

3 C# ТІЛІНІҢ ГРАФИКАЛЫҚ ИНТЕРФЕЙСІ

3.1 System.Drawing атаулар кеңістігі

Microsoft .NET Framework шеңберінде жұмыс жасауға арналған C# (GDI+) қосымшасының графикалық интерфейсі атаулар кеңістігінде біріктірілетін кластар жиынынан тұрады. C# тілінің GDI+ негізгі атаулар кеңістігі болып System.Drawing есептеледі.

Осы атаулар кеңістігіндегі кластар «сурет салуға» арналған объектілер мен құралдар тізбегін анықтайды.

System.Drawing атаулар кеңістігіндегі ең жиі қолданылатын класс Brush класы болып есептеледі (Brushes, SolidBrush, т.б.). Brush объектісі (бояу жаққыш) геометриялық фигураның ішкі аумағын толтыру үшін қолданылады.

Brush типі — абстрактілі базалық класс, қалған типтер Brush типінен туындайды және басқа көптеген мүмкіндіктерді ұсынады.

Pen (Pens, SystemPens). Pen (қаламұш) — класс объектісі, оның көмегімен түзу және қисық сызықтарды сызуға болады. Pen класында статикалық қасиеттер жиыны анықталған (мысалы, түсін орнату)

Font (FontFamily). Font типіндегі объектілер қаріп сипаттамаларын анықтайды (атауы, өлшемі, жазылуы, т.б.). FontFamily бір топқа жататын, бірақ өзгешеліктері бар қаріптер жиынын ұсынады.

Graphics. Осы класс монитор экранына геометриялық фигураларды, суреттерді, мәтіндерді шығару үшін қасиеттер мен әдістер жиынын анықтайды.

Region. Бұл класс геометриялық фигураның алатын орнын анықтайды.

Point (PointF). Осы құрылымдар нүкте координаттарымен жұмыс жасауды қамтамасыз етеді. Point int типіндегі мәндермен, ал PointF – float типіндегі мәндермен жұмыс істейді.

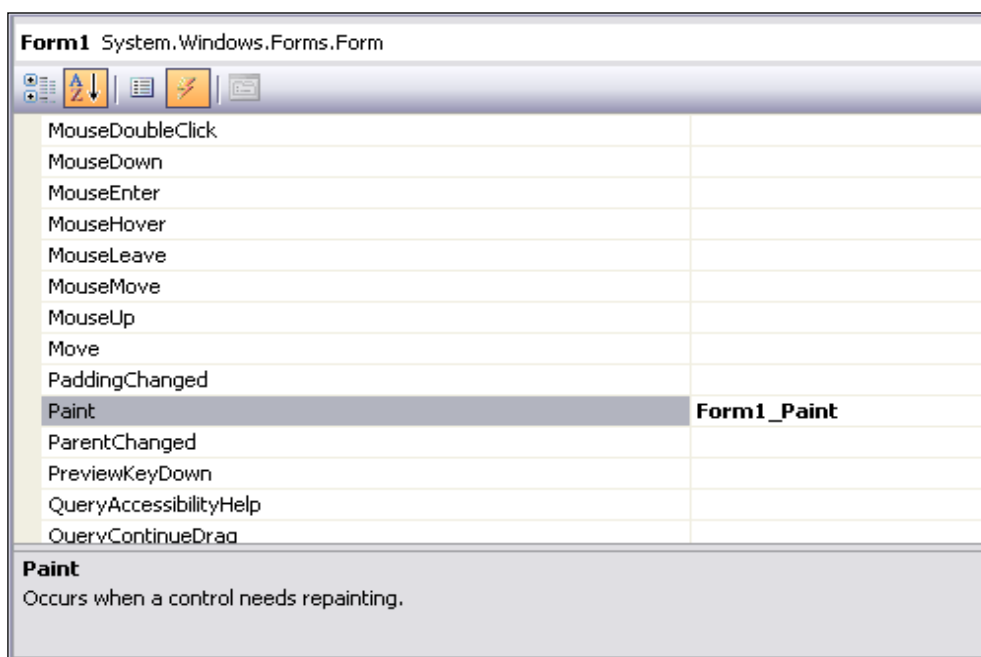
System.Drawing атаулар кеңістігінде Icon, Image, Color, Bitmap кластары және басқа да кластар бар.

3.2 Graphics класы

C# тілінде «сурет салудың» негізгі класы Graphics класы болып есептеледі. Ол бағдарламаның формасында графикалық ақпаратты шығаруға арналған. Қосымша терезеде бір бейнені салу үшін Graphics класының объектісін қолдануы немесе құруы керек. Одан кейін осы объектінің әдістері мен қасиеттерін пайдалана отырып, қосымша терезесінде түрлі фигураларды, мәтіндерді салу мүмкін болады.

Қосымшада Graphics класының объектісін құрамас бұрын «сурет салу» бойынша оқиғалар өңдеуішін анықтап алу керек.

Windows жүйесінде терезенің орын ауыстыруы мен өлшемінің өзгеруін арнайы WM_PAINT хабары «бақылайды». Ол қосымшаны керекті уақытта терезені қайта салу қажеттігі жөнінде хабардар етіп отырады. Тереземен кез келген жұмыс – оның монитор экранында орын ауыстыруы, өлшемдерінің өзгеруі, т.б. Windows жүйесінің терезе «суретін қайта салу» талабымен қатар жүреді. Қосымшада WM_PAINT оқиғалар өңдеуші осындай хабарды алғаннан кейін терезе емесе оның бөлігінің суретін қайта салуы тиіс. Форманың WM_PAINT оқиғалар өңдеушісін алдын ала дайындау үшін форма терезесінің қасиеттер терезесінде PAINT пунктін тышқанмен екі рет басу керек (3.1-сурет).



3.1-сурет – WM_PAINT өңдеушісін құру

Нәтижесінде WM_PAINT оқиғалар өңдеушісі құрылады (Form1_Paint – 3.1-сурет). Бұл өңдеуші терезе бейнесін қайта салу керек болған жағдайларда әрдайым орындалады.

Paint оқиғасының өңдеушісі мына түрде құрылады:

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
}
```

Form1_Paint өңдеушісіне екі параметр жіберіледі. Бірінші параметр арқылы оқиғаны туындатқан объектке сілтеме жіберіледі, біздің жағдайымызда сілтеме Form1 формасына болады (қай жерде сурет салу керек).

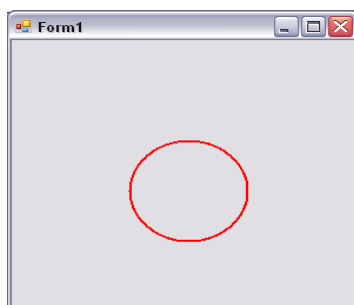
Екінші параметрді қарастыратын болсақ, осы параметр арқылы PaintEventArgs класының объектісіне сілтеме жіберіледі. Ол объект оқуға ғана арналған ClipRectangle қасиетіне ие. ClipRectangle қасиеті арқылы облыстың шекаралары беріледі және оны Paint оқиғасының өңдеушісі қайта

салуы керек. Осы шекаралар Rectangle класының объекттері ретінде беріледі. Осы кластың Left, Right, Width, Height қасиеттері, облысының орналасу орны мен өлшемін анықтауға көмектеседі. Paint оқиғасының өңдеуші ClipRectangle қасиетін ескермей, терезені толық қайтадан салады. Мысалы,

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Pen myPen = new Pen(Color.Red, 2);
    Graphics g = e.Graphics;
    g.DrawEllipse(myPen, 100, 100, 100, 100);
}
```

Осы мысалда қызыл түсте, 2 пиксельдегі қалыңдықта салу үшін myPen объектісі – «перо» құрылады. Қайта салынатын облыс (бүкіл форма) үшін Graphics типіндегі g объектісі құрылады. Назар аударыңыз, new қолданылмайды. Одан кейін g объектісі үшін эллипсті салу әдісі іске қосылады.

Graphics типіндегі g объектісінің ерекшелігі – объектің монитор құрылғысының контекстімен нұсқауыш болып келуі (компьютер бейнекартасының драйверімен қосымшаны байланыстыратын Windows жүйесінің арнайы бағдарламалары). Құрылғылар контекстерінің көмегімен Windows жүйесі қосымшалар мен компьютер құрылғыларының драйверлер үйлесімділігін қамтамасыз етеді, мысалы бейнекарта типіне тәуелсіз бағдарлама коды өзгеріссіз қалады, ал бейнекартаны басқару бойынша барлық мәселелерді монитор құрылғысының контексі шешеді. Бағдарламаның жұмысы 3.2-суретте көрсетілген.



3.2-сурет – Эллипсті шығару

C# тіліндегі басқа кластар сияқты Graphics класының қасиеттері мен әдістері бар. Олардың кейбірін қарастырып өтейік.

Clear() – бұл әдіс Graphics объектісін пайдаланушы таңдап алған түске бояйды.

Кейбір геометриялық фигураларды «салуға» арналған әдістердің үлкен тобы: DrawArc(), DrawBezier(), DrawBeziers(), DrawCurve(), DrawEllipse(), DrawIcon(), DrawLine(), DrawLines(), DrawPie(), DrawPath(), DrawRectangle(), DrawRectangles(), DrawString().

Геометриялық фигуралардың ішкі облыстарын толтыру үшін алдында Fill сөзі бар әдістер қолданылады, мысалы, FillPie(), FillEllipse() немесе FillRectangle().

Кейбір әдістердің жұмысын «Визуальное проектирование приложений С#» кітабынан (авторлары А.В. Фролов, Г.В. Фролов. 10-бөлім) оқи аласыз [3]. Ескеру керек, облыстарды қайта салуға арналған Graphics типіндегі объекті қолдану үшін Form1_Paint оқиғалар өңдеушісіне тиісті PaintEventArgs класының объектісінен бөлек форма мен басқару элементтерінің кластарында сипатталатын CreateGraphics әдісін қолдануға болады (). Мысалы,

```
Graphics g = this.CreateGraphics(); немесе  
Graphics g = Graphics.FromHwnd(this.Handle);
```

Графикалық объекті Image объектісі арқылы құруға болады. Осы тәсіл берілген суретті (кескінді) өзгерту үшін қолданылады, мысалы,

```
Bitmap bm = new Bitmap( "d:\\picture.bmp" );  
Graphics g = Graphics.FromImage( bm );
```

3.3 Бағдарламаны жүзеге асыру мысалы

Мысал ретінде монитор экранына граф типіндегі құрылымды шығару бағдарламасын қарастырайық.

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();  
        }  
        public static int p = 0;  
        private void Form1_Paint(object sender, PaintEventArgs e)  
        {  
            Graphics g = e.Graphics;  
            if (p == 0)  
            {  
                String st;  
                int i, j;  
                int xc, yc;  
                //g.Clear(Color.White);  
                int[,] a = new int[9, 9]  
                {{0, 6, 1000, 4, 1000, 1000, 1000, 1000, 1000},
```

```

        {6, 0, 5,1000,3,6,1000,1000,1000},
        {1000,5,0,5,1000,3,1000,1000,1000},
        {4,1000,5,0,1000,1000,4,1000,1000},
        {1000,3,1000,1000,0,7,1000,7,1000},
        {1000,6,3,1000,7,0,6,4,1000},
        {1000,1000,1000,4,1000,6,0,9,6},
        {1000,1000,1000,1000,7,4,9,0,8},
        {1000,1000,1000,1000,1000,1000,6,8,0}};
int[,] V = new int[9, 2]
    {{200, 40},
     {100, 80},
     {200, 100},
     {280, 60},
     { 60, 140},
     {200, 160},
     {340, 140},
     {160, 220},
     {360, 260}};
e.Graphics.Clear(Color.Bisque);
Pen myPen = new Pen(Color.Red, 2);
for (i = 0; i < 9; i++)
for (j = 0; j < 9; j++)
if ((a[i, j] != 0) && (a[i, j] != 1000))
{
g.DrawLine(myPen, V[i, 0], V[i, 1], V[j, 0], V[j, 1]);
xc = (int)(V[i, 0] + V[j, 0]) / 2;
yc = (int)(V[i, 1] + V[j, 1]) / 2;
st = Convert.ToString(a[i, j]);
g.DrawString(st, new Font("Times new Roman",8),
Brushes.Blue, xc - 6, yc - 15);
}
for (i = 0; i < 9; i++)
{
g.FillEllipse(Brushes.White, V[i, 0] - 12, V[i, 1] - 12,
25, 25);
st = Convert.ToString(i);
g.DrawString(st, new Font("10_IC_1", 12), Brushes.Black,
V[i, 0] - 6, V[i, 1] - 6);
}
}
if (p == 1)
{
g.Clear(Color.White);
g.DrawRectangle(new Pen(Brushes.Blue, 2), 5, 5, 380, 270);
Pen myPen = new Pen(Color.Black, 2);
Point[] myPoints =
{
new Point(10, 10),
new Point(100, 40),
new Point(150, 24),
new Point(100, 100),
};
g.DrawPolygon(myPen, myPoints);
}

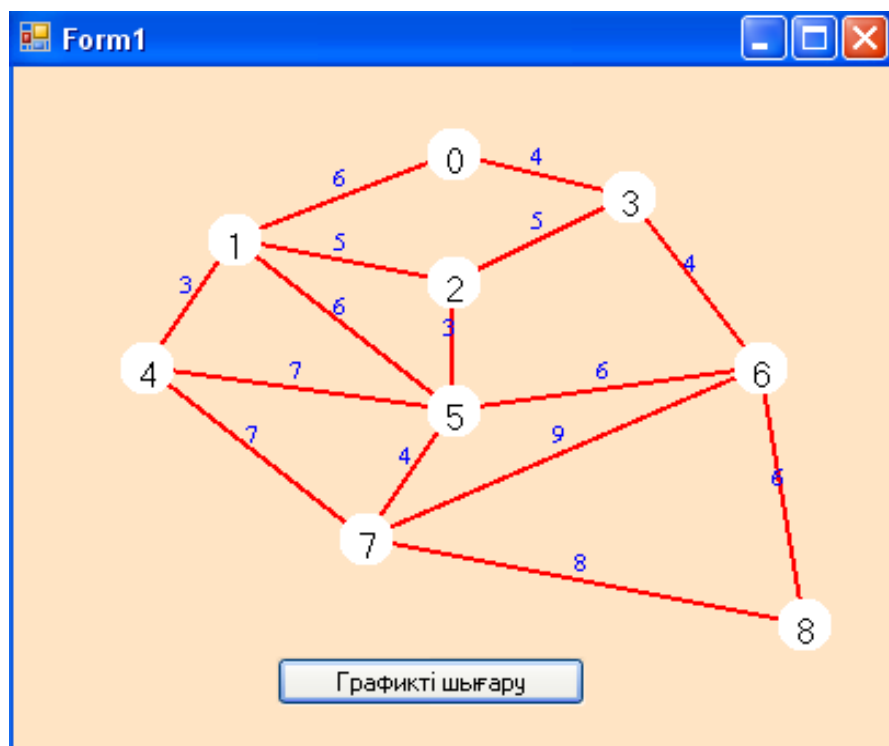
```

```

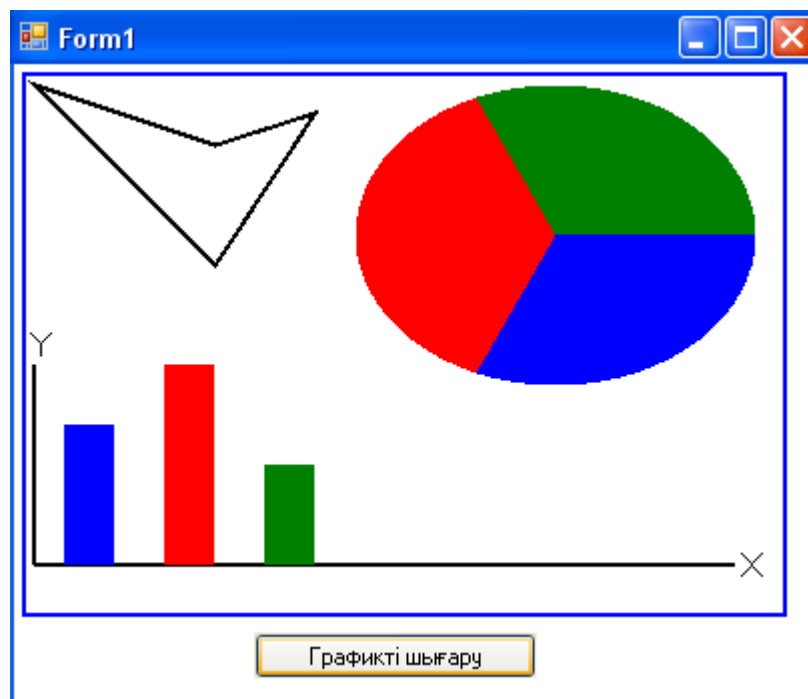
g.DrawLine(myPen, 10, 250, 360, 250);
g.DrawLine(myPen, 10, 250, 10, 150);
g.DrawString("Y", new Font("10_IC_1", 12), Brushes.Black,
5,130);
g.DrawString("X", new Font("10_BT_1", 12), Brushes.Black,
360,240);
g.FillRectangle(Brushes.Blue, 25, 180, 25, 70);
g.FillRectangle(Brushes.Red, 75, 150, 25, 100);
g.FillRectangle(Brushes.Green, 125, 200, 25, 50);
g.FillPie(Brushes.Blue, 30, 10, 200, 150, 0, 120);
g.FillPie(Brushes.Red, 30, 10, 200, 150, 120, 120);
g.FillPie(Brushes.Green, 30, 10, 200, 150, 240, 120);
}
}
private void button2_Click(object sender, EventArgs e)
{
if (p == 0) p = 1; else p = 0;
this.Invalidate();
}
}
}

```

Бағдарлама жұмысы 3.3 және 3.4 суреттерінде көрсетілген.



3.3-сурет – Графты шығару



3.4-сурет – Түрлі графикалық фигураларды шығару

Бұл мысал «Визуальное проектирование приложений С#» кітабынан алынған (авторлары - А.В. Фролов, Г.В. Фролов) [3].

Бағдарламада ақпаратты шығаруды басқару глобальді р айнымалысы арқылы орындалады. Егер $p = 0$ (бағдарлама іске қосылғанда) болса, онда форма терезесіне графтың суреті шығады. Форма терезесіне «Графикті шығару» батырмасын басқанда p мәні қарама-қарсы мәнге ауысады. Егер $p = 0$ -ге тең болса, онда ол 1-ге тең болады және керісінше.

Батырманы басу оқиғасының өңдеуішінде ағымдағы объектке арналған әдіс бар - `this.Invalidate()`; (`this` нұсқағышы бағдарлама жұмысында қолданылатын ағымдағы объект адресін сақтайды,). Қарастырылған мысалда ағымдағы объект болып форма есептеледі. `this.Invalidate()`; әдісі Windows операциялық жүйесінен форма үшін `WM_PAINT` хабарламасын дайындауды талап етеді. Бағдарлама осы хабарды алғаннан кейін `private void Form1_Paint(object sender, PaintEventArgs e)` оқиғасының өңдеуішін іске қосады, яғни P айнымалысының мәні жаңа болса форма терезесі қайтадан салынады. Сонымен, Windows операциялық жүйесінің `WM_PAINT` хабарын қосымшадан «алуға» («формировать») болады.

`Form1.cs` файлының кодын қарастырайық. Бірінші бөлім граф суретінен тұрады. Қосымша граф төбелерінің координаттар массиві – V қосылған. Алдын ала граф қағаз бетінде салынып, координаттардың мәндері суреттен алынған. Форма терезесін тазартамыз – `e.Graphics.Clear(Color.Bisque);` - `Color.Bisque` түсімен бояймыз. Қызыл түсті таңдаймыз, сызықтың қалыңдығы 2 пиксельге тең болады - `Pen myPen = new Pen(Color.Red, 2);`. Граф төбелерінің арасында сызықтарды салу циклін іске қосамыз. Граф төбелерінің

арасындағы орташа нүкте үшін бір уақытта *x* және *y* координаттары есептеледі. Осы координаттарды қолдана отырып, сыбайлас матрицалардан алынған оның мәнін қабырға сызығының үстіне шығарамыз. `g.DrawString` жазбасының пішімі бойынша мәтінді шығарған кезде қаріп типін, өлшемін көрсету керек. Келесі циклде граф төбелері салынады және жазылады. `g.FillEllipse` әдісі тіктөртбұрышты аймаққа іштей сызылған эллипсті бояйды, оған оның орналасу орны мен өлшемдері параметрлер арқылы беріледі.

Форманың «екінші» терезесінде терезені ақ түспен «тазарту» орындалды. Мысал ретінде көрсету мақсатында көпбұрыш, *X* пен *Y* координаттарының осінің сызығы, тіктөртбұрыштар (гистограмма) жиыны және шеңберлі диаграммалар «салынды».

`g.DrawPolygon(myPen, myPoints);` әдісі көпбұрышты салуға көмектеседі, олардың төбелерінің координаттары массив арқылы анықталады. Әдістің бірінші параметрі сызықтың түсін анықтайды.

`g.FillRectangle` әдісі боялған тіктөртбұрыштарды салуға мүмкіндік береді, тіктөртбұрыштың сол жақ төбесіндегі координаттары, жалпақтығы, ұзындығы берілген. Бірінші параметр ретінде тіктөртбұрыштың ішкі аумағын бояу түсі беріледі.

Шеңберлі диаграмма `g.FillPie` әдісі арқылы салынады, ол «боялған» сегменттерді көрсетуге арналған. Бірінші параметр ретінде әдіске сегменттің ішкі аумағын «бояу» түсін жіберу керек. Келесі төрт параметрлер іштей сызылған эллипсі бар тіктөртбұрыштың орналасуын және өлшемін анықтайды. Соңғы екі параметр эллипс сегментін шектейтін бұрыштарды анықтайды: бастапқы бұрышты және оның сағат тілі бағытындағы өсімшесін.

3.4 Өзін-өзі тексеру сұрақтары

- 1 C# тілінде GDI+ нені білдіреді?
- 2 Brush класының объектісі нені анықтайды?
- 3 Pen класының объектісі нені анықтайды?
- 4 Graphics класы нені анықтайды?
- 5 Windows жүйесінде қандай хабарлама терезе өлшемінің өзгеруін және терезе орнының ауысқанын «бақылайды»?
- 6 Қосымша формасының терезесін қайта салу өңдеуіш қалай аталады?
- 7 Form1_Paint өңдеуішінің бірінші формалды параметр нені анықтайды?
- 8 Form1_Paint өңдеуішінің екінші формалды параметрі нені анықтайды?
- 9 Монитор құрылғысының контекст ұғымы.

10 Қандай әдіс Windows операциялық жүйесінен форма үшін WM_PAINT хабарын дайындауды талап етеді?

