

Organizator

Proiectul realizat de mine este un organizator, adica o agenda, in care se pot face mai multe programari si in care se pot retine intru-un fisier.

```
private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e) {
    // MessageBox::Show( "Selected Item Text: " +
selectedItem->ToString() + "\n" +
    // "Index: " + selectedIndex.ToString());
    combOra->SelectedIndex=0;
    combOra->Items->Count;
    for (int i=0;i<combOra->Items->Count;i++)
    {
        combOra->SelectedIndex=i;
        Object^ item=combOra->SelectedItem;
        String^ aux=item->ToString();
        array<String^>^row1=gcnew array
<String^>{(aux)};

        dataGridView1->Rows->Add(row1);
    }
}
```

```

        dataGridView1->SelectionMode =
DataGridViewSelectionMode::FullRowSelect;
        ClearGrid();
        combOra->Text="8:00";
        fisier fis("programari.txt",1);
        v_programari = fis.citire_fisier();
        if (v_programari.size()==0)
MessageBox::Show("File programari.txt not found!");
        string today_date;
        MarshalString(monthCalendar1-
>TodayDate.ToShortDateString(),today_date);
        for (unsigned int i=0; i<v_programari.size();i+
+)
        {
            if (v_programari[i].data == today_date)
            {
                int index =
GetRowIndex(v_programari[i].ora);
                dataGridView1->Rows[index]-
>Cells[1]->Value = gcnew String(v_programari[i].subiect.c_str());
                dataGridView1->Rows[index]-
>Cells[2]->Value = gcnew String(v_programari[i].detalii.c_str());
                dataGridView1->Rows[index]-
>Cells[0]->Style->BackColor=Color::Red;
                dataGridView1->Rows[index]-
>Cells[1]->Style->BackColor=Color::Red;
                dataGridView1->Rows[index]-
>Cells[2]->Style->BackColor=Color::Red;
            }
        }
    }
    private: System::Void monthCalendar1_DateSelected(System::Object^
sender, System::Windows::Forms::DateRangeEventArgs^ e) {
        ClearGrid();
        String^ data_sel = String::Format( "{0}", e-
>Start.ToShortDateString() );
        // dataGridView1->Rows[1]->Cells[1]-
>FormattedValue->ToString();
        string sel_date;
        int index;
        MarshalString(data_sel,sel_date);
        for (unsigned int i=0; i<v_programari.size();i+
+)
        {
            if (v_programari[i].data == sel_date)
            {
                index =
GetRowIndex(v_programari[i].ora);
                dataGridView1->Rows[index]-
>Cells[1]->Value = gcnew String(v_programari[i].subiect.c_str());
                dataGridView1->Rows[index]-
>Cells[2]->Value = gcnew String(v_programari[i].detalii.c_str());
                dataGridView1->Rows[index]-
>Cells[0]->Style->BackColor=Color::Red;
                dataGridView1->Rows[index]-
>Cells[1]->Style->BackColor=Color::Red;

```

```

                                dataGridView1->Rows[index]-
>Cells[2]->Style->BackColor=Color::Red;
                                }
                                }
                                }
private: System::Void butAdauga_Click(System::Object^ sender,
System::EventArgs^ e) {
    bool test = false;
    string ora;
    MarshalString(combOra->Text, ora);
    int index = GetRowIndex(ora);
    if (dataGridView1->Rows[index]->Cells[1]-
>FormattedValue->ToString()=="")
    {
        dataGridView1->Rows[index]->Cells[1]->Value =
textBSubiect->Text;
        dataGridView1->Rows[index]->Cells[2]->Value =
textBDetalii->Text;
        dataGridView1->Rows[index]->Cells[0]->Style-
>BackColor=Color::Red;
        dataGridView1->Rows[index]->Cells[1]->Style-
>BackColor=Color::Red;
        dataGridView1->Rows[index]->Cells[2]->Style-
>BackColor=Color::Red;
        string date_sel, detalii, subiect;
        MarshalString(textBSubiect->Text, subiect);
        MarshalString(textBDetalii->Text, detalii);
        MarshalString(monthCalendar1->SelectionRange-
>Start.ToShortDateString(), date_sel);
        program temp;
        temp.data = date_sel;
        temp.ora = ora;
        temp.detalii= detalii;
        temp.subiect = subiect;
        v_programari.push_back(temp);
        test = true;
        textBSubiect->Text = "";
        textBDetalii->Text = "";
    }
    if(test == false)
    {
        MessageBox::Show("Exista deja o programare pt
data si ora selectata!");
    }

}

private: void MarshalString ( System::String^ s, std::string& os )
{
    using namespace
System::Runtime::InteropServices;
    const char* chars = (const char*)
(Marshal::StringToHGlobalAnsi(s)).ToPointer();
    os = chars;

Marshal::FreeHGlobal (IntPtr((void*) chars));

```

```
}
```

```
private: System::Void ClearGrid()
{
    for (unsigned int i=0;i<13;i++)
    {
        dataGridView1->Rows[i]->Cells[1]->Value ="";
        dataGridView1->Rows[i]->Cells[2]->Value ="";
        dataGridView1->Rows[i]->Cells[0]->Style-
>BackColor = Color::LightGreen;
        dataGridView1->Rows[i]->Cells[1]->Style-
>BackColor = Color::LightGreen;
        dataGridView1->Rows[i]->Cells[2]->Style-
>BackColor =Color::LightGreen;
    }
}
```

```
private: System::Void Form1_FormClosing(System::Object^ sender,
System::Windows::Forms::FormClosingEventArgs^ e) {
    fisier fis("programari.txt",0);
    for (unsigned int i=0;i<v_programari.size();i++)
    {
        fis.scriere_fisier(v_programari[i]);
    }
}
```

```
private: System::Void butSterge_Click(System::Object^ sender,
System::EventArgs^ e) {
    int index = dataGridView1->CurrentRow->Index;
    string ora,data,subiectul;

    MarshalString(dataGridView1->Rows[index]->Cells[0]-
>FormattedValue->ToString(),ora) ;
    MarshalString(monthCalendar1->SelectionRange-
>Start.ToShortDateString(),data);
    MarshalString(dataGridView1->Rows[index]->Cells[1]-
>FormattedValue->ToString(),subiectul);
    if (subiectul!="")
    {
        unsigned int i=0;
        for ( i=0;i<v_programari.size();i++)
        {
            string temp_data =
v_programari[i].data;
            string temp_ora =
v_programari[i].ora;
            string temp_sub =
v_programari[i].subiect;
            if
((v_programari[i].data==data)&&(v_programari[i].ora==ora)&&(v_programari
[i].subiect==subiectul))
            {
                break;
            }
        }
        if (i<v_programari.size())
```

```

        {

            v_programari.erase(v_programari.begin()+i);
                                dataGridView1->Rows[index]-
>Cells[1]->Value="";
                                dataGridView1->Rows[index]-
>Cells[2]->Value="";
                                dataGridView1->Rows[index]-
>Cells[0]->Style->BackColor = Color::LightGreen;
                                dataGridView1->Rows[index]-
>Cells[1]->Style->BackColor = Color::LightGreen;
                                dataGridView1->Rows[index]-
>Cells[2]->Style->BackColor =Color::LightGreen;
                                }
                                else MessageBox::Show("Info not found in
file!!");

        }

    }

```