**AddEventForm.cs**

using System;

using System.Windows.Forms;

namespace Organizer1

{

public partial class AddEventForm : Form

{

public Organizer organizer;

public AddEventForm()

{

InitializeComponent();

}

public void setOrganize(ref Organizer organizer)

{

this.organizer = organizer;

}

private void AddEventForm\_Load(object sender, EventArgs e)

{

comboBox1.Items.Clear();

comboBox1.Items.Add(EventType.METEENG.ToString());

comboBox1.Items.Add(EventType.TASK.ToString());

comboBox1.Select(0, 1);

}

private void button1\_Click(object sender, EventArgs e)

{

if (DateTime.Now.Ticks > monthCalendar1.SelectionStart.Ticks)

{

MessageBox.Show("Данная дата уже прошла!");

return;

}

organizer.add(new Data(

textBox1.Text,

monthCalendar1.SelectionStart,

dateTimePicker1.Value.TimeOfDay,

(EventType)Enum.Parse(typeof(EventType), comboBox1.Text, true)

));

this.Close();

}

}

}

**AuthForm.cs**

using System;

using System.Windows.Forms;

namespace Organizer1

{

public partial class AuthForm : Form

{

const String login = "admin";

const String password = "admin";

public AuthForm()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

String tmpLogin = textBox1.Text.Trim();

String tmpPassword = textBox2.Text.Trim();

if (!tmpLogin.Equals(login) || !tmpPassword.Equals(password))

{

MessageBox.Show("Данные аккаунта не верны");

return;

}

else

{

MessageBox.Show("Авторизация прошла успешно");

OrganizerForm organizerForm = new OrganizerForm();

organizerForm.Show();

this.Hide();

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

Boolean isShowPassword = checkBox1.Checked;

textBox2.PasswordChar = !isShowPassword ? '\*' : '\0';

}

private void button2\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("До встречи", "Прощальный", MessageBoxButtons.OK) == DialogResult.OK)

{

Close();

}

}

}

}

**Data.cs**

using System;

namespace Organizer1

{

public enum EventType

{

TASK = 0,

METEENG = 1,

}

public struct Data

{

public string Event { get; set; }

public DateTime Date { get; set; }

public TimeSpan Time { get; set; }

public EventType eventType { get; set; }

public Data(String Event, DateTime Date, TimeSpan Time, EventType eventType)

{

this.Event = Event;

this.Date = Date;

this.Time = Time;

this.eventType = eventType;

}

}

}

**Organizer.cs**

using System;

using System.Collections.Generic;

namespace Organizer1

{

public class Organizer

{

public List<Data> list = new List<Data>();

public Organizer(Data data)

{

list.Add(data);

}

public void add(Data data)

{

this.list.Add(data);

}

public Organizer()

{

}

public void remove(int index)

{

if (index < list.Count)

{

list.RemoveAt(index);

}

}

public List<Data> findByText(string text)

{

List<Data> tmpList = new List<Data>();

list.ForEach(item =>

{

if (item.Event.IndexOf(text) >= 0)

{

tmpList.Add(item);

}

});

return tmpList;

}

public List<Data> findByCategory(EventType type)

{

List<Data> tmpList = new List<Data>();

list.ForEach(item =>

{

if (item.eventType == type)

{

tmpList.Add(item);

}

});

return tmpList;

}

public void sortByTime(int direction = 0)

{

list.Sort((x, y) =>

{

int timeX = x.Time.Hours \* 3600 + x.Time.Minutes;

int timeY = y.Time.Hours \* 3600 + y.Time.Minutes;

if (direction == 0)

{

if (timeX > timeY)

return 1;

if (timeX < timeY)

return -1;

if (timeX == timeY)

return 0;

}

return 0;

});

}

}

}

**OrganizerForm.cs**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Windows.Forms;

namespace Organizer1

{

public partial class OrganizerForm : Form

{

public Organizer organizer = new Organizer();

public OrganizerForm()

{

InitializeComponent();

}

private void OrganizerForm\_FormClosing(object sender, FormClosingEventArgs e)

{

Close();

}

private void button3\_Click(object sender, EventArgs e)

{

AddEventForm eventForm = new AddEventForm();

eventForm.Show();

eventForm.setOrganize(ref organizer);

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

refreshDataGridView();

}

public void refreshDataGridView()

{

dataGridView1.Rows.Clear();

organizer.list.ForEach(item =>

{

DataGridViewRow row = new DataGridViewRow();

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Date.Year}-{item.Date.Month}-{item.Date.Day}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Time.Hours}:{item.Time.Minutes}:{item.Time.Seconds}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = item.Event.ToString() });

row.Cells.Add(new DataGridViewTextBoxCell { Value = item.eventType.ToString(), Style = new DataGridViewCellStyle { BackColor = item.eventType == EventType.TASK ? Color.Red : Color.Green } });

dataGridView1.Rows.Add(row);

});

}

private void OrganizerForm\_Activated(object sender, EventArgs e)

{

refreshDataGridView();

}

private void button2\_Click(object sender, EventArgs e)

{

groupBox3.Visible = !groupBox3.Visible;

button2.Text = groupBox3.Visible ? "Скрыть поиск" : "Найти";

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

List<Data> tmp = organizer.findByText(textBox1.Text);

dataGridView1.Rows.Clear();

tmp.ForEach(item =>

{

DataGridViewRow row = new DataGridViewRow();

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Date.Year}-{item.Date.Month}-{item.Date.Day}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Time.Hours}:{item.Time.Minutes}:{item.Time.Seconds}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = item.Event.ToString() });

dataGridView1.Rows.Add(row);

});

}

private void button4\_Click(object sender, EventArgs e)

{

organizer.sortByTime(0);

refreshDataGridView();

}

private void OrganizerForm\_Load(object sender, EventArgs e)

{

comboBox1.Items.Clear();

comboBox1.Items.Add(EventType.METEENG.ToString());

comboBox1.Items.Add(EventType.TASK.ToString());

comboBox1.Select(0, 1);

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

List<Data> tmp = organizer.findByCategory((EventType)Enum.Parse(typeof(EventType), comboBox1.Text, true));

dataGridView1.Rows.Clear();

tmp.ForEach(item =>

{

DataGridViewRow row = new DataGridViewRow();

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Date.Year}-{item.Date.Month}-{item.Date.Day}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = $"{item.Time.Hours}:{item.Time.Minutes}:{item.Time.Seconds}" });

row.Cells.Add(new DataGridViewTextBoxCell { Value = item.Event.ToString() });

dataGridView1.Rows.Add(row);

});

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

refreshDataGridView();

}

}

}