

13 ДЕЛЕГАТТАР

13.1 Делегат ұғымы

«Біртипті» әдістерді қолдана отырып есептеу жүргізуді қажет ететін көптеген есептер болады, мысалы, нақты типтегі математикалық функциялар (тригонометриялық функциялар, логарифмдер, экспонент, т.б.). Кейде осындай есептерде формалды параметрі ретінде есептелетін функциясының атауы болып келетін әдісті қолдану қажеті туындайды.

C# тілінде әдістер тек класс ішінде ғана орналастырыла алады. Сондықтан C# тілінде «біртипті» әдістерге сілтемелерді сақтауға мүмкіндік беретін арнайы класс қосылған. Ол класс Делегат деп аталады.

Делегат дегеніміз – әдістерге сілтемелерді сақтауға арналған арнайы класс.

Анықтама бойынша класс типіндегі айнымалы объект болып келеді. Әрқайсысы функция болып келетін (немесе функцияға сілтеме) объекттердің жиынтығын сипатауға мүмкіндік беретін класс функционалдық тип деп аталады.

Сонымен, C# тілінде делегаттар функционалдық типтерді сипаттау үшін арналған.

Осындай кластың даналары – функцияларға (әдістерге) сілтемелер болып табылады, айнымалыларға сияқты оларға да компьютер жадысынан орындар бөлінеді (олардың алғашқы адрестері функцияларға кіру «нүктелері» болып келеді және сілтемелермен беріледі).

Делегаттың осындай ерекшелігі оны қолданудың екі жолын анықтады - есепті өз бетімен шешу үшін немесе оқиғаларды қолдау үшін (келесі бөлімде қарастырылады).

Осы бөлімде кейбір есептерді шешу үшін делегаттардың жеке қолданылуын қарастырамыз.

13.2 Делегаттың сипаттамасы

Формат записи делегата фактически задает сигнатуру (описание) методов, которые могут быть вызваны с его помощью:

Делегаттың жазылу пішімінде әдістердің сипаттамасы берілген:

```
[спецификаторлар] delegate <тип> <атауы>  
(<параметрлер>);
```

мұнда

спецификаторлар делегатқа қол жеткізу шартын анықтайды;

delegate — қызметтік сөз;

<типi> — қайтарылатын нәтиженің типі;

<атауы> — делегат атауы (бірегей идентификатор);

<параметрлер> — шақырудың формалды параметрлері.

Мысалы, нақты типтегі аргументі бар нақты типтегі барлық функцияларды сипаттау келесі түрде жазылады:

```
public delegate double Funk(double argum);
```

Осы сипаттамаға сәйкес кез-келген функцияны класс конструкторының, яғни делегаттың шақыру параметрі ретінде қолдануға болады. Ол делегаттың нақты данасын - функцияға сілтемені қайтарады. Делегат жұмысының ерекшелігі – сілтемелердің қосымшаны компиляциялау кезеңінде емес, қосымша жұмысының барысында (динамикалық түрде) құрылуында.

13.3 Делегатты қолдану мысалы

Делегат жұмысының механизмін жақсы түсіну үшін келесі есептің шешімін қарастырайық: нақты типтегі аргументі бар нақты типтегі бес функцияны есептеу үшін қосымшаны дайындау керек: Sin(x), Log(x), Cos(x), Exp(x), Round(x). Есептеу функциясы үшін делегатты қолдану керек.

Form1.cs файлының коды:

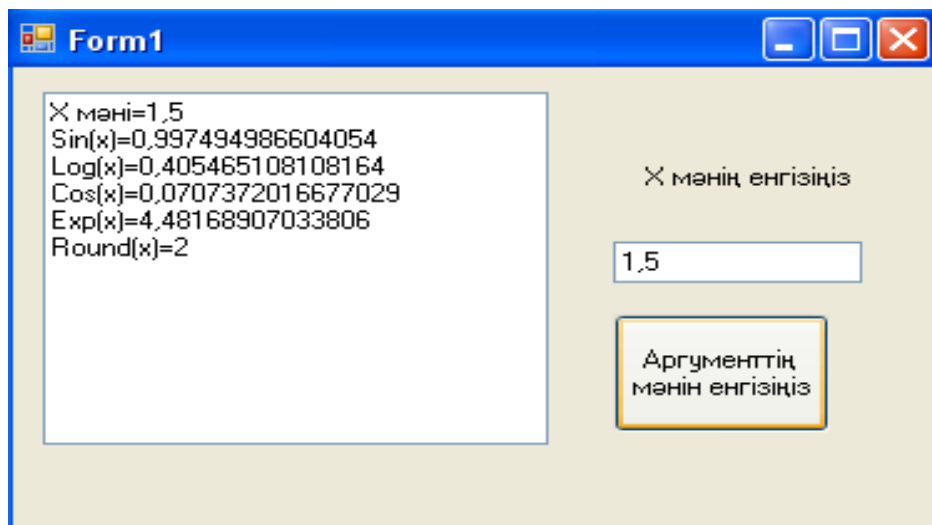
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        public delegate double Funk(double x); // делегатты жариялау
        public double rab(Funk f, double x) { return f(x); }
        // делегатты қолданатын функцияны жариялау
        private void button1_Click(object sender, EventArgs e)
        {
            string st;
            double x, y;
            textBox2.Text = "";
            x = Convert.ToDouble(textBox1.Text);
            st = "x-тің мәні=" + textBox1.Text + "\r\n";
            textBox2.AppendText(st);
            y = rab(Math.Sin, x); // делегатты қолдану
            st = "Sin(x)=" + y.ToString() + "\r\n";
            textBox2.AppendText(st);
            y = rab(Math.Log, x); // делегатты қолдану
            st = "Log(x)=" + y.ToString() + "\r\n";
```

```

textBox2.AppendText(st);
y = rab(Math.Cos, x); // делегатты қолдану
st = "Cos(x)=" + y.ToString() + "\r\n";
textBox2.AppendText(st);
y = rab(Math.Exp, x); // делегатты қолдану
st = "Exp(x)=" + y.ToString() + "\r\n";
textBox2.AppendText(st);
y = rab(Math.Round, x); // делегатты қолдану
st = "Round(x)=" + y.ToString() + "\r\n";
textBox2.AppendText(st);
}
}
}

```

Қосымша жұмысы 13.1-суретте көрсетілген.



13.1-сурет – Делегатты қолдану

Қосымша кодына мына бөлікті қосуға болады, мысалы,

```

private void button1_Click(object sender, EventArgs e)
{
    string st;
    Funk[] ff={Math.Sin, Math.Log, Math.Cos,
Math.Exp,Math.Round};
    string[] sfu = { "Sin", "Log", "Cos", "Exp", "Round" };
    double x, y;
    textBox2.Text = "";
    x = Convert.ToDouble(textBox1.Text);
    st="X мәні =" +textBox1.Text+ "\r\n";
    textBox2.AppendText(st);
    for (int i = 0; i < 5; i++)
    {
        y = rab(ff[i], x);
        st = sfu[i]+"=" + y.ToString() + "\r\n";
        textBox2.AppendText(st);
    }
}

```

```
}
```

Бағдарламада делегаттарды жариялау және қолдану идеясы осы мысал бағдарламасының кодында толық қарастырылған.

Делегаттардың ерекшелігін тағы да бір рет атап өтейік: шақырылатын әдістерді динамикалық түрде шақыру (бағдарламаның орындалуы барысында). Делегаттардың осы ерекшелігі бағдарламаның базалық конструкцияларын құру кезінде жиі қолданылады, осы конструкцияда әр түрлі бағдарлама үзінділерін жазуға болады, мысалы «біртипті» әдістері бар менюдің фрагменттері.

13.4 Делегаттардың үйлесімділігі

Делегаттардың «үйлесімділігін» қарастыра отырып екі сұрақты талқылау керек – делегаттар типінің үйлесімділігі және делегаттардың объекттер даналарының үйлесімділігі.

Делегаттар типтерінің жазылу пішімдері үйлесімді болғанымен, олар бір-бірімен үнемі үйлесімді бола бермейді, мысалы,

```
public delegate void Funk1();
public delegate void Funk2();
. . .
Funk1 d1 = Metod1;
Funk2 d2 = d1; // типтер үйлесімділігі туралы қате
```

Егер делегатты жариялау үшін бір ғана тип қолданылған болса, онда делегаттардың даналары үйлесімді (тең) болады. Егер екі делегат бір әдіске бағытталған сілтемені қолданатын болса, онда бір типті делегаттардың даналары бір-біріне тең деп есептеледі, мысалы,

```
public delegate void Funk1();
. . .
Funk1 d1 = Metod1;
Funk1 d2 = Metod1;
```

Осы мысалдағы d1 және d2 делегаттарының даналарын бір-біріне тең деп есептеуге болады.

13.5 Делегаттардың базалық кластарының әдістері

Айта кететін жәйт, .NET платформасындағы CTS-те абстрақтылы System.Delegate және System.MulticastDelegate кластары бар, олар арқылы құрылатын делегаттар кейбір әдістерді мұралануы мүмкін.

Құрылатын делегаттар System.MulticastDelegate –тан мұраланатын әдістер мен қасиеттердің тізімі төменде көрсетілген:

```
public MethodInfo Method {get:}– бұл қасиет делегат нұсқайтын әдістің атауын қайтарады;
```

`public object Target {get:}` – егер делегат әдісті – класс мүшесін нұсқайтын болса, онда `Target` әдіс орналасқан кластың атауын; егер әдіс статикалық болса, онда `Target null` мәнін қайтарады;

`public static Delegate Combine(Delegate[])` – делегат өңдейтін әдістер тобына жаңа әдістерді қосады;

`public static Delegate Remove(Delegate source, Delegate value)` – `source` делегатының әдістер тізімінен `value` әдісін жояды;

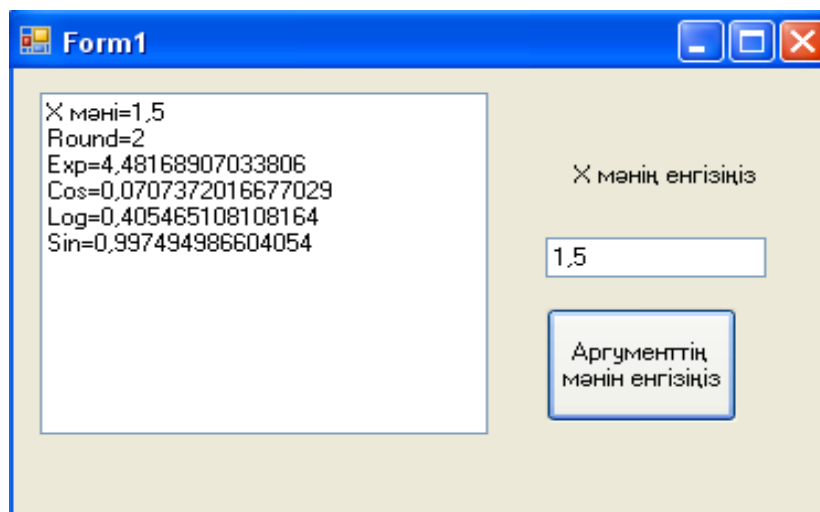
`public sealed override Delegate[] GetInvocationList()` – делегатпен байланысқан барлық әдістерді қайтарады.

Делегаттар үшін көпадресті делегат деген ұғым бар – саны бойынша кез келген болатын функцияларды нұсқайтын делегат. C# тілінің барлық делегаттары `System.MulticastDelegate`-тің туынды класы болғандықтан, C# тіліндегі кез келген делегат көп адресті делегат болып келеді.

Нақты типтегі функцияларды есептеуді орындайтын бағдарламаның делегаты үшін «көп адресті» қасиетін қолданаық. Бағдарлама коды батырманы басу өңдеуішінде жазылған, сондықтан кодтың осы үзіндісі ғана көрсетілген:

```
private void button1_Click(object sender, EventArgs e)
{
    string st;
    Funk[] ff = { Math.Sin, Math.Log, Math.Cos, Math.Exp,
Math.Round };
    string[] sfu = { "Sin", "Log", "Cos", "Exp", "Round" };
    double x, y;
    textBox2.Text = "";
    x = Convert.ToDouble(textBox1.Text);
    st = "X мәні=" + textBox1.Text + "\r\n";
    textBox2.AppendText(st);
    Funk df = Math.Sin;
    for (int i = 1; i < 5; i++)
    df += ff[i]; //df = df + ff[i];
    for (int i = 4; i >= 0; i--)
    {
        y = rab(df, x);
        df -= ff[i]; //df = df - ff[i];
        st = sfu[i] + "=" + y.ToString() + "\r\n";
        textBox2.AppendText(st);
    }
}
```

Қосымша жұмысы 13.2-суретінде көрсетілген.



13.2-сурет –Көпадресті делегатты қолдану

13.2-суретте көрсетілгендей «көп адресті» делегаттың жұмысы стектің принципі бойынша ұйымдастырылған.

Делегаттар әдістерге арналған сілтемелерді сақтау үшін арнайы класс ретінде қолданылады. Айта кететін жәйт, кез келген объект (санымен қатар делегат объектісі) сілтеме түрінде болады және оны кейбір әдістерге параметр ретінде жазуға болады.

Осылайша функционалды параметрлеу қамтамасыз етіледі: әдістерге тек қана деректерді ғана емес, сонымен қатар оларды өңдейтін түрлі әдістерді беруге болады.

13.6 Өзін-өзі тексеру сұрақтары

- 1 Делегат ұғымы.
- 2 Қандай класс функционалды тип деп аталады?
- 3 Не нәрсе делегат типіндегі кластың данасы болып табылады?
- 4 Делегатты жариялағанда қандай қызметтік сөз қолданылады?
- 5 Делегатты шақыру параметрі ретінде қандай әдісті немесе функцияны қолдануға болады?
- 6 Делегатты шақыру параметрлеріне сілтемелер қай уақытта құрылады?
- 7 Делегаттардың типтері қай уақытта үйлесімді болады?
- 8 Делегаттардың даналары қай уақытта үйлесімді болады?
- 9 Делегаттарды құрған кезде .NET платформасының қандай абстрактлы кластарын мұралауға болады?
- 10 Көпадресті делегат ұғымы.

