

## 5 ДИАЛОГТЫҚ МЕНЮДІ ҚОЛДАНУ

### 5.1 Матрицамен жұмыс жасауға арналған оқиғалар өңдеуіштері

Қосымшада менюмен және инструменттік панельмен жұмыс жасауды жалғастыра отырып бізге 6\*6 матрицасын құруға және оны қосымша терезесіне шығаруға арналған хабарламалар өңдеуіштерінің кодын жазу керек.

Матрица құрылғаннан кейін өңдеуіш жұмысының сәтті аяқталғаны туралы хабарламаны экранға шығару керек, мысалы, «Матрица құрылды».

«Матрицаны шығару» әдісінің нәтижесінде матрица экранға шағады, сондықтан қосымша хабарламаны шығарудың қажеті жоқ.

Form1.cs файлының коды үзінділер бойынша қарастырылады.

Form1.cs файлының коды:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public static int[,] a = new int[6, 6];
        public Form1()
        {
            InitializeComponent();
        }
        private void sozMToolStripMenuItem_Click(object sender,
            EventArgs e)
        {
            Random rnd = new Random();
            for (int i = 0; i < 6; i++)
            for (int j = 0; j < 6; j++)
            a[i, j] = rnd.Next() % 90 + 10;
            richTextBox1.AppendText("Matrisa kyrildi \n");
        }
        private void printToolStripMenuItem_Click(object sender,
            EventArgs e)
        {
            string ss;
            richTextBox1.AppendText("\n");
            for (int i = 0; i < 6; i++)
            {
                ss = "";
                for (int j = 0; j < 6; j++)
                ss = ss + Convert.ToString(a[i, j]) + "\t";
                richTextBox1.AppendText(ss + "\n");
            }
        }
    }
}
```

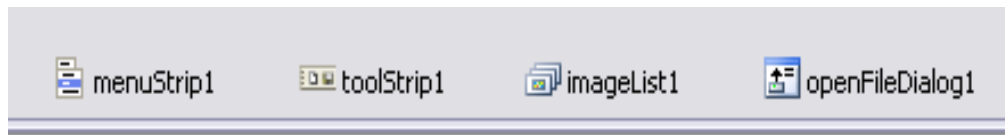
```
}  
}
```

Басқа оқиғалар өңдеуіштерінің коды әзірше бос.

## 5.2 Файлды ашуға арналған оқиғалардың өңдеуіштері

Инструменттік панельде файлды ашу батырмасы – LoadT командасы бар. Осы оқиға өңдеуішін жазу бойынша әрекеттердің ретін қарастырайық.

Қосымшада файлды ашу бойынша оқиға өңдеуішін жазу үшін бізге OpenFileDialog элементі керек болады. Осы элементтің таңбашасын Toolbox терезесінен көшірейік. OpenFileDialog элементінің таңбашасы форма астындағы панельде пайда болады. (5.1-сурет).



5.1-сурет –OpenFileDialog элементін қосу

OpenFileDialog элементін қолдану бағдарламалаудың бүтін бір технологиясы болып келеді, ол компьютердегі бумаларды, дисктерді, файлдарды қолдануға мүмкіндік береді. Осы элемент Toolbox терезесіндегі кез келген элемент сияқты файлдармен жұмыс жасаудың көптеген әдістері бар класс ретінде берілген, мысалы, openFileDialog1.ShowDialog әдісі экранға файлды таңдаудың стандартты диалогтық терезесін көрсетеді (5.2-сурет).

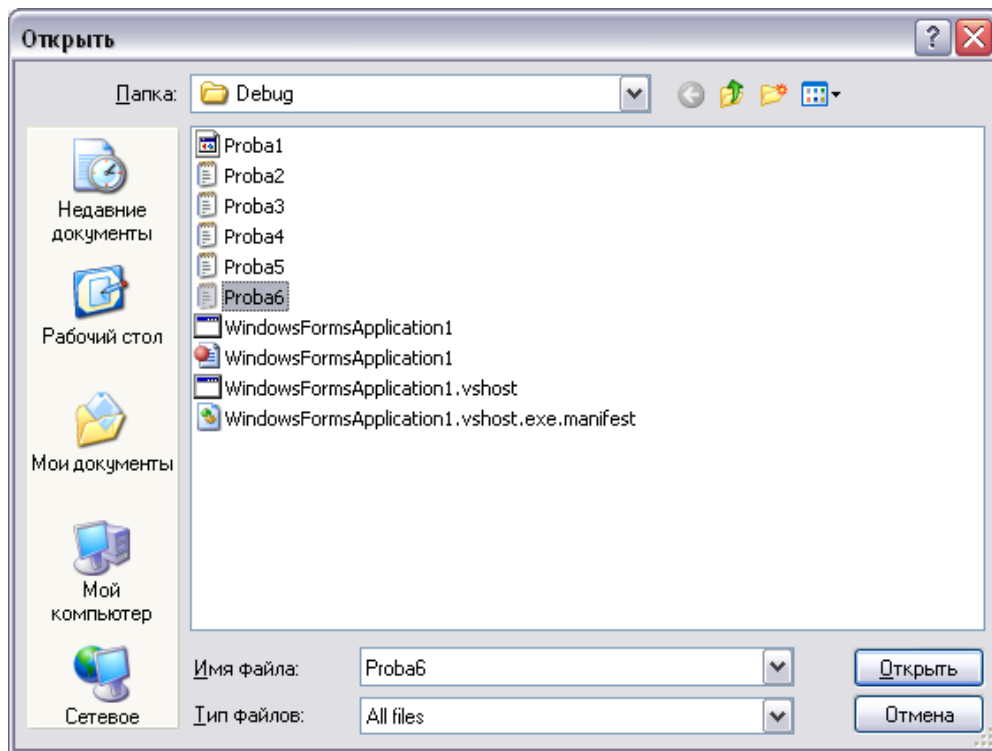
openFileDialog1 элементінің қасиеттерін күйге келтіре отырып ашылатын файлдардың атауларына фильтрді анықтауға болады. Ол үшін openFileDialog1 элементінің Properties терезесінде Filter қасиетін өзгерту керек. Осы қасиетке келесі мәтіндік жолды меншіктеңіз:

```
Text files|*.txt|RTF files|*.rtf| All files|*.*
```

Фильтрдің жолы «|» символымен бөлінген бөліктерден тұрады. Бірінші бөлік файл типінің атауын Text files деп белгілейді, қалған бөліктерде файл атауларының қалқасы (маскасы) анықталған. Мәтіндік файлдар үшін \*.txt қалқасы (маскасы) қолданылады.

Одан кейін RTF files форматы орналасқан. RTF файлдарна \*.rtf қалқасы (маскасы) қолданылады.

Қосымшада барлық типтегі файлдар ашылу үшін (All files) \*.\* маскасы (қалқа) қолданылады (мысал 5.2-суретте көрсетілген).



5.2-сурет – Диалогтық терезені ашу бойынша бағдарлама жұмысы

Есептің шарты бойынша біз мәтіндік файлдармен жұмыс істейтін боламыз, бірақ басқа типтегі файлдарды да ашып көруге болады.

Қосымша файлының коды толықтырылды:

```
private void loadToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK &&
        openFileDialog1.FileName.Length > 0)
    {
        try
        {
            richTextBox1.LoadFile(openFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
        }
        catch (System.ArgumentException ex)
        {
            richTextBox1.LoadFile(openFileDialog1.FileName,
RichTextBoxStreamType.RichText);
        }
        this.Text = "Файл [" + openFileDialog1.FileName + "];
    }
}
```

Егер файлды таңдау терезесінде пайдаланушы «Открыть» батырмасын басқан болса, онда ShowDialog әдісі DialogResult.OK мәнін қайтарады. Шартқа файл таңдалғаны жөнінде қосымша тексеру енгізілген (таңдап алынған файлға дейінгі жолдың ұзындығы – openFileDialog1.FileName.Length 0-ден үлкен болуы керек).

Файлды richTextBox1.LoadFile әдісімен ашу барысында оған екі параметр жіберіледі. Бірінші параметр – файлға дейінгі жолды, ал екінші параметрге файл типін анықтаймыз.

Мәтіндік файлдармен жұмыс жасау керек болғандықтан, негізгі тип RichTextBoxStreamType.PlainText болып табылады.

Егер тандап алынған файлдың форматы LoadFile әдісіндегі PlainText форматынан өзге болса, онда System.ArgumentException ерекше жағдайы (исключение) туындайды. Өңдеуіш файлды RichText типінде файл ретінде қайтадан ашуға әрекеттенеді. Бұл тип RTF форматына сәйкес. Қосымша ақпарат үшін ашылған файлдың толық жолы қосымшаның негізгі терезесінің бас жағында көрсетіледі.

### **5.3 Файлға жазу бойынша оқиғалардың өңдеуіштері (Файлға жазу үшін қолданылатын оқиғалардың өңдеуіштері)**

Инструменттік панельде кезек бойынша келесі болып қосымшаның терезесіндегі жазбаларды файлға жазуға арналған батырма (SaveT командасы) орналасқан. Осы оқиға өңдеуішін жазу бойынша әрекеттердің ретін қарастырайық.

Қосымшада файлға жазуды орындайтын өңдеуішті құру үшін бізге SaveFileDialog элементі қажет болады. Осы элементтің таңбашасын Toolbox терезесінен форма терезесіне көшірейік. SaveFileDialog1 элементінің таңбашасы форманың астында орналасқан панельде пайда болады.

SaveFileDialog1 элементінің қасиеттерін күйге келтіре отырып, оның Filter және FileName қасиеттерін редакциялау керек.

Filter қасиетіне мәтіндік файлдың типін көрсету керек – Text files|\*.txt. Ал FileName қасиетіне құжат атауын – doc1.txt мәнін меншіктеу керек. Файлдар осы атаумен сақталады.

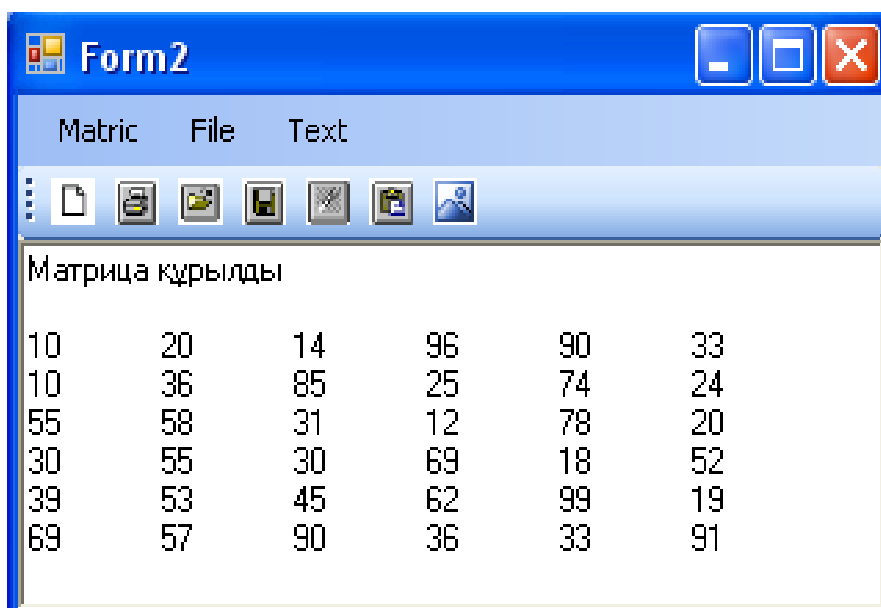
Қосымша файлының коды толықтырылды:

```
private void saveTToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == DialogResult.OK &&
        saveFileDialog1.FileName.Length > 0)
    {
        richTextBox1.SaveFile(saveFileDialog1.FileName,
            RichTextBoxStreamType.PlainText);
        this.Text = "Файл [" + saveFileDialog1.FileName + "];"
    }
}
```

Файлдың толық жолы қосымшаның негізгі терезесінің бас жағында көрсетіледі.

richTextBox1.SaveFile әдісінің екінші параметрі сақталатын файлдың типін анықтайды. Қарастырылған мысалда бағдарлама тек мәтіндік файлдармен жұмыс істеуге бағытталған. Егер осы параметр арқылы

RichTextBoxStreamType.RichText мәні жіберілсе, онда құжат RTF форматында сақталады.



5.3-сурет – Қосымшада мәтіндік файлмен жұмыс жасау мысалы

#### 5.4 Мәтінмен жұмыс істеуге арналған оқиғалардың өңдеуіші

Қосымшаның пайдаланушыға арналған аумағында мәтінмен жұмыс жасау үшін менюдің тазарту мен түрлендіру командаларына сәйкес екі оқиғалар өңдеуіші қарастырылған.

Мәтінді тазарту бойынша оқиға өңдеуішінің ішіне барлық жолдарды өшіруді орындайтын richTextBox1 мәтін редакторының бір әдісі жазылған:

```
private void clearToolStripMenuItem_Click(object sender,
EventArgs e)
{
    richTextBox1.Clear();
}
```

Мәтіндегі түрлендіруді жүргізетін оқиға өңдеуішісі пайдаланушыға арналған аумақта орналасқан мәтіннен 6\*6 матрицасының мәндерін (әдетте файлдан алынатын) бөліп алу үшін қолданылады.

Мәтінді өңдеу әдісінің коды төменде келтірілген:

```
private void obraboToolStripMenuItem_Click(object sender,
EventArgs e)
{
    string ss;
    string[] clova;
    int k, n;
    int[] masi = new int[100];
    n = 0;
    ss = richTextBox1.Text;
```

```

clova = ss.Split('\t', ' ');
for (int i = 0; i < clova.Length; i++)
{
k = clova[i].Length;
if (k == 2 || k == 3)
{
masi[n] = Int32.Parse(clova[i]); n++; }
richTextBox1.AppendText(i.ToString() + " = " + clova[i] +
" k= " + k.ToString() + "\n");
}
int j1, j2; j1 = j2 = 0;
for (int i = 0; i < 36; i++)
{
a[j1, j2] = masi[i];
j2++;
if (j2 == 6) { j1++; j2 = 0; }
if (j1 == 6 && j2 == 6) break;
}
}

```

Мына жерде кейбір түсініктемелерді келтіріп өту керек.

String типіндегі clova массиві жарияланды, оған пайдаланушыға арналған аумақта орналасқан мәтіннен Split әдісі арқылы табуляция, бос орын немесе қайтару таңбасы арқылы ерекшеленген символдардың кез келген тіркестері жазылады:

```

ss = richTextBox1.Text;
clova = ss.Split('\t', ' ', '\n');

```

clova.Length - мәтіндегі барлық сөздердің санын анықтайды.

Циклда ұзындығы 2 немесе 3 символға тең барлық сөздер бүтін сандарға түрлендіріледі және masi бүтін сандар массивіне жазылады. Барлық сөздер мен олардың ұзындықтары бір мезгілде экранға шығады (бақылауды жүргізу үшін).

Оқиға өңдеуішінің соңғы бөлігінде массивтегі ерекшеленген бүтін сандар 6\*6 матрицасына жазылады.

Пайдаланушы мына нәтижелерді көреді:

Матрица құрылды

```

10    20    14    96    90    33
10    36    85    25    74    24
55    58    31    12    78    20
30    55    30    69    18    52
39    53    45    62    99    5
69    57    90    36    33    91
0 = Матрица k= 7
1 = құрылды k= 7
2 =

```

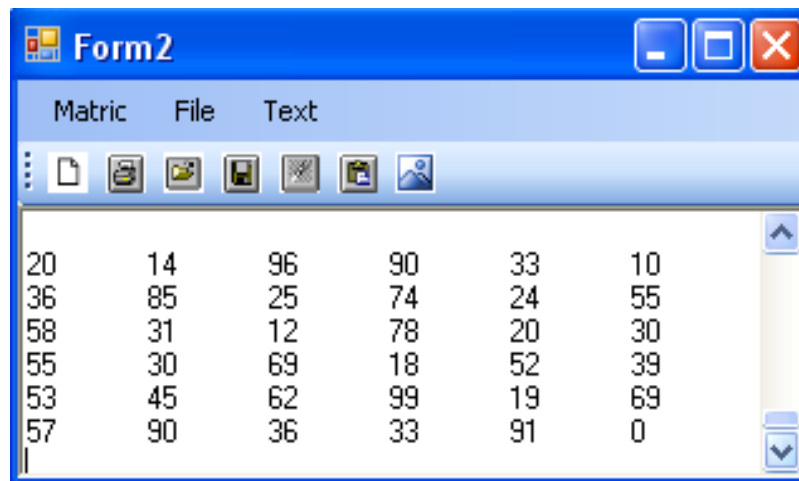
```

10 k= 4
3 = 20 k= 2
4 = 14 k= 2

```

5 = 96 k= 2  
6 = 90 k= 2  
7 = 33 k= 2  
8 =  
10 k= 3  
9 = 36 k= 2  
10 = 85 k= 2  
11 = 25 k= 2  
12 = 74 k= 2  
13 = 24 k= 2  
14 =  
55 k= 3  
15 = 58 k= 2  
16 = 31 k= 2  
17 = 12 k= 2  
18 = 78 k= 2  
5 = 20 k= 2  
20 =  
30 k= 3  
21 = 55 k= 2  
22 = 30 k= 2  
23 = 69 k= 2  
24 = 18 k= 2  
25 = 52 k= 2  
26 =  
39 k= 3  
27 = 53 k= 2  
28 = 45 k= 2  
29 = 62 k= 2  
30 = 99 k= 2  
31 = 5 k= 2  
32 =  
69 k= 3  
33 = 57 k= 2  
34 = 90 k= 2  
35 = 36 k= 2  
36 = 33 k= 2  
37 = 91 k= 2  
38 = k= 1

Пайдаланушы аумағын тексергеннен кейін оны тазартуға болады. Одан кейін матрицаны экранға шығару режимін іске қосамыз. (5.4-сурет):



5.4-сурет – Пайдаланушы аумағындағы мәтінді түрлендіргеннен кейін матрицаны экранға шығару

«Матрица құрылды» хабарламасы шығарылған жоқ.

### 5.5 Өзін-өзі тексеру сұрақтары

- 1 Диалогтық категорияларға қандай басқару элементтері жатады?
- 2 Файлды ашу бойынша диалогтық терезені құру үшін қандай басқару элементі қолданылады?
- 3 Қандай әдіс файлды ашу үшін Windows жүйесінің стандартты диалогтық терезесін экранға шығарады?
- 4 OpenFileDialog басқару элементі арқылы тек мәтіндік файлдарды ашу үшін осы элементті қалай күйге келтіруге болады?
- 5 Файлды ашу бойынша диалогтық терезеде пайдаланушы «Открыть» батырмасын басса не болады?
- 6 `if (openFileDialog1.ShowDialog() == DialogResult.OK && openFileDialog1.FileName.Length > 0)` жазбасында `openFileDialog1.FileName.Length > 0` шарты нені білдіреді?
- 7 RichTextBox1 басқару элементінің қандай әдісі файлды ашу үшін қолданылады?
- 8 SaveFileDialog басқару элементінің қызметі қандай ?
- 9 SaveFileDialog басқару элементінің FileName қасиеті нені анықтайды?
- 10 richTextBox1.SaveFile әдісінің RichTextBoxStreamType.PlainText параметрі нені анықтайды?



