LUCRARE 7_____13/03/2024

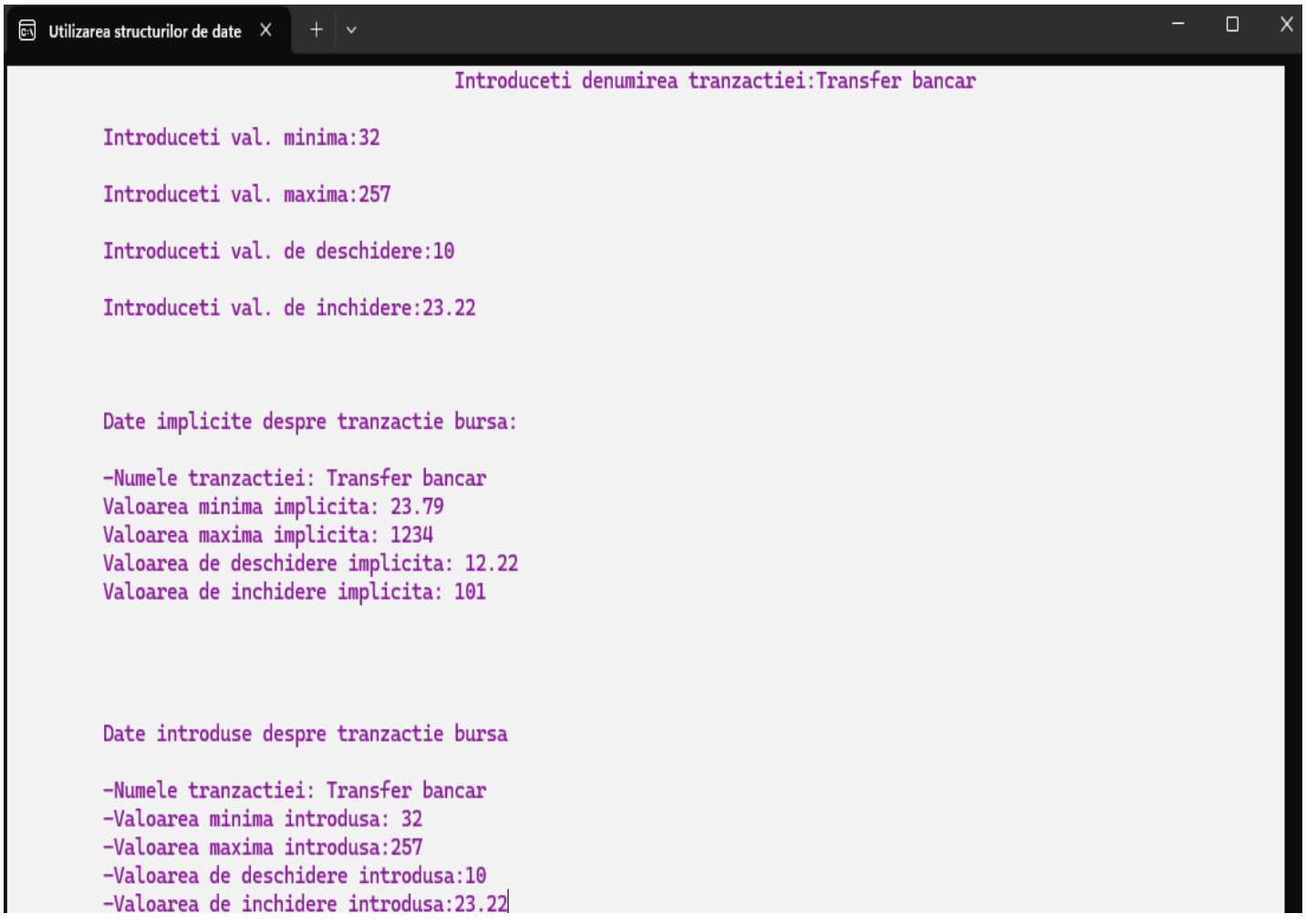
Codul C++:

```cpp
#include <iostream>
#include <string>
using namespace std;
struct tranz_bursa
{
    string nume_actiune;
    double val_min, val_max, val_desch, val_inch;
    tranz_bursa()
    {
        nume_actiune = "Transfer bancar";
        val_min = 23.79;
        val_max = 1234;
        val_desch = 12.22;
        val_inch = 101;
    }
    tranz_bursa(string n, double v_min, double v_max, double v_d, double v_i)
    {
        nume_actiune = n;
        val_min = v_min;
        val_max = v_max;
        val_desch = v_d;
        val_inch = v_i;
    }
    void show_data(bool def = false)
    {
        if (def)
        {
            cout << "\n\n\n\t Date implicite despre tranzactie bursa: ";
            cout << "\n\n\t -Numele tranzactiei: Transfer bancar";
            cout << "\n\t Valoarea minima implicita: 23.79";
            cout << "\n\t Valoarea maxima implicita: 1234";
            cout << "\n\t Valoarea de deschidere implicita: 12.22";
            cout << "\n\t Valoarea de inchidere implicita: 101";
        }
        else
        {
            cout << "\n\n\n\t Date introduse despre tranzactie bursa";
            cout << "\n\n\t -Numele tranzactiei: " << nume_actiune;
            cout << "\n\t -Valoarea minima introdusa: " << val_min;
            cout << "\n\t -Valoarea maxima introdusa:" << val_max;
            cout << "\n\t -Valoarea de deschidere introdusa:" << val_desch;
            cout << "\n\t -Valoarea de inchidere introdusa:" << val_inch;
        }
    }
};
```

```cpp
int main(void)
{
    string den;
    double vmi, vma, vd, vi;
    system("TITLE Utilizarea structurilor de date");
    system("COLOR F5");
    cout << "\n\t\t\t\t\t\t\t In program";
    cout << "\n\t\t\t\t\t -Se defineste structura tranz_bursa";
    cout << "\n\t\t\t\t\t -Se defineste constructorul tranz_bursa";
    cout << "\n\t\t\t\t\t  Introduceti denumirea tranzactiei:";
    getline(cin, den);
    cout << "\n\t Introduceti val. minima:"; cin >> vmi;
    cout << "\n\t Introduceti val. maxima:"; cin >> vma;
    cout << "\n\t Introduceti val. de deschidere:"; cin >> vd;
    cout << "\n\t Introduceti val. de inchidere:";  cin >> vi;
    tranz_bursa t(den, vmi, vma, vd, vi);
    t.show_data(true);
    cout << endl << endl;
    t.show_data(0);
    cin.ignore();
    cin.get();
    return 0;

}
```

Caseta de afisare:



Codul C#:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace lucr7csharp
{
    3 references
    public partial class Form1 : Form
    {
        2 references
        struct tranz_bursa
        {
            1 reference
            public DateTime Date { get; set; }
            7 references
            public double Price { get; set; }
            1 reference
            public int val_min { get; set; }
            1 reference
            public int val_max { get; set; }
            1 reference
            public int val_desch { get; set; }
            1 reference
            public int val_inch { get; set; }
            0 references
            public int line_start { get; set; }
            0 references
            public int line_end { get; set; }
        }
        private Random random = new Random();
        private tranz_bursa[] data;
        1 reference
        public Form1()
        {
            InitializeComponent();
            InitializeData();
            this.Paint += DrawChart;
        }
```

```
1 reference
private void InitializeData()
{
    int numDataPoints = 20;
    data=new tranz_bursa[numDataPoints];
    DateTime startDate = DateTime.Today.AddDays(-numDataPoints);
    for (int i = 0; i < numDataPoints; i++)
    {
        data[i].Date = startDate.AddDays(i);
        data[i].Price = random.Next(5,31);
    }

}

1 reference
private void DrawChart(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    int candle_w = 10, candle_h = 50, max_h=ClientSize.Height-50;
    double max_price = GetMaxPrice();
    int candle_spacing = (ClientSize.Width - 50) / data.Length;

    Point[] candleCenters = new Point[data.Length];
    for(int i = 0;i<data.Length;i++)
    {
        int candleTop = (int)((data[i].Price/max_price)*max_h);
        int candleBot = max_h - candleTop;
        int candleCenterX = i * candle_spacing + candle_spacing / 2;
        candleCenters[i] = new Point(candleCenterX, max_h - candleTop);
    }

    g.DrawLines(Pens.DarkSlateBlue, candleCenters);

    for(int i = 0;i<data.Length;i++)
    {
        int candleTop = (int)((data[i].Price / max_price) * max_h);
        int candleBot=max_h - candleTop;
        int candle_left = i * candle_spacing + candle_spacing / 2 - candle_w / 2;

        g.FillRectangle(data[i].Price > data[Math.Max(0,i-1)].Price ? Brushes.MediumPurple: Brushes.DeepPink,candle_left,max_h-candleTop,candle_w,candle_h);
```

```
            int lineX = candle_left + candle_w / 2;
            int lineTop = max_h - candleTop - 15;
            int lineBottom = lineTop + 80;
            g.DrawLine(Pens.Blue, lineX, lineTop,lineX, lineBottom);
            data[i].val_min = lineTop;
            data[i].val_max = lineBottom;
            data[i].val_desch = max_h - candleTop;
            data[i].val_inch = max_h - candleTop;
            int secondLineY = max_h - candleTop + 25;

            g.DrawLine(Pens.Black,lineX,lineTop,lineX,secondLineY);
        }
    }
    1 reference
    private double GetMaxPrice()
    {
        double max_price = 50;
        foreach(var item in data)
        {
            if(item.Price > max_price)
                max_price = item.Price;
        }
        return max_price;
    }
}
}
```

Caseta de executie: