

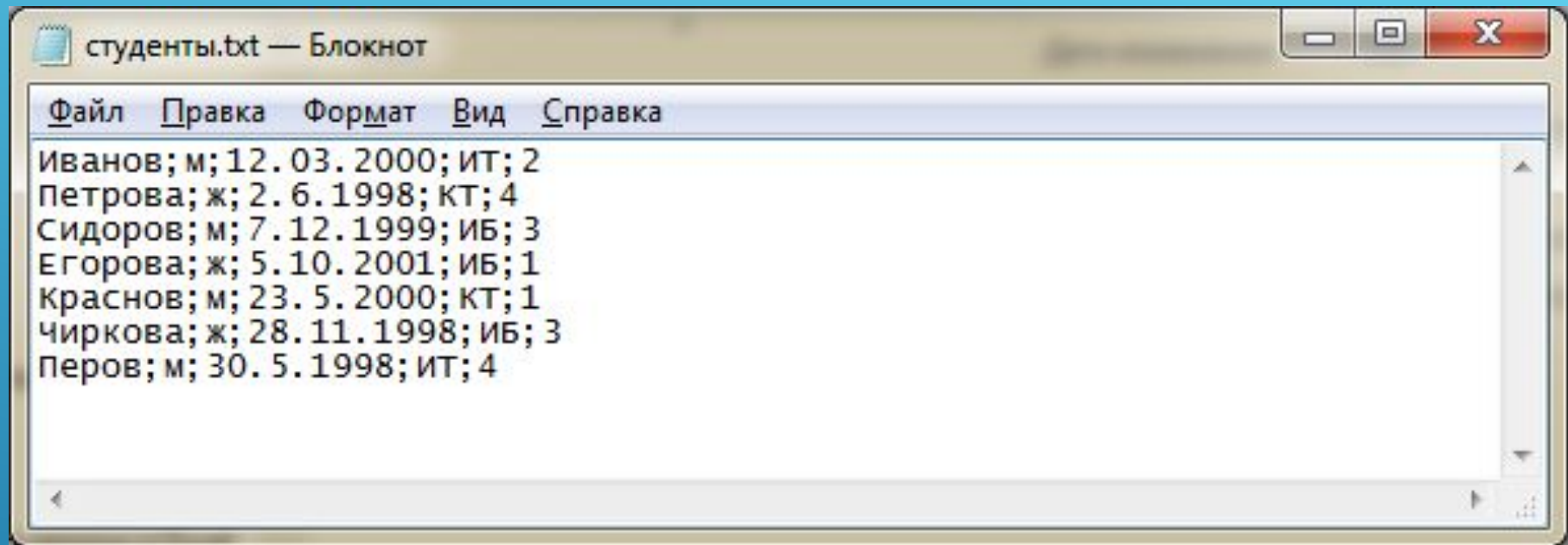
ПРОЕКТМАССИВ СТУДЕНТОВ

Чтение из файла, сортировка и отбор данных

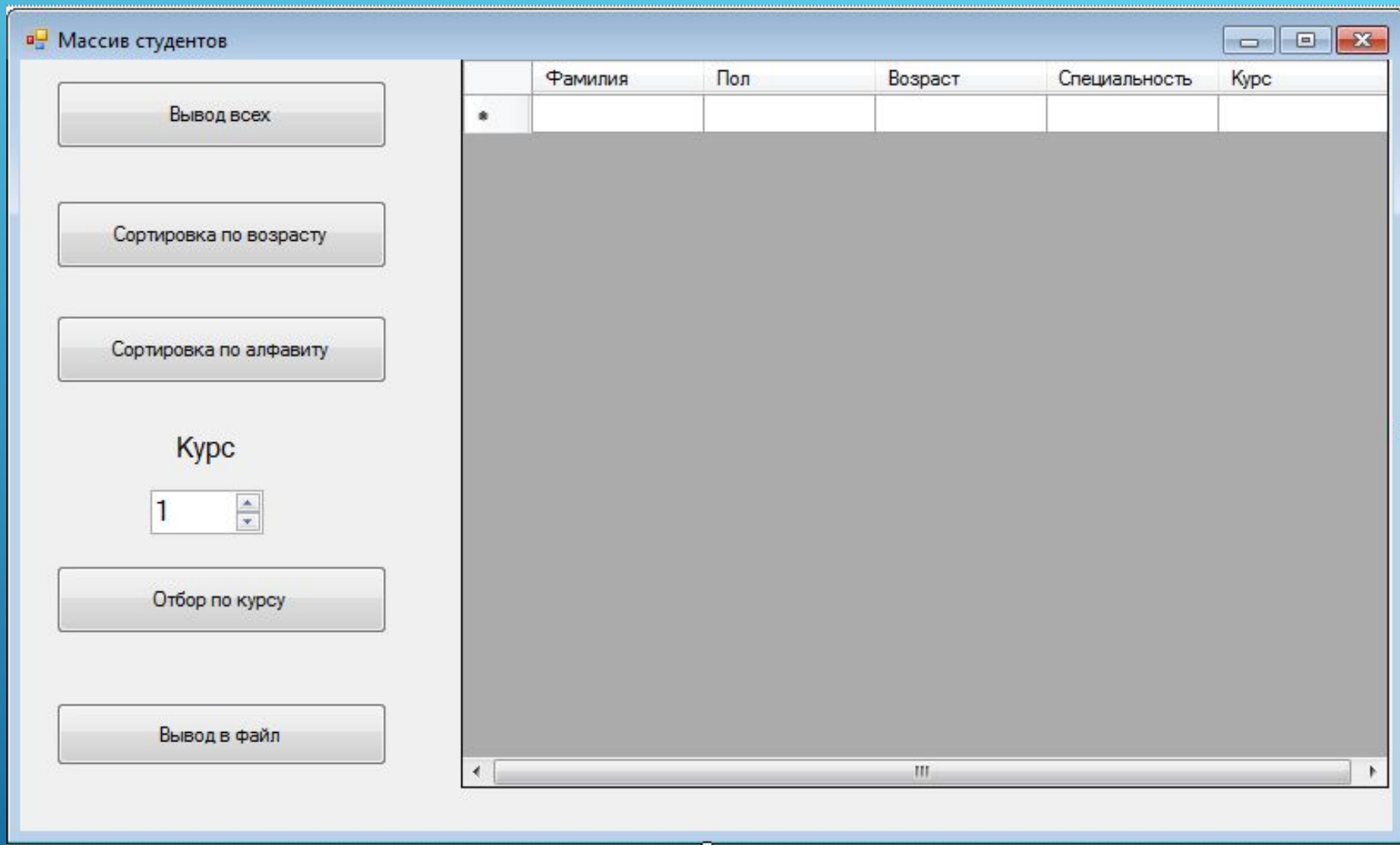
A decorative graphic consisting of several parallel white lines of varying thicknesses, slanted diagonally from the bottom-left towards the top-right, set against a blue gradient background.

```
struct student
{
    public string fam;
    public char pol;
    public DateTime d_rogd;
    public string spec;
    public byte curs;
    public byte vozr()
    {
        return Convert.ToByte((DateTime.Today - d_rogd).Days / 386.25);
    }
}
```

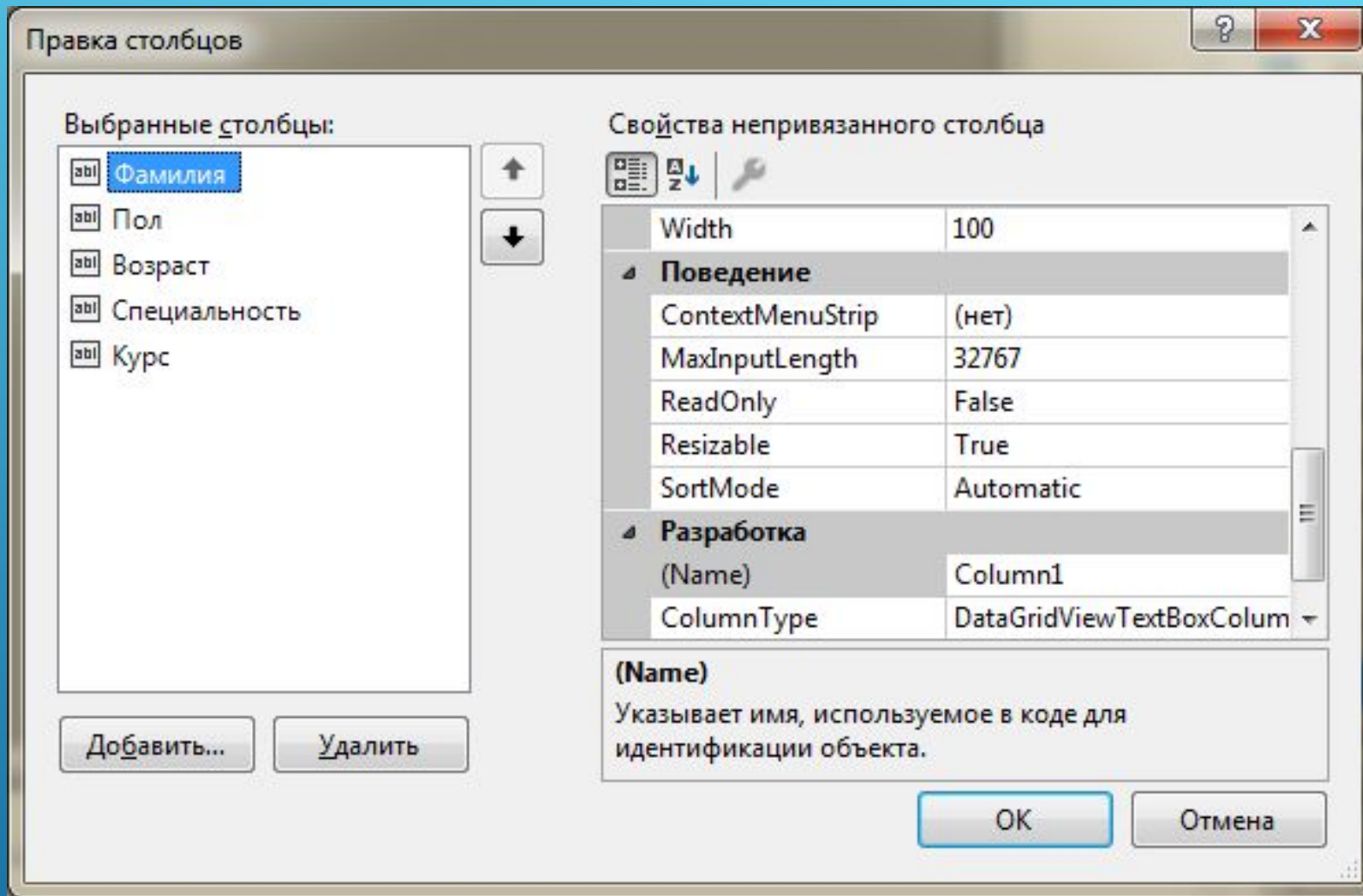
СТРУКТУРА СТУДЕНТ



Текстовый файл “студенты.txt”



ВИД КОНСТРУКТОРА ФОРМЫ



DataGridView
Свойство Columns

```
student[] massiv;  
int n;  
int k = 0;  
StreamWriter writefl; //поток для записи данных
```

Переменные формы

```
public Form1() //Начало конструктора формы
{
    string[] rows;
    InitializeComponent();
    if (File.Exists("студенты.txt"))
    {
        rows = File.ReadAllLines("студенты.txt");
        n = rows.Length;
        massiv = new student[n];
    }
    else
    {
        MessageBox.Show("Файл не найден");
        return;
    }
}
```

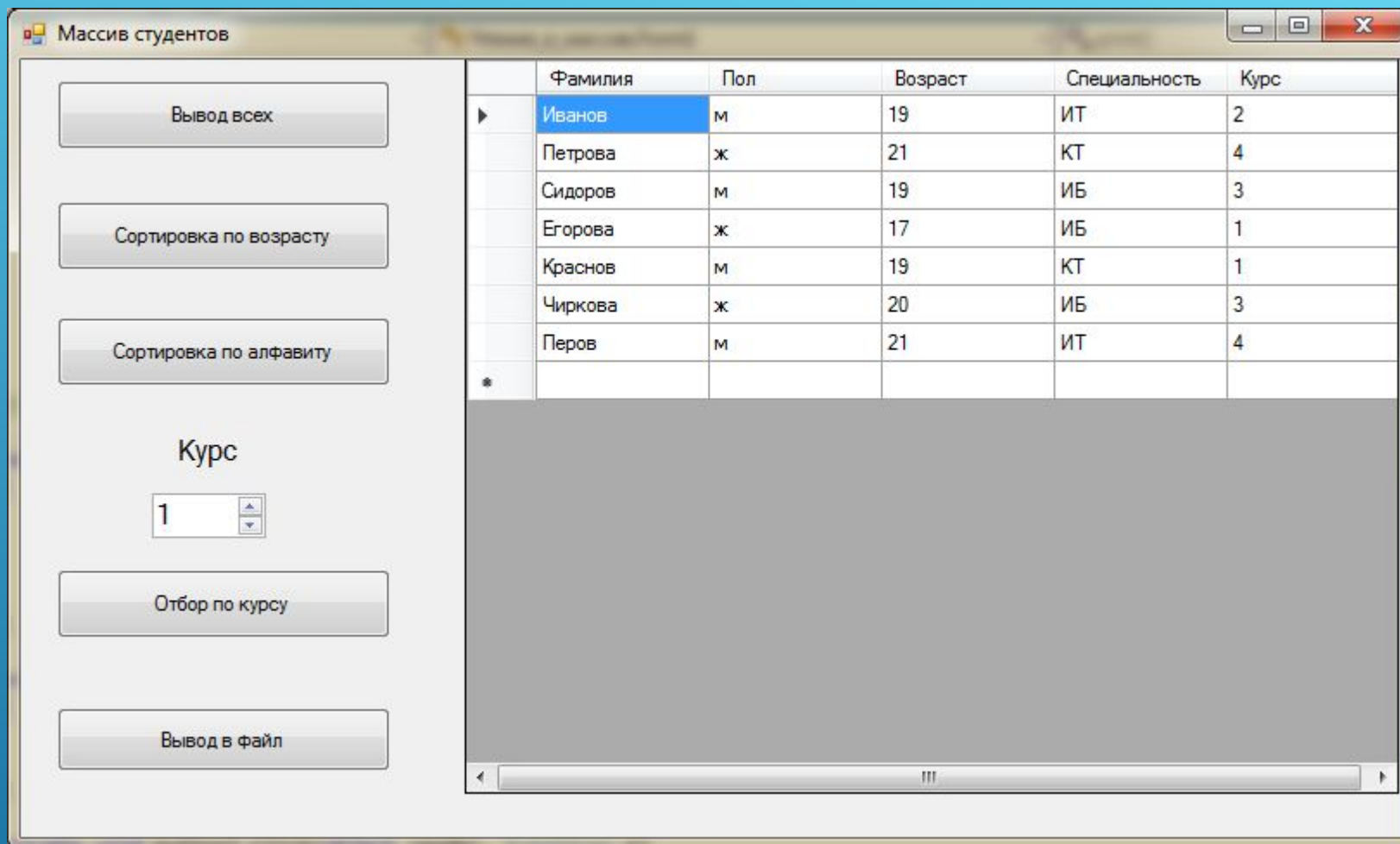
Чтение данных из файла

```
for (int i = 0; i < n; i++)
{
    string[] str = rows[i].Split(';');
    massiv[i].fam = str[0];
    massiv[i].pol = Convert.ToChar(str[1]);
    massiv[i].d_rogd = Convert.ToDateTime(str[2]);
    massiv[i].spec = str[3];
    massiv[i].curs = Convert.ToByte(str[4]);
}
print();
} // Конец Конструктора формы
```

ЗАПИСЬ СЧИТАННЫХ ДАННЫХ В МАССИВ


```
void print()
{
    dataGridView1.Rows.Clear();
    foreach (var t in massiv)
        dataGridView1.Rows.Add(t.fam, t.pol.ToString(), t.vozr().ToString(),
                                t.spec, t.curs.ToString());
}
```

Метод вывода массива в DataGridView1



Вид формы при запуске проекта

```
private void button1_Click(object sender, EventArgs e)
{
    print();
}
```

Обработчик события кнопки «Вывод всех»

```
private void button2_Click(object sender, EventArgs e)
{
    student p;
    for (int i = n-1; i >0; i--)
        for (int j = 0; j <i; j++)
        {
            if (massiv[j].vozn() > massiv[j + 1].vozn())
            {
                p = massiv[j];
                massiv[j] = massiv[j + 1];
                massiv[j + 1] = p;
            }
        }
    print();
}
```

Обработчик события кнопки «Сортировка по возрасту»

Массив студентов

Вывод всех

Сортировка по возрасту

Сортировка по алфавиту

Курс

1

Отбор по курсу

Вывод в файл

| | Фамилия | Пол | Возраст | Специальность | Курс |
|---|---------|-----|---------|---------------|------|
| ▶ | Егорова | ж | 17 | ИБ | 1 |
| | Иванов | м | 19 | ИТ | 2 |
| | Сидоров | м | 19 | ИБ | 3 |
| | Краснов | м | 19 | КТ | 1 |
| | Чиркова | ж | 20 | ИБ | 3 |
| | Петрова | ж | 21 | КТ | 4 |
| | Перов | м | 21 | ИТ | 4 |
| * | | | | | |

Результат сортировки

```
private void button3_Click(object sender, EventArgs e)
{
    student p;
    for (int i = n-1; i >0; i--)
        for (int j = 0; j <i; j++)
            {
                if (massiv[j].fam.CompareTo(massiv[j+1].fam)==1)
                {
                    p = massiv[j];
                    massiv[j] = massiv[j + 1];
                    massiv[j + 1] = p;
                }
            }
    print();
}
```

Обработчик события кнопки «Сортировка по фамилии»

Массив студентов

Вывод всех

Сортировка по возрасту

Сортировка по алфавиту

Курс

1

Отбор по курсу

Вывод в файл

| Фамилия | Пол | Возраст | Специальность | Курс |
|---------|-----|---------|---------------|------|
| Егорова | ж | 17 | ИБ | 1 |
| Иванов | м | 19 | ИТ | 2 |
| Краснов | м | 19 | КТ | 1 |
| Перов | м | 21 | ИТ | 4 |
| Петрова | ж | 21 | КТ | 4 |
| Сидоров | м | 19 | ИБ | 3 |
| Чиркова | ж | 20 | ИБ | 3 |

Результат сортировки

```
private void button4_Click(object sender, EventArgs e)
{
    byte k = Convert.ToByte(numericUpDown1.Value);

    dataGridView1.Rows.Clear();
    foreach (var t in massiv)
        if(t.curs==k)
            dataGridView1.Rows.Add(t.fam, t.pol.ToString(), t.vozr().ToString(),
                                    t.spec, t.curs.ToString());
}
```

Обработчик события кнопки «Отбор по курсу»

Массив студентов

Вывод всех

Сортировка по возрасту

Сортировка по алфавиту

Курс

3

Отбор по курсу

Вывод в файл

| Фамилия | Пол | Возраст | Специальность | Курс |
|---------|-----|---------|---------------|------|
| Сидоров | м | 19 | ИБ | 3 |
| Чиркова | ж | 20 | ИБ | 3 |
| * | | | | |

Результат отбора

```
private void button5_Click(object sender, EventArgs e)
{
    string fl;
    k++;
    fl = "Результат " + k.ToString();
    writefl = File.CreateText(fl + ".txt");
    string result = string.Format("{0,12}{1,5}{2,8}{3,19}{4,7}", "Фамилия", "Пол",
        "Возраст", "Специальность", "Курс");

    writefl.WriteLine(result);
    for (int k = 0; k < dataGridView1.Rows.Count - 1; k++)
    {
        result = string.Format("{0,12}{1,5}{2,8}{3,19}{4,7}",
            dataGridView1.Rows[k].Cells[0].Value,
            dataGridView1.Rows[k].Cells[1].Value,
            dataGridView1.Rows[k].Cells[2].Value,
            dataGridView1.Rows[k].Cells[3].Value,
            dataGridView1.Rows[k].Cells[4].Value);

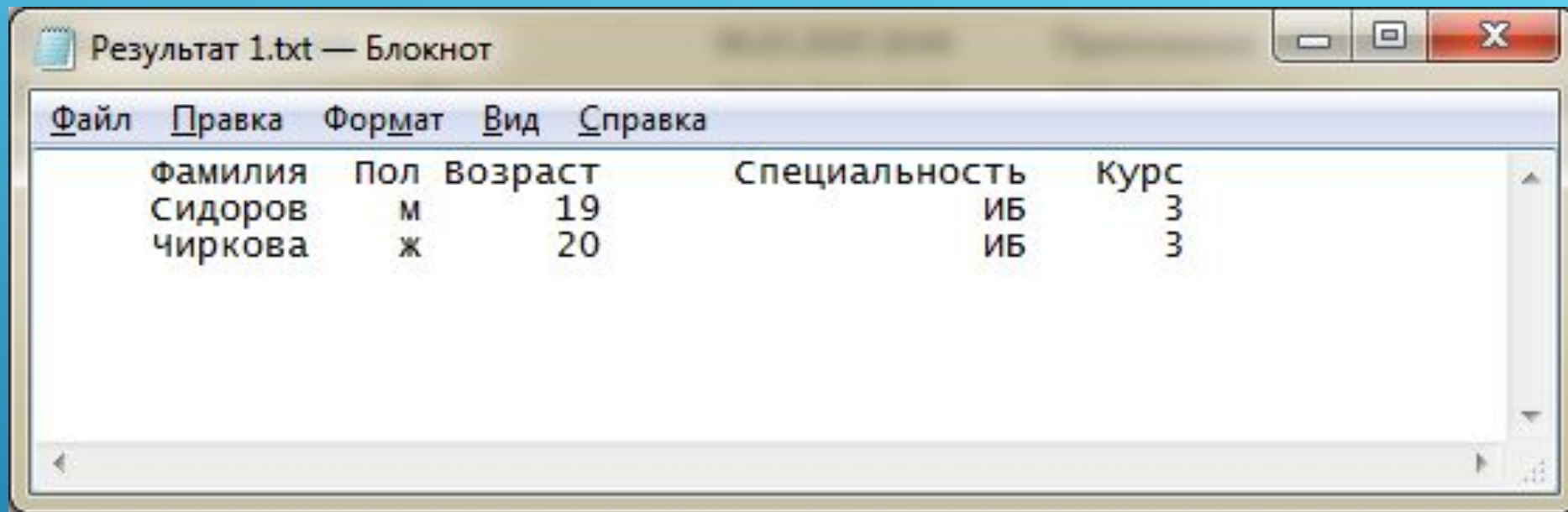
        writefl.WriteLine(result);
    }
    writefl.Close();
}
```

Обработчик события кнопки «Вывод в файл»

The screenshot shows a software application window titled "Массив студентов". On the left side, there are several control buttons: "Вывод всех", "Сортировка по возрасту", "Сортировка по алфавиту", a "Курс" section with a dropdown menu showing "3", "Отбор по курсу", and "Вывод в файл". The main area on the right contains a table with the following data:

| | Фамилия | Пол | Возраст | Специальность | Курс |
|---|---------|-----|---------|---------------|------|
| ▶ | Сидоров | м | 19 | ИБ | 3 |
| | Чиркова | ж | 20 | ИБ | 3 |
| * | | | | | |

Результат отбора



Результат вывода в файл