МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

Язык программирования C#. Разработка простых графических и сетевых программ

Дисциплина: «Объектно-ориентированное программирование»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Сорокин А.Н. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-41 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

**Цель работы:** научиться создавать простые графические и сетевые приложения на языке программирования C#.

# Практические задания

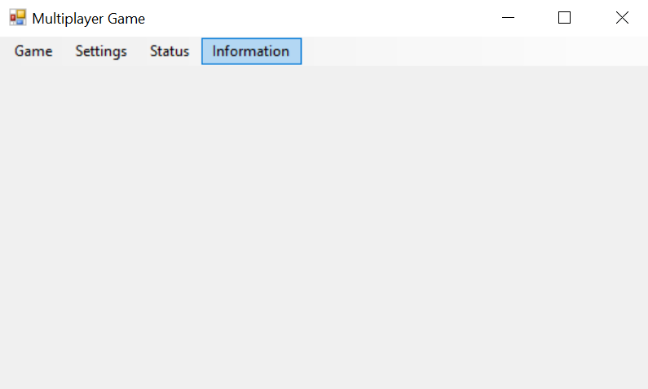
1. Добавить пункт меню для вызова диалогового окна с информацией о программе.

2. Добавить в программу возможность хранения имени игрока. При создании нового игрока пользователь должен указать имя игрока.

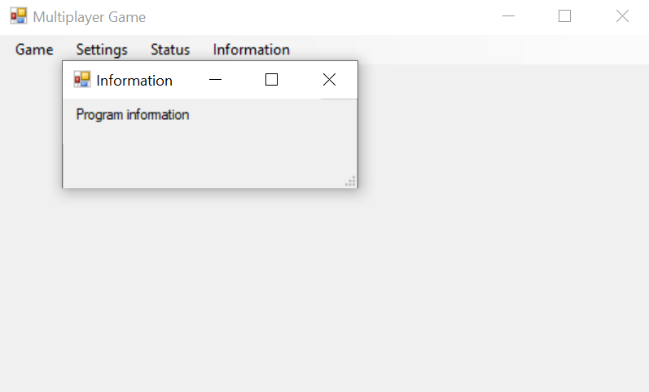
3. Добавить пункт меню для вызова диалогового окна с настройками игрока. В диалоговом окне пользователь может поменять имя игрока и его координаты.

4. Добавить пункт меню для вывода статистики игры. В самом простом варианте можно вывести список всех игроков.

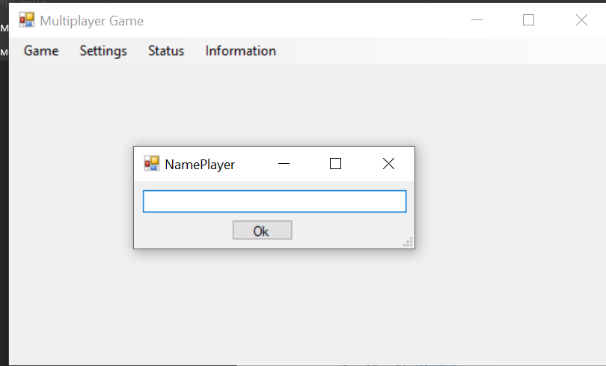
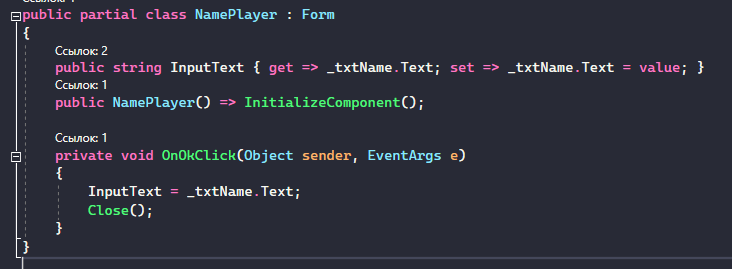
1. Вызов формы информация происходит в меню.



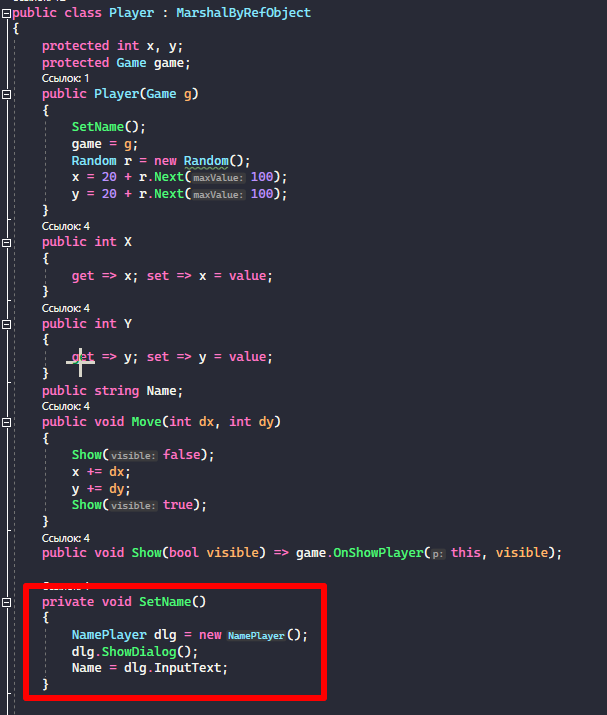
Окно информации.



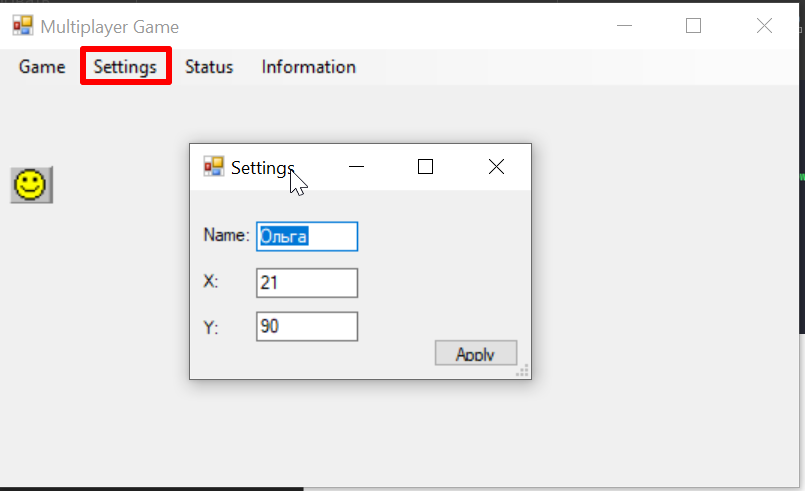
2. Добавление имени игрока.

Вызов формы создания игрока.

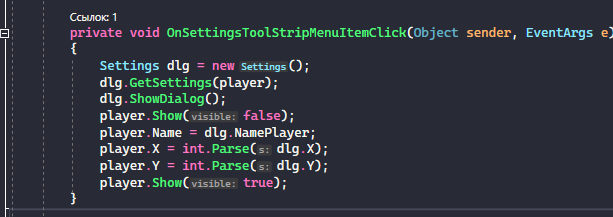


3. Настройки игрока.

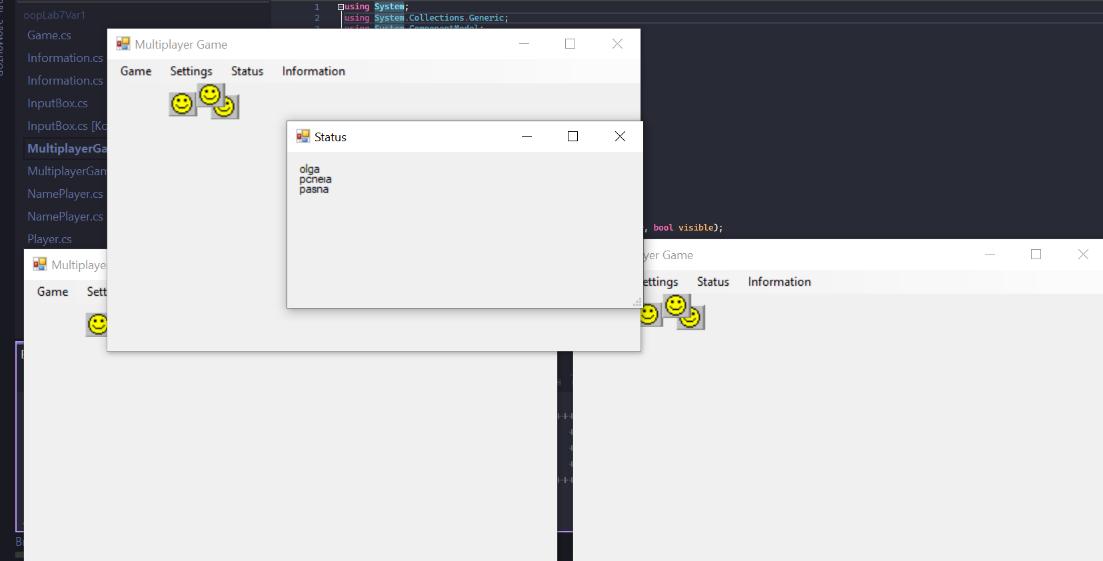




Вызов формы настроек игрока в (MultiplayerGame.cs)

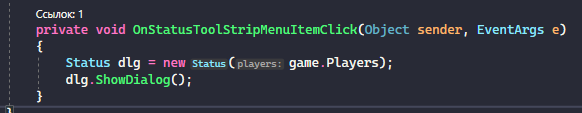


4.Вывод статистики





Вызов формы статистики в (MultiplayerGame.cs)



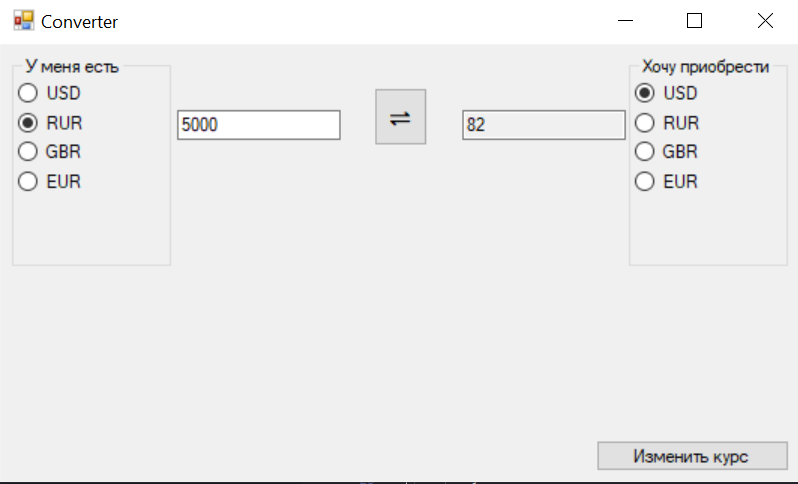
# Дополнительное задание

# Вариант 1

Создайте диалоговое приложение «Конвертер валюты», позволяющее конвертировать денежную сумму из одной валюты в другую. Курсы валют хранятся в файле или базе данных и могут редактироваться в приложении.

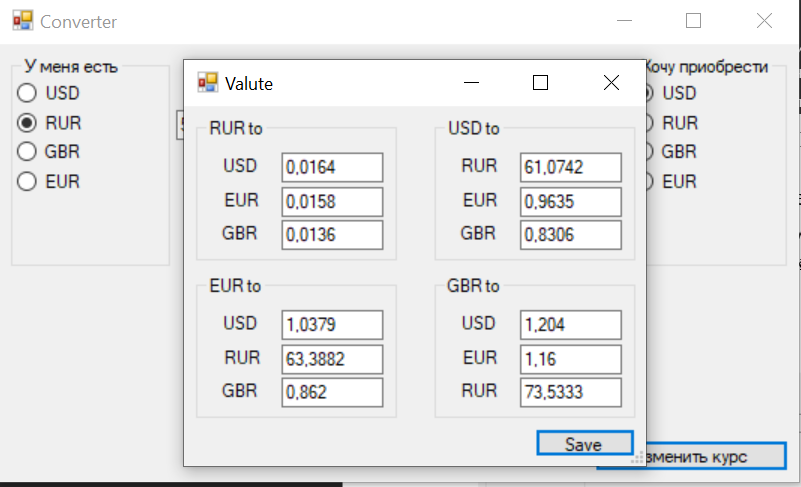
Главная форма конвертера.

При первом запуске приложения курс валют создается по умолчанию. В дальнейшем сохраняется в файл "valutes.data".



При нажатии кнопки со стрелками, происходит разворот валют. (В данном случае «хочу приобрести» станет RUR, «у меня есть» – USD, значение в текстовом блоке поменяется в зависимости от курса новой валюты).

При нажатии кнопки изменить курс откроется диалоговое окно, в котором возможно изменить курс валют относительно друг друга.



При нажатии “Save”, курс меняется и конвертер меняет значение исходя из нового курса, новые значения сохраняются в файл. В дальнейшем при запуске приложения конвертер будет использовать эти значения.

Код программы:

public partial class Form1 : Form

{

private Valuta \_valuta = new Valuta();

public Form1()

{

LoadValute();

InitializeComponent();

\_txtHave.Text = "5000";

}

private void LoadValute()

{

if(!File.Exists("valutes.data"))

\_valuta = new Valuta();

else

{

BinaryFormatter formatter = new BinaryFormatter();

using(FileStream fs = new FileStream("valutes.data", FileMode.OpenOrCreate))

{

\_valuta = (Valuta)formatter.Deserialize(fs);

}

}

}

private void OnHaveTextChanged(Object sender, EventArgs e)

{

try

{

double.Parse(\_txtHave.Text);

}

catch

{

\_txtHave.Text = "";

}

double res = 0;

if(\_txtHave.Text == "")

{

\_txtWant.Text = "";

}

else if(\_rbWantEUR.Checked)

{

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToEUR;

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToEUR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToEUR;

}

else if(\_rbWantRUR.Checked)

{

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToRUR;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToRUR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToRUR;

}

else if(\_rbWantGBR.Checked)

{

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToGBR;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToGBR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToGBR;

}

else if(\_rbWantUSD.Checked)

{

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToUSD;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToUSD;

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToUSD;

}

\_txtWant.Text = res.ToString();

}

private void OnChangeWellClick(Object sender, EventArgs e)

{

var temp = \_txtHave.Text;

Form2 dlg = new Form2(\_valuta);

dlg.ShowDialog();

LoadValute();

\_txtHave.Text = "";

\_txtHave.Text = temp;

}

private void OnTurnClick(Object sender, EventArgs e)

{

string rbHave = "";

string rbWant = "";

foreach(RadioButton item in \_gbHave.Controls)

{

if(item.Checked)

rbHave = item.Name;

}

foreach(RadioButton item in \_gbWant.Controls)

{

if(item.Checked)

rbWant = item.Name;

}

switch(rbHave)

{

case "\_rbHaveUSD": \_rbWantUSD.Checked = true; break;

case "\_rbHaveRUR": \_rbWantRUR.Checked = true; break;

case "\_rbHaveEUR": \_rbWantEUR.Checked = true; break;

case "\_rbHaveGBR": \_rbWantGBR.Checked = true; break;

}

switch(rbWant)

{

case "\_rbWantUSD": \_rbHaveUSD.Checked = true; break;

case "\_rbWantRUR": \_rbHaveRUR.Checked = true; break;

case "\_rbWantEUR": \_rbHaveEUR.Checked = true; break;

case "\_rbWantGBR": \_rbHaveGBR.Checked = true; break;

}

}

private void \_rbHaveUSD\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbHaveUSD.Checked)

{

double res = 0;

if(\_rbWantUSD.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbWantGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToGBR;

if(\_rbWantEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToEUR;

if(\_rbWantRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToRUR;

\_txtWant.Text = res.ToString();

}

}

private void \_rbHaveGBR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbHaveGBR.Checked)

{

double res = 0;

if(\_rbWantGBR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbWantUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToUSD;

if(\_rbWantEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToEUR;

if(\_rbWantRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToRUR;

\_txtWant.Text = res.ToString();

}

}

private void \_rbHaveRUR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbHaveRUR.Checked)

{

double res = 0;

if(\_rbWantRUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbWantGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToGBR;

if(\_rbWantEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToEUR;

if(\_rbWantUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToUSD;

\_txtWant.Text = res.ToString();

}

}

private void \_rbHaveEUR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbHaveEUR.Checked)

{

double res = 0;

if(\_rbWantEUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbWantGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToGBR;

if(\_rbWantUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToUSD;

if(\_rbWantRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToRUR;

\_txtWant.Text = res.ToString();

}

}

private void \_rbWantUSD\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbWantUSD.Checked)

{

double res = 0;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToUSD;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToUSD;

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToUSD;

\_txtWant.Text = res.ToString();

}

}

private void \_rbWantRUR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbWantRUR.Checked)

{

double res = 0;

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToRUR;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToRUR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToRUR;

\_txtWant.Text = res.ToString();

}

}

private void \_rbWantGBR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbWantGBR.Checked)

{

double res = 0;

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToGBR;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.EUR.ToGBR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToGBR;

\_txtWant.Text = res.ToString();

}

}

private void \_rbWantEUR\_CheckedChanged(Object sender, EventArgs e)

{

if(\_rbWantEUR.Checked)

{

double res = 0;

if(\_rbHaveEUR.Checked)

res = double.Parse(\_txtHave.Text);

if(\_rbHaveGBR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.GBR.ToEUR;

if(\_rbHaveRUR.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.RUR.ToEUR;

if(\_rbHaveUSD.Checked)

res = double.Parse(\_txtHave.Text) \* \_valuta.USD.ToEUR;

\_txtWant.Text = res.ToString();

}

}

}

public partial class Form2 : Form

{

public Valuta Valuta;

public Form2(Valuta valuta)

{

Valuta = valuta;

InitializeComponent();

LoadValute();

}

private void LoadValute()

{

\_txtRURtoUSD.Text = Valuta.RUR.ToUSD.ToString();

\_txtRURtoEUR.Text = Valuta.RUR.ToEUR.ToString();

\_txtRURtoGBR.Text = Valuta.RUR.ToGBR.ToString();

\_txtUSDtoRUR.Text = Valuta.USD.ToRUR.ToString();

\_txtUSDtoEUR.Text = Valuta.USD.ToEUR.ToString();

\_txtUSDtoGBR.Text = Valuta.USD.ToGBR.ToString();

\_txtGBRtoRUR.Text = Valuta.GBR.ToRUR.ToString();

\_txtGBRtoUSD.Text = Valuta.GBR.ToUSD.ToString();

\_txtGBRtoEUR.Text = Valuta.GBR.ToEUR.ToString();

\_txtEURtoRUR.Text = Valuta.EUR.ToRUR.ToString();

\_txtEURtoUSD.Text = Valuta.EUR.ToUSD.ToString();

\_txtEURtoGBR.Text = Valuta.EUR.ToGBR.ToString();

}

private void \_btnSave\_Click(Object sender, EventArgs e)

{

Valuta.RUR.ToUSD = double.Parse(\_txtRURtoUSD.Text);

Valuta.RUR.ToEUR = double.Parse(\_txtRURtoEUR.Text);

Valuta.RUR.ToGBR = double.Parse(\_txtRURtoGBR.Text);

Valuta.USD.ToRUR = double.Parse(\_txtUSDtoRUR.Text);

Valuta.USD.ToEUR = double.Parse(\_txtUSDtoEUR.Text);

Valuta.USD.ToGBR = double.Parse(\_txtUSDtoGBR.Text);

Valuta.GBR.ToRUR = double.Parse(\_txtGBRtoRUR.Text);

Valuta.GBR.ToUSD = double.Parse(\_txtGBRtoUSD.Text);

Valuta.GBR.ToEUR = double.Parse(\_txtGBRtoEUR.Text);

Valuta.EUR.ToRUR = double.Parse(\_txtEURtoRUR.Text);

Valuta.EUR.ToUSD = double.Parse(\_txtEURtoUSD.Text);

Valuta.EUR.ToGBR = double.Parse(\_txtEURtoGBR.Text);

SaveValuta();

Close();

}

private void SaveValuta()

{

BinaryFormatter formatter = new BinaryFormatter();

using(FileStream fs = new FileStream("valutes.data", FileMode.OpenOrCreate))

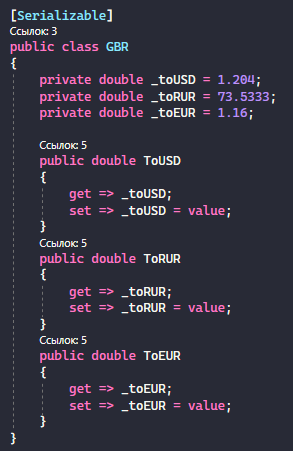
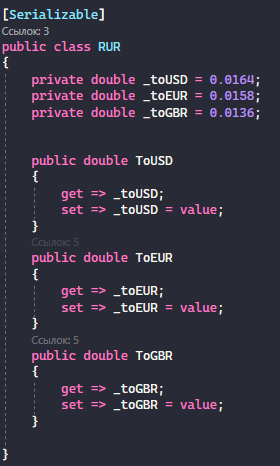
{

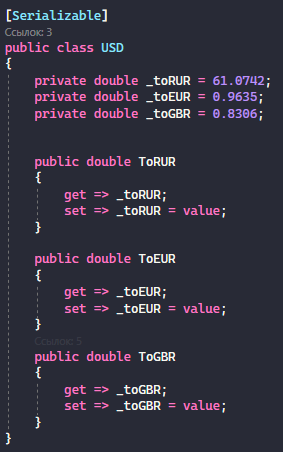
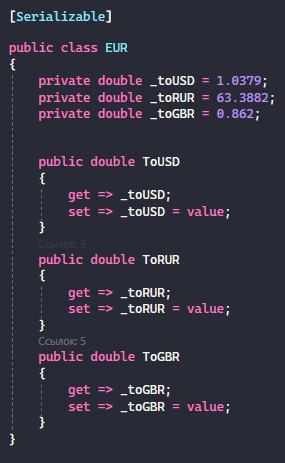
formatter.Serialize(fs, Valuta);

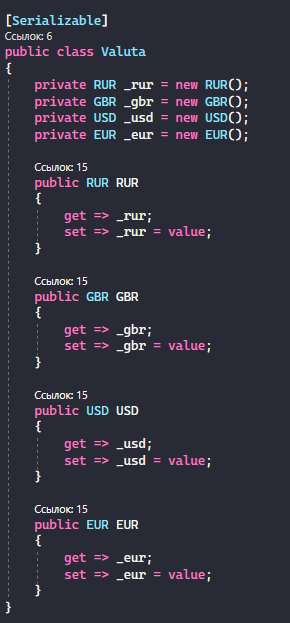
}

}

}



**Вывод:** в ходе лабораторной работы научились создавать простые графические и сетевые приложения на языке программирования C#.

# Контрольные вопросы

1. Что понимается под терминами «делегат» и «событие»? Для чего применяются делегаты и события?

Делегаты представляют такие объекты, которые указывают на методы. То есть делегаты - это указатели на методы и с помощью делегатов мы можем вызвать данные методы.

События сигнализируют системе о том, что произошло определенное действие. И если нам надо отследить эти действия, то как раз мы можем применять события.

2. Что понимается под термином «пространство имен» в языке C#? Приведите примеры описания и использования пространств имен.

Пространства имен позволяют организовать код программы в логические блоки, позволяют объединить и отделить от остального кода некоторую функциональность, которая связана некоторой общей идеей или которая выполняет определенную задачу.

namespace oopLab7Var1Converter.