

ПРОЕКТИРОВАНИЕ ПРИЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА БРОНИРОВАНИЯ И ПРОДАЖИ БИЛЕТОВ В ЖЕЛЕЗНОДОРОЖНЫХ КАССАХ.

Григорьев А.Д.

*Технический колледж ФГОУ ВПО «Тамбовский государственный
технический университет»*

Тамбов, Россия

Предметная область представляет собой процесс продажи билетов в железнодорожных кассах. Ставится задача разработать информационную модель рассматриваемой области. В системе должны поддерживаться режимы учета пассажиров, выбора направления и цены билета, выбор даты на данное направление, поиск по всей введенной информации.

Для решения поставленных задач необходимо разработать информационную модель предметной области.

Выделенные нами объекты представим в виде переменных и классов, связи между ними будут реализовываться с помощью интерфейса оконного приложения.

Объекты, планируемые к использованию:

```
this.button1 = new System.Windows.Forms.Button();- кнопка расчета цены
this.family = new System.Windows.Forms.TextBox();- текстовое поле для ввода фамилии
this.name = new System.Windows.Forms.TextBox();текстовое поле для ввода имени
this.patronymic = new System.Windows.Forms.TextBox();текстовое поле для отчества
this.passport = new System.Windows.Forms.TextBox();текстовое поле для ввода паспортных
данных
this.price = new System.Windows.Forms.TextBox();текстовое поле для вывода расчетного
значения цены
this.sv = new System.Windows.Forms.RadioButton();- переключатель комфортности
this.kupe = new System.Windows.Forms.RadioButton();- переключатель комфортности
this.up = new System.Windows.Forms.ComboBox();-выпадающий список пункта отправления
this.down = new System.Windows.Forms.ComboBox();()-выпадающий список пункта назначения
this.listView1 = new System.Windows.Forms.ListView();- таблица для вывода всех данных
this.monthCalendar1 = new System.Windows.Forms.MonthCalendar();-календарь
this.button2 = new System.Windows.Forms.Button();-кнопка для бронирования билета
this.textBox1 = new System.Windows.Forms.TextBox();-поле запроса поиска
this.button3 = new System.Windows.Forms.Button();- кнопка поиска
```

Окно приложения должно обеспечивать необходимые функции системы, связи между ее компонентами, его модель может выглядеть как на рисунке 1:

№	Фамилия	Паспорт	Направление	Тип	Цена	Дата
1	Григорьев А.Д.	48468248624	Тамбов-Москва	Купе	3000 руб.	26.01.2014
2	Hghjgh P.P.	682468246	Тамбов-Москва	Св	2000 руб.	28.02.2014

Рисунок 1 – Главное окно программы

Исходными данными для приложения являются анкетные данные покупателя билетов, выбор пунктов отправления и назначения, выбор комфортности, дата поездки. В зависимости от этих данных рассчитывается цена билетов.

Основные элементы программного кода:

Определение Цены билеты:

```
int top, bottom, raz;
double cena=2000;
top = up.SelectedIndex;
bottom = down.SelectedIndex;
raz = Math.Abs(top - bottom);
if ((up.SelectedIndex == down.SelectedIndex) || (family.Text == "") || (name.Text == "")
|| (patronymic.Text == "") || (passport.Text == ""))
{
    MessageBox.Show("Проверьте правильность ввода данных");
}
else
{
    if (raz == 0)
    {
        price.Text = "0";
    }
    if (raz == 1)
    {
        if (kupe.Checked)
        {
            price.Text = (cena * 1.5).ToString();
        }
        else
    }
```

```

        {
            price.Text = (cena).ToString();
        }
    }
    if (raz == 2)
    {
        if (kupe.Checked)
        {
            price.Text = (cena * 1.5 * 2).ToString();
        }
        else
        {
            price.Text = (cena * 2).ToString();
        }
    }
    if (raz == 3)
    {
        if (kupe.Checked)
        {
            price.Text = (cena * 1.5 * 3).ToString();
        }
        else
        {
            price.Text = (cena * 3).ToString();
        }
    }
    if (raz == 4)
    {
        if (kupe.Checked)
        {
            price.Text = (cena * 1.5 * 4).ToString();
        }
        else
        {
            price.Text = (cena * 4).ToString();
        }
    }
    button3.Enabled = true;
}

```

В зависимости от направления и от комфортности цена билетов различается, за это отвечает коэффициент.

Следующая процедура отвечает за поиск данных по файлу.

```

int k = 0;
if (!System.IO.File.Exists("ddd.txt"))
{
    System.IO.File.WriteAllText("ddd.txt", "");
}
StreamReader sw = new StreamReader("ddd.txt");
char[] c = { '@' };
string[] s;

```

```

while (!sw.EndOfStream)
{
    try
    {
        s = sw.ReadLine().Split(c, StringSplitOptions.RemoveEmptyEntries);
        int.TryParse(s[0], out n);
    }
    catch
    {
        sw.Close();
        return;
    }
    //s[1] = s[1].Substring(0, s[1].Length - 5);
    string v = s[1].Substring(0, s[1].Length - 5);
    if (v == textBox1.Text)
    {
        string w = s[0] + " " + s[1] + " " + s[2] + " " + s[3] + " " + s[4] + " " + s[5] + " " +
s[6];
        MessageBox.Show(w);
        k = 1;
    }
}
if(k==0)
{
    MessageBox.Show("Бронирование не найдено");
}
sw.Close();
textBox1.Text = "";
}

```

Поиск ведется по файлу ddd.txt в котором содержится необходимая информация о пассажирах. Создается буферная переменная `string[] s`;

В которую помещается информация из файла. Условием завершения просмотра файла является конец потока `while (!sw.EndOfStream)`. далее происходит сравнение условия заданного в элементе `textBox1.Text` с фамилией пассажира. При выполнении условия поиска информация помещается в переменную `w` и выводится в окно сообщений. При отрицательном результате поиска выводится сообщение «бронирование не найдено».

В следующем фрагменте кода происходит запись информации в файл ddd.txt `n++`;

```

string k = n.ToString();
string u = up.Text + "-" + down.Text;
string f;
string Na, Ot;
char[] n1 = name.Text.ToCharArray();
string m = monthCalendar1.SelectionStart.Date.ToShortDateString();
//MessageBox.Show(((char)((int)n1[0] - 32)).ToString());
Na = getBig(name.Text);
Ot = getBig(patronymic.Text);
//MessageBox.Show(getBig(family.Text));

```

```

f = getBig(family.Text) + " " + Na[0] + "." + Ot[0] + ".";
string pri = price.Text + " руб.";
if (sv.Checked)
{
    t = "СВ";
}
else
{
    t = "Купе";
}

listView1.Items.Add(k);
listView1.Items[n - 1].SubItems.Add(f);
listView1.Items[n - 1].SubItems.Add(passport.Text);
listView1.Items[n - 1].SubItems.Add(u);
listView1.Items[n - 1].SubItems.Add(t);
listView1.Items[n - 1].SubItems.Add(pri);
listView1.Items[n - 1].SubItems.Add(m);
if (!System.IO.File.Exists("ddd.txt"))
{
    System.IO.File.WriteAllText("ddd.txt", "");
}

StreamWriter sw = new StreamWriter("ddd.txt", true);
sw.WriteLine(k + "@" + f + "@" + passport.Text + "@" + u + "@" + t + "@" + pri +
"@ " + m);
sw.Close();

```

Вся информация выводится в таблицу на главной форме приложения, параллельно ведется запись в файл ddd.txt.

В следующем программном коде происходит считывание данных из файла и загрузку в таблицу.

```

up.SelectedIndex = 0;
down.SelectedIndex = 0;

StreamReader sw = new StreamReader("ddd.txt");
char[] c = { '@' };
string[] s;
if (!System.IO.File.Exists("ddd.txt"))
{
    System.IO.File.WriteAllText("ddd.txt", "");
}

while (!sw.EndOfStream)
{
    try
    {
        s = sw.ReadLine().Split(c, StringSplitOptions.RemoveEmptyEntries);
        int.TryParse(s[0], out n);
    }
    catch
    {

```

```
        sw.Close();  
        return;  
    }  
  
    listView1.Items.Add(s[0]);  
    listView1.Items[n - 1].SubItems.Add(s[1]);  
    listView1.Items[n - 1].SubItems.Add(s[2]);  
    listView1.Items[n - 1].SubItems.Add(s[3]);  
    listView1.Items[n - 1].SubItems.Add(s[4]);  
    listView1.Items[n - 1].SubItems.Add(s[5]);  
    listView1.Items[n - 1].SubItems.Add(s[6]);  
}  
sw.Close();
```

Открывается поток для чтения данных, проверяется на существование файла, далее идет цикл обеспечивающий запись в таблицу.