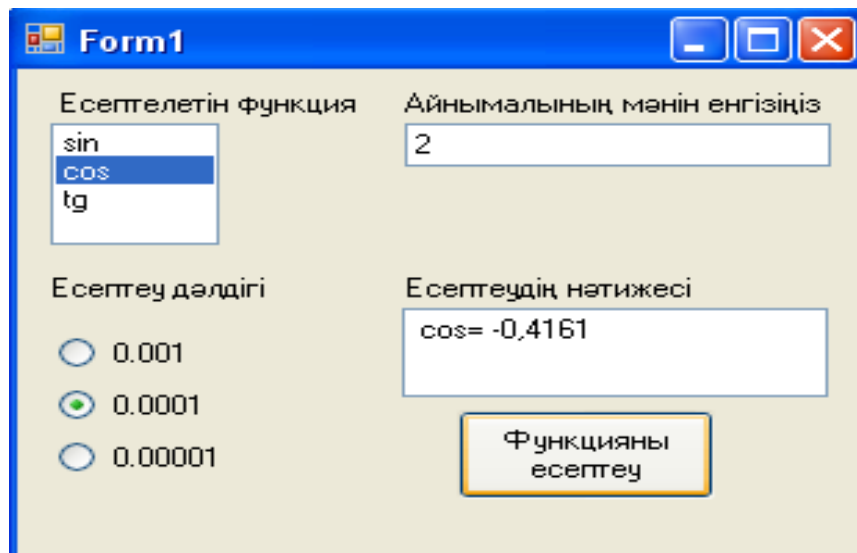
```
using System;  
using System.Collections.Generic;
```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            string ff = "F3";
            string fu = "sin";
            Double x=0;
            // бұрыштың мәнін анықтаймыз
            x = Convert.ToDouble(textBox2.Text);
            //Выбор функции
            if (listBox1.SelectedIndex == 1) fu = "cos";
            if (listBox1.SelectedIndex == 2) fu = "tn";
            // есептеу дәлдігі
            if (radioButton1.Checked)
            {
                ff = "F3";
            }
            else
            if (radioButton2.Checked)
            {
                ff = "F4";
            } else
            if (radioButton3.Checked)
            {
                ff = "F5";
            };
            switch (fu)
            {
                case "sin": textBox1.Text = " sin= " +
                    Math.Sin(x).ToString(ff); break;
                case "cos": textBox1.Text = " cos= " +
                    Math.Cos(x).ToString(ff); break;
                case "tn": textBox1.Text = " tn= " +
                    (Math.Sin(x) / Math.Cos(x)).ToString(ff); break;
            }
        }
    }
}

```

Қосымшаның жұмысы 2.1-суретте көрсетілген.



2.1-сурет – Функцияны есептеу бойынша қосымшаның жұмысы

Бағдарлама жұмысына қосымша түсіндірме берудің қажеті жоқ. Басқа басқару элементтері, яғни меню, диалогтық терезелер, суреттер, т.б. оқу құралының келесі бөлімдерінде қарастырылатын болады.

2.4 Өзін-өзі тексеру сұрақтары

- 1 NET Платформасының Windows.Forms.Designer не үшін қолданылады?
- 2 Форма класының сипаттамасында “partial” қызметтік сөзі нені білдіреді?
- 3 Хабарламалар өңдеуіштері форма класына арналған сипаттаманың қандай бөлігінде орналасады?
- 4 Windows қосымшасының орындалуы қандай әдістен басталады?
- 5 Бағдарлама іске қосылғанда қандай әдіс Form1 класының объектісін құрайды?
- 6 Label басқару элементінің қызметі?
- 7 Label басқару элементінің қандай қасиеті форма терезесіне ақпаратты шығара алады?
- 8 Label басқару элементінің қандай қасиеті арқылы оның «мөлдірлігін» анықтауға болады?
- 9 Батырманың қандай қасиеті арқылы оған суретті орналастыруға болады?
- 10 Диалогтық терезе жабылғанша қосымшамен ары қарай жұмыс жасауға кедергі келтіретін диалогтық терезені қалай атайды?