Оценяване

1. **Задача 1:**
   1. Изграждане на формата – среден 3;
   2. Създаване на програмния код към формата – добър 4;
2. **Задача 2**
   1. В зависимост от валидирането на данните до 5.
3. Отличен 6 – студентът получава допълнително условие, свързано с конкретната задача, но не присъстващо в условието на задачата, например – начисляване на данък ДДС.

**Вариант 1**

Да се създаде Windows приложение, чиято форма да представя кандидат да определена длъжност в офис, информация за който е следната:

* Входящ номер ID - максимум 4 символа;
* Име, презиме и фамилия - максимум 30 символа;
* Брой точки, получени от теста за всяка една от следните офис програми: MS\_Word и MS\_PowerPoint – максимално до 10 точки, а за MS\_Excel и MS\_Access - 15 точки.
* Допълнителни опции - владеене на западен език: английски, френски, немски, всеки от които носи 2 точки.

**Задача 1:** Да се създаден форма от две области:

* **Област Входни данни**: включваща подходящи контроли за представяне на кандидата. Владеенето на език да се представи чрез checkbox. Бутон **New** – за нулиране на всички текстови контроли във формата.
* **Област Резултати**, включваща:
  + Бутон Резултати – натискането на който води до извеждане на входящият номер и името на кандидата в една текстова кутия, а общият брой точки от теста в друга текстова кутия.

**Задача 2:** Да се създаде метод за валидиране на входните данни: входящ номер и точките от теста. Бутон **Резултати** се активира, когато са въведени входните данни.

**Задача 3: За отличен.** Добавяне на бутон със следната функционалност: в текстовата кутия където са изведени името и номера на кандидата стила на шрифта да се промени на  ***Bold***.

ПРИМЕРНА ЗАДАЧА ПО ВСП

Да се създаде Windows приложение за клиенти на фирма за дограма. Фирмата предлага три типа прозорци: двукрилен, трикрилен и прозорец с врата. В поръчката на клиента се включват:

Номер на клиента – 10 символа;

Име на клиента – 30 символа;

За всеки тип прозорец се задава :

* Брой прозорци от дадения тип;
* Единична цена на дадения тип прозорец;

Отстъпки:

* От 5 до 10 -5%;
* От 10 до 15 – 10%;
* Над 15 прозореца – 15%

Задача 1:

Да се създаде форма от две области:

Област входни данни: включваща подходящи контроли за представяне на поръчката за един клиент. Отстъпките да се представят чрез радио бутони.Бутон New – за нулиране на всички текстови контроли във формата.

Област резултати, включваща:

Бутон изчисление-натискането на който да води до извеждане данните за клиентите в една текстова кутия, а стойностите на неговата поръчка в друга.

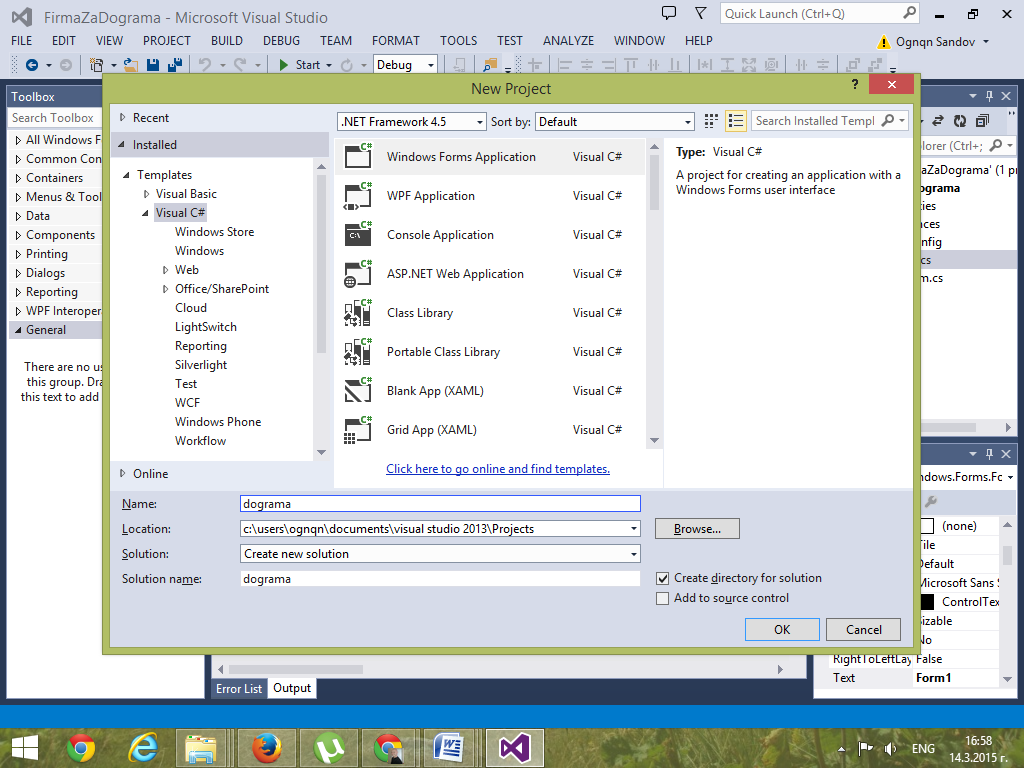
Задача 2:

Да се създаде метод за валидиране на входните данни. Бутон Изчисление се активира когато входните данни са въведени.

ФИРМА ЗА ДОГРАМА

2

1.Създава се ново Windows приложение.

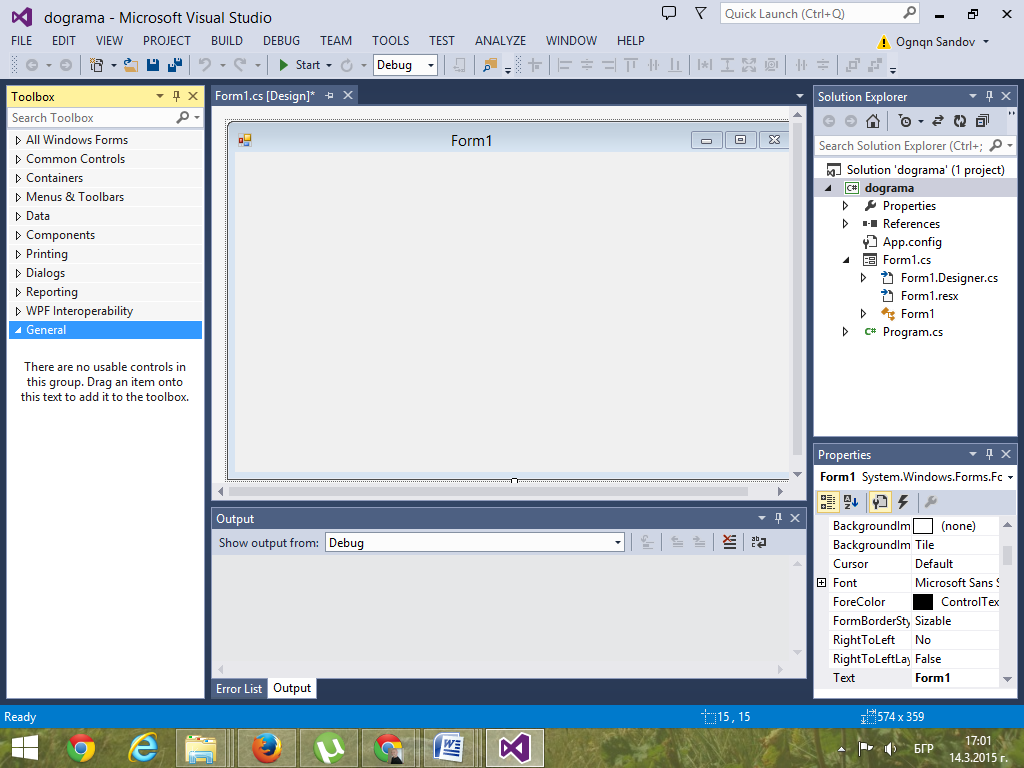


3

4

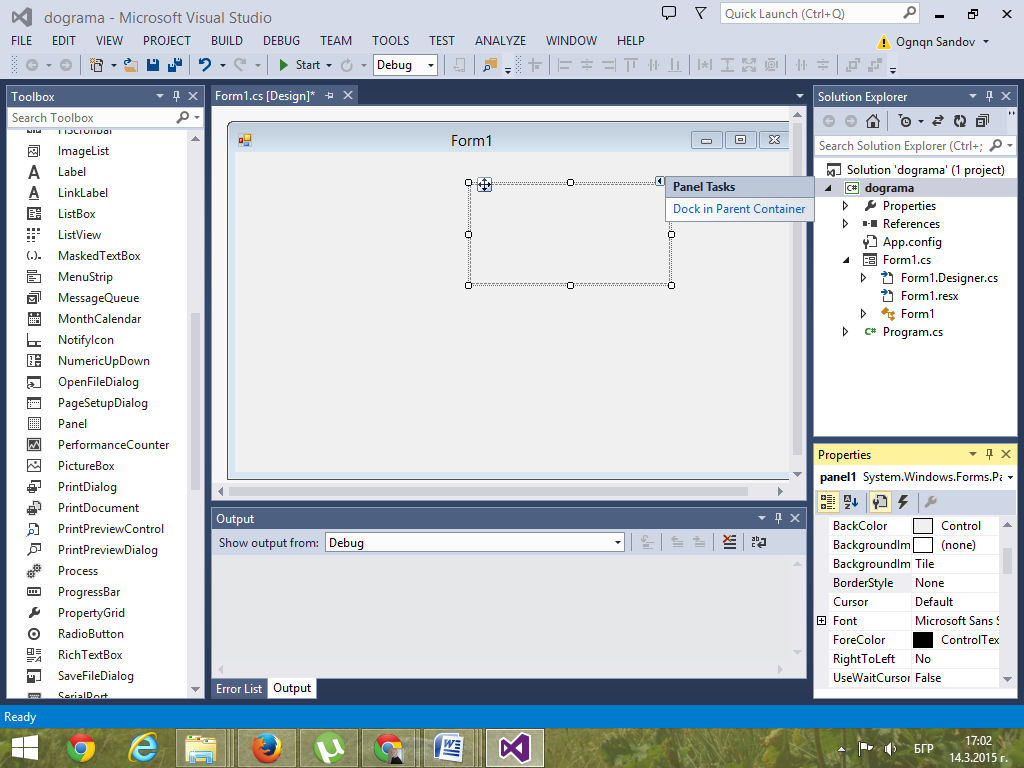
1

2. При правилно изпълнение на точка 1 трябва да виждате следващата картинка.



3. От прозореца ***Toolbox*** , чрез двоен клик или с влачене се поставя един панел във формата.

1



3

2

4. В прозореца ***Properties*** посредством свойството ***Dock*** задаваме позиция ***Top***на панела, което ще го накара да заеме горната част на формата. За целта трябва да е прихванат панела(4) и да свети прозорчето за свойствата(5) а не прозорчето за събитията (6).



4

6

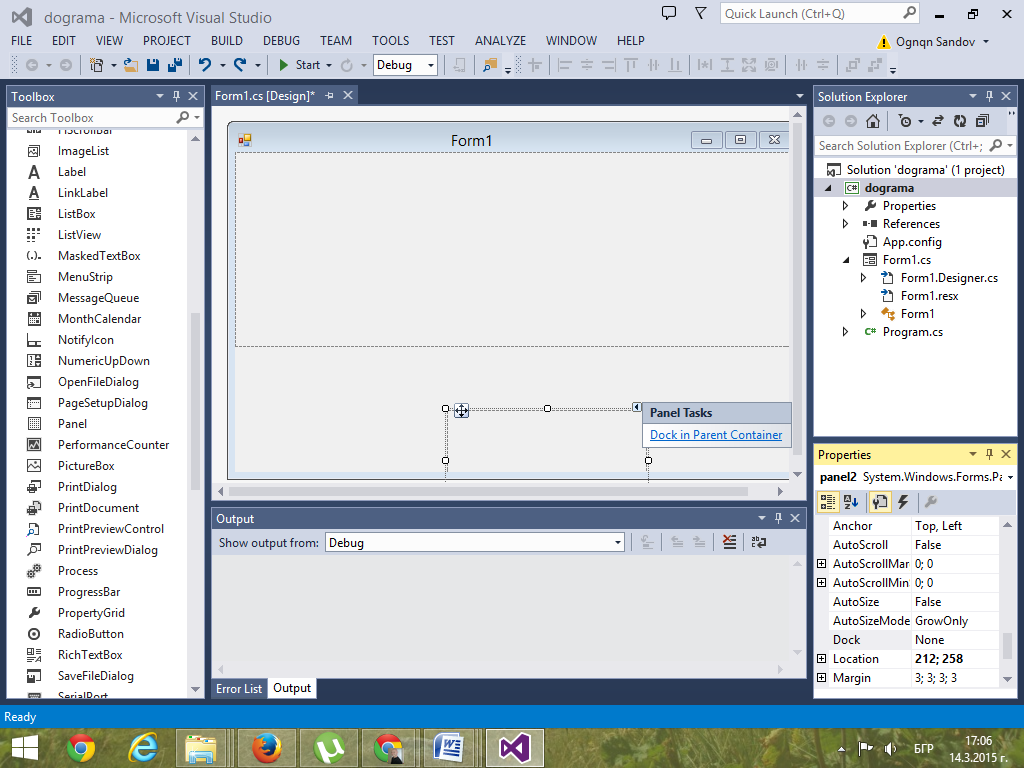
5

3

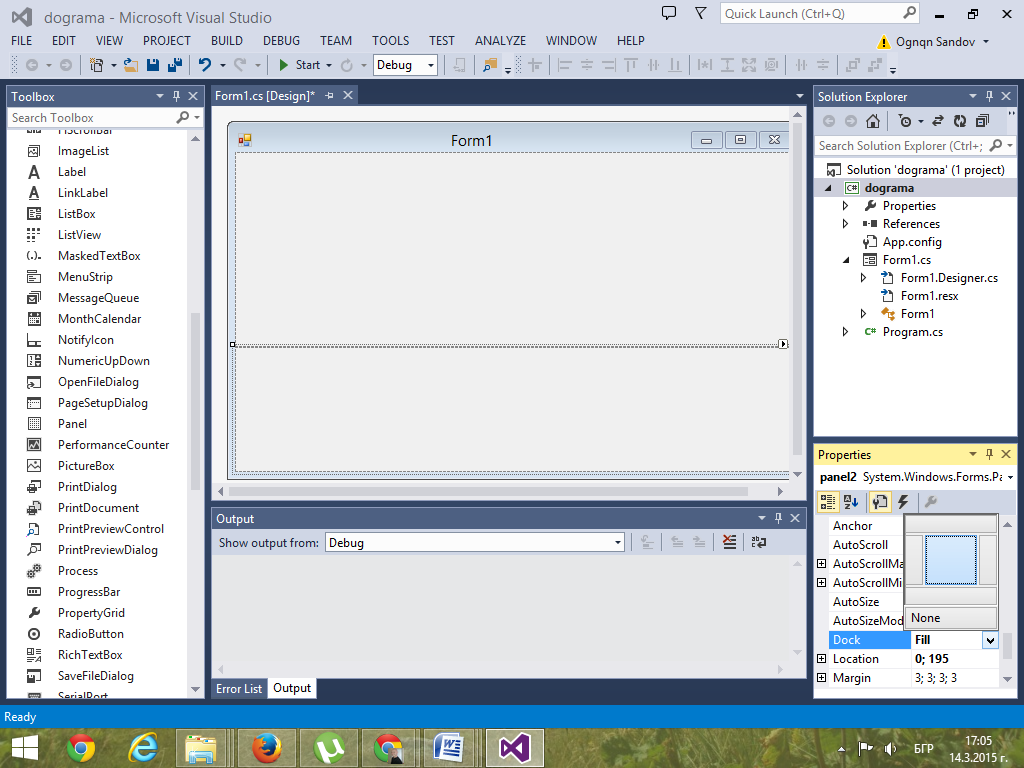
2

10

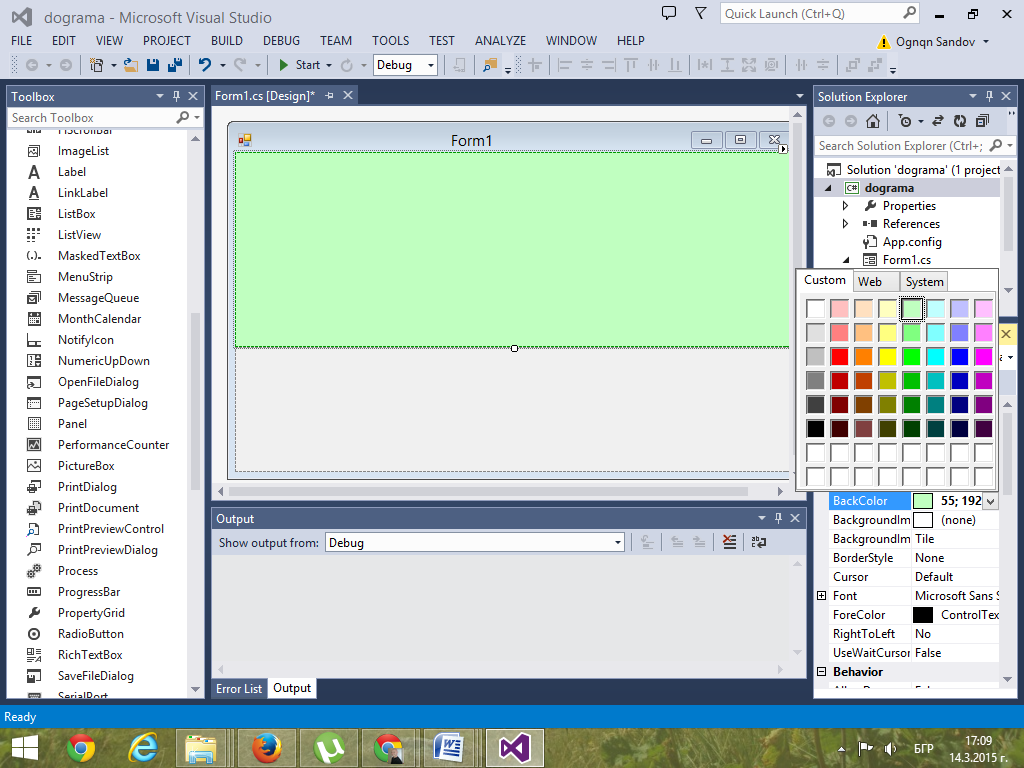
5.Следваща стъпка е да се постави още един панел във формата.



6. Свойството ***Dock*** на втория панел се установява на ***Fill*.**



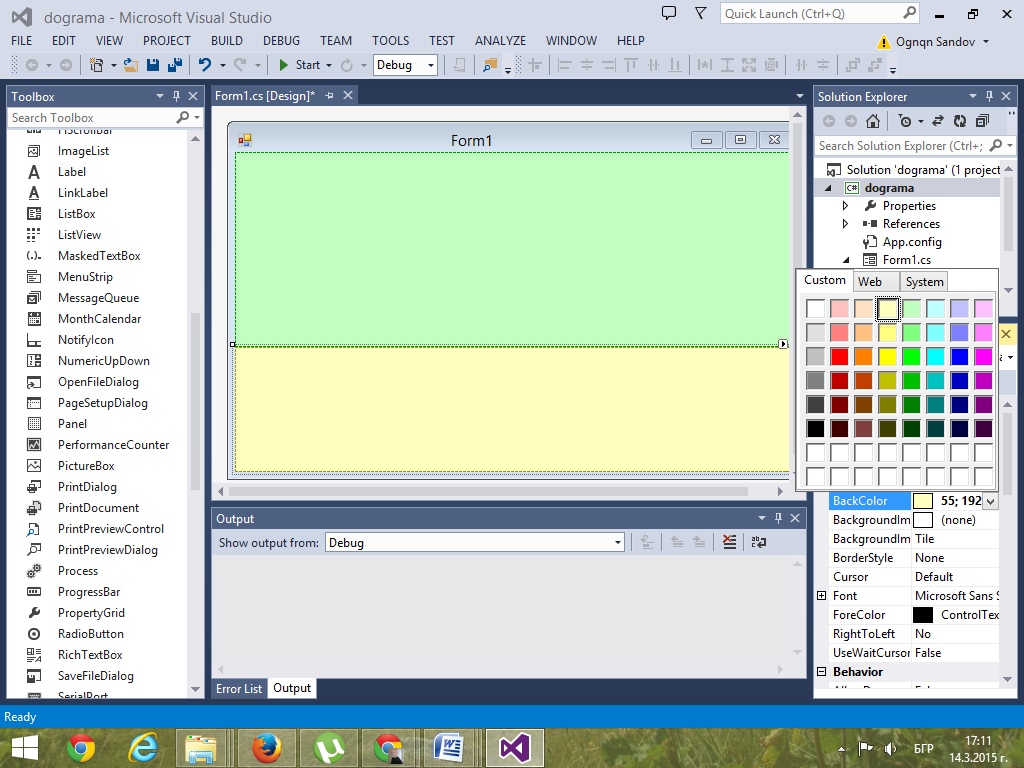
7.Пак в прозореца ***Properties*** посредством свойството ***Back Color*** се избират цветове за двата панела.



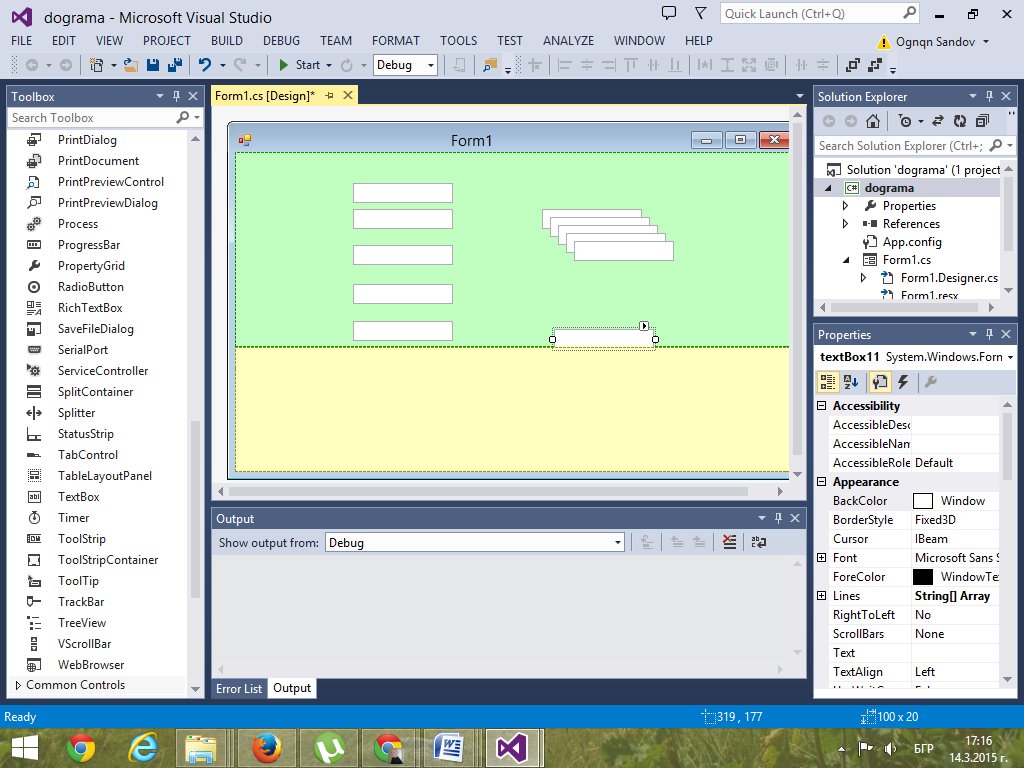
2

1

3

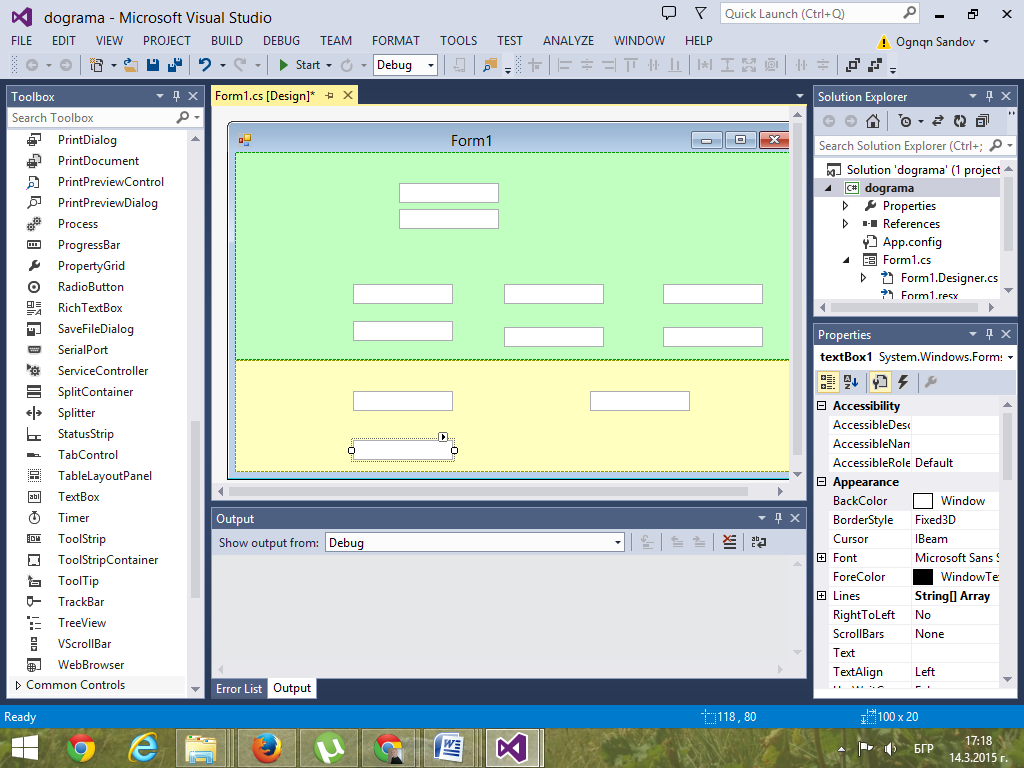


8. От прозореца ***Toolbox*** посредством двоен клик или влачене във формата се поставят нужния брой ***TextBox*** контроли. В случая 10.

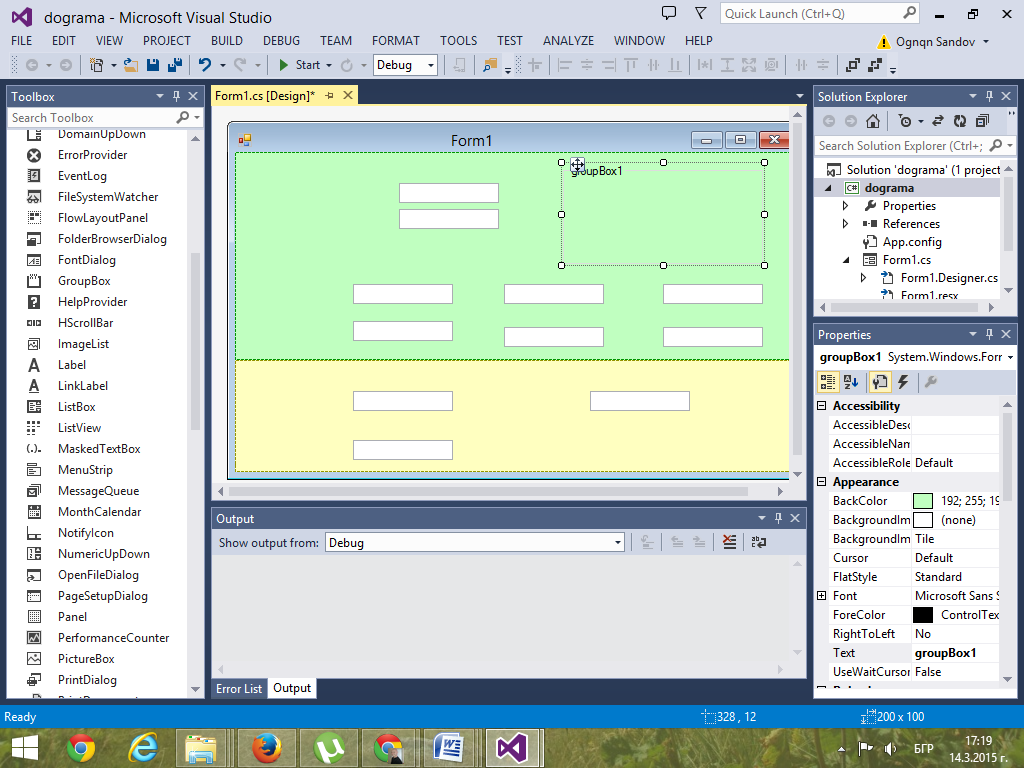


1

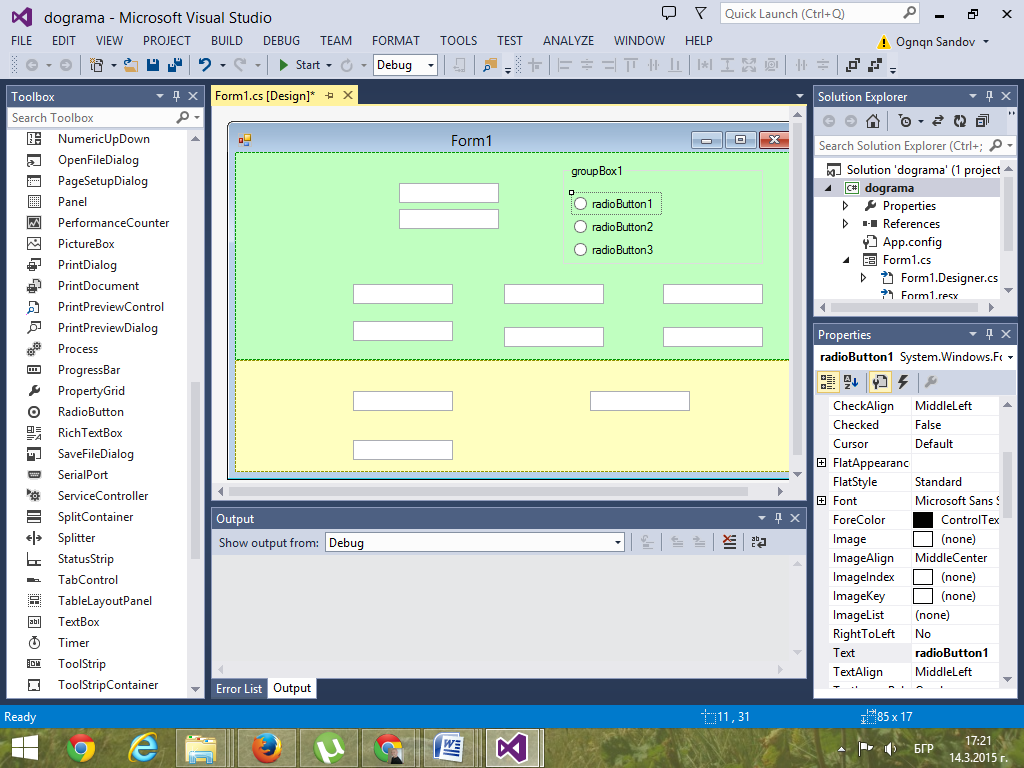
9. ***TextBox*** контролите се поставят на подходящи позиции върху двата панела.



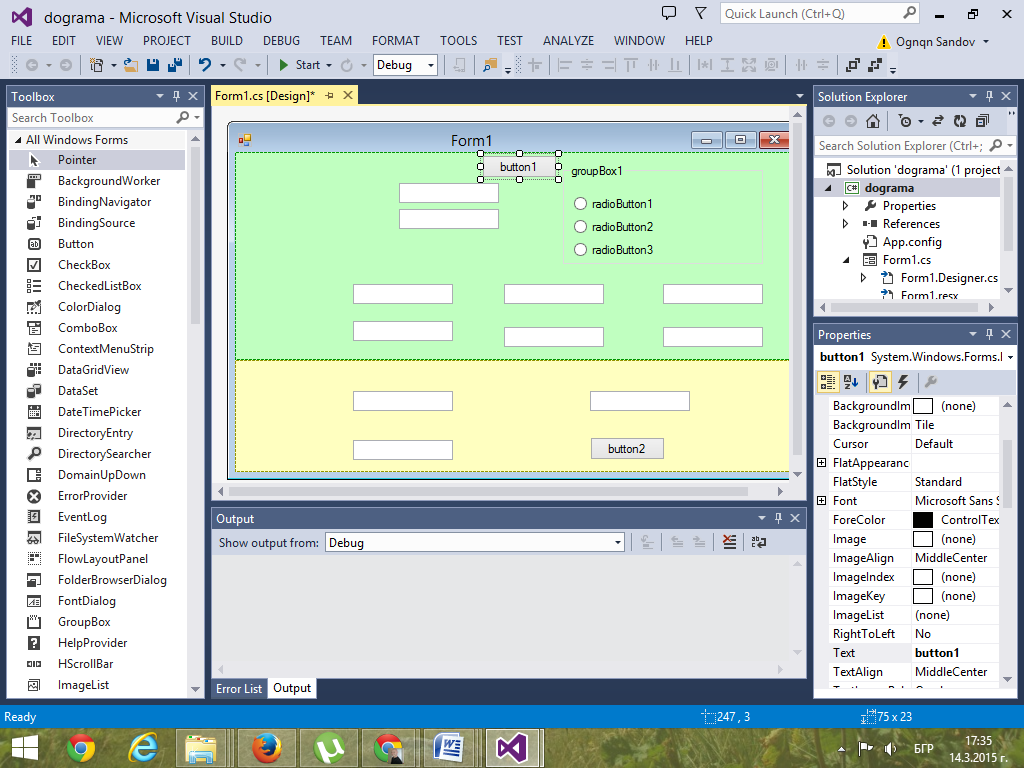
10. От прозореца ***Toolbox*** посредством двоен клик или влачене във формата на подходяща позиция се поставя един ***GroupBox*** контейнер в който да се поставят ***RadioButton*** или ***CheckBox*** контроли.



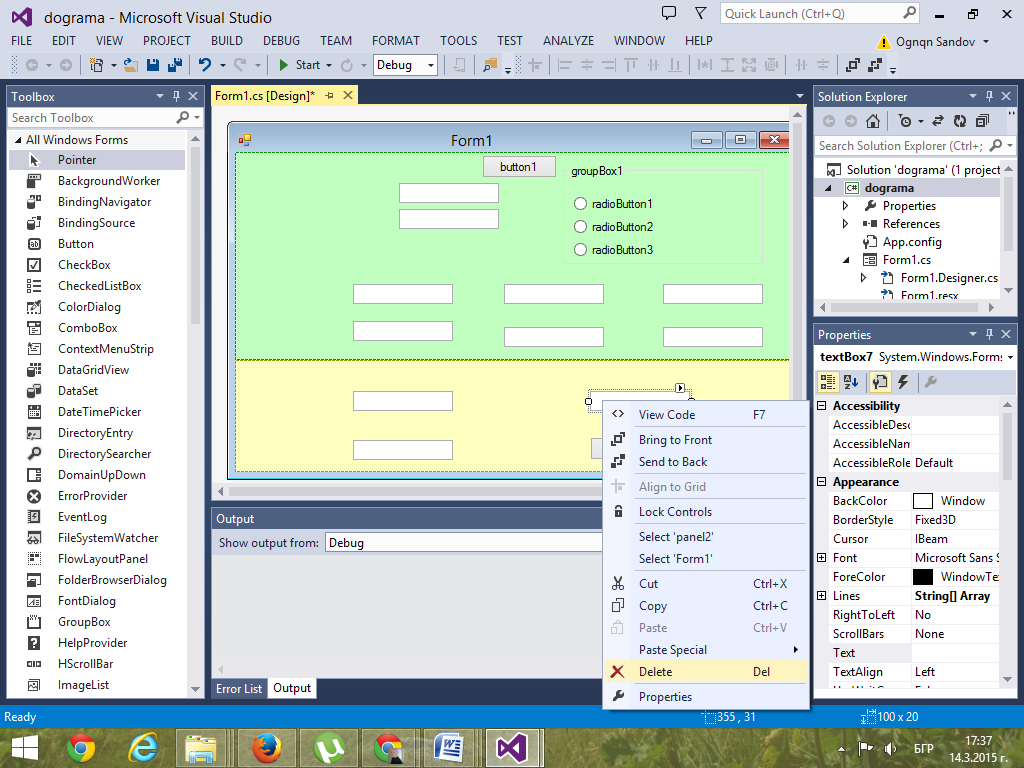
11. В GroupBox се поставят три радио бутона.



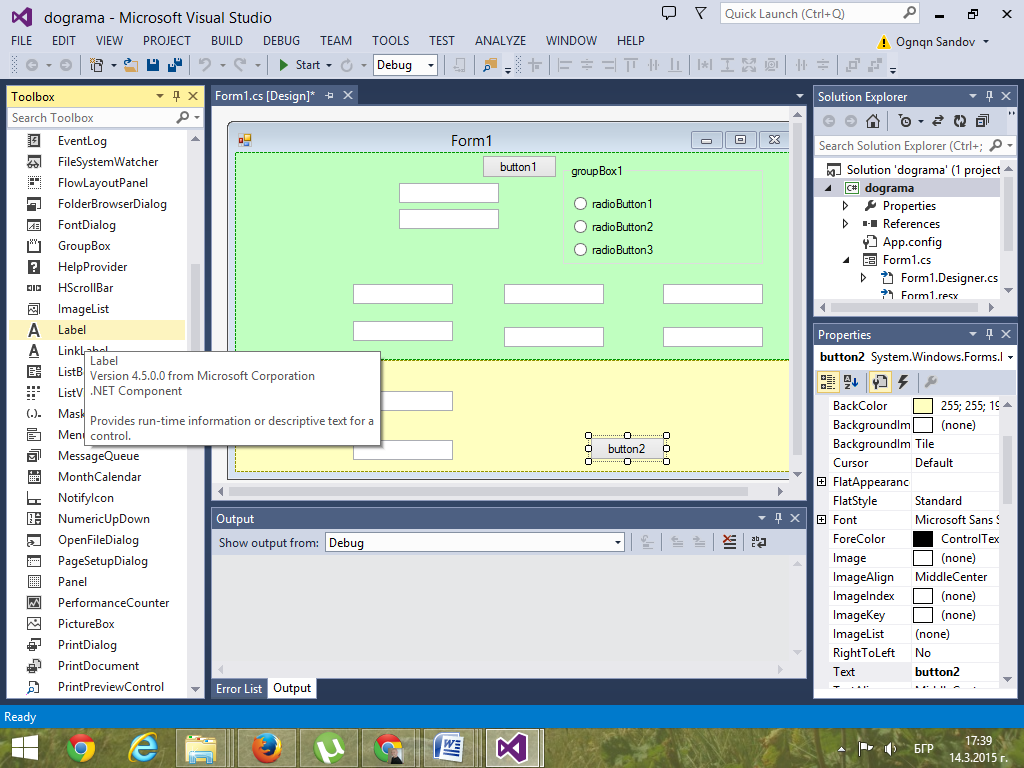
12. Поставят се два бутона.



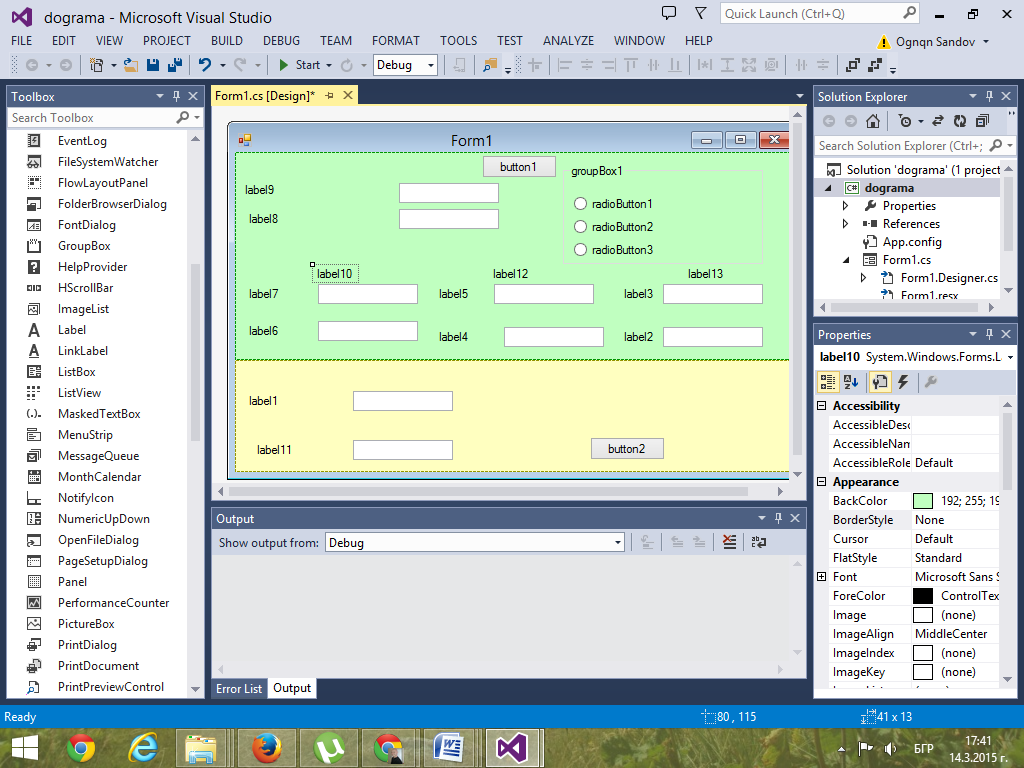
Ако във формата случайно е поставено нещо излишно посредством клик с десен бутон на мишката върху излишният елемент се избира ***Delete***.



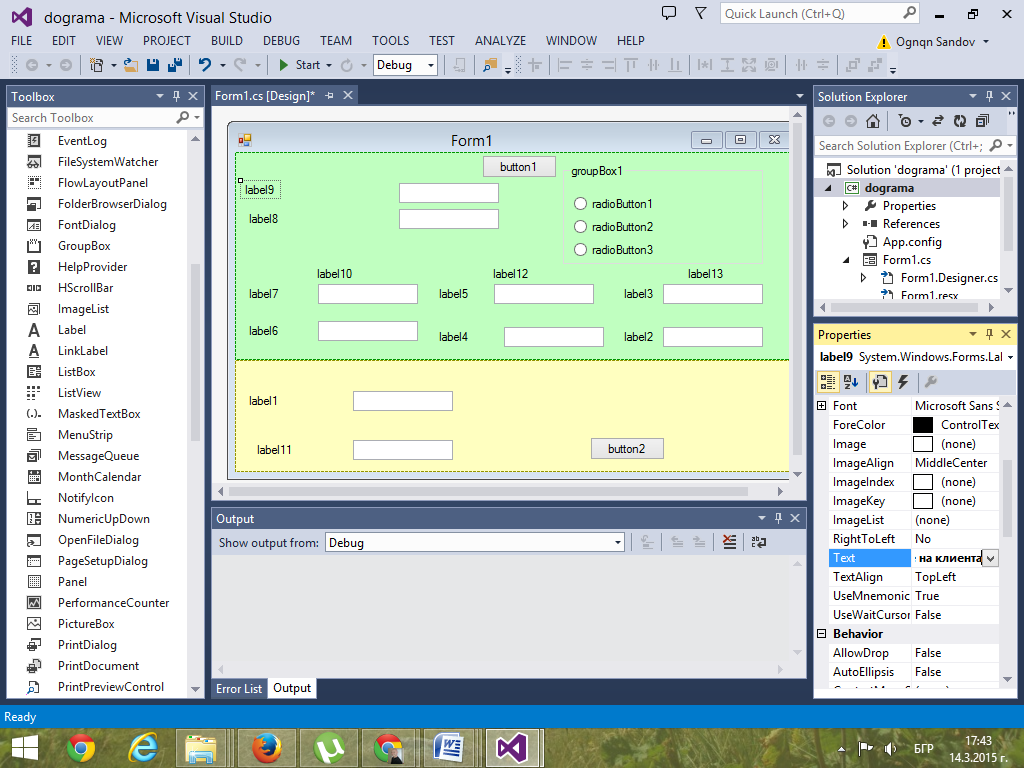
13. За да бъде описано какво трябва да бъде записано или какво трябва да изведе за всеки ***TextBox*** контрол се поставя по един **Label** контрол.



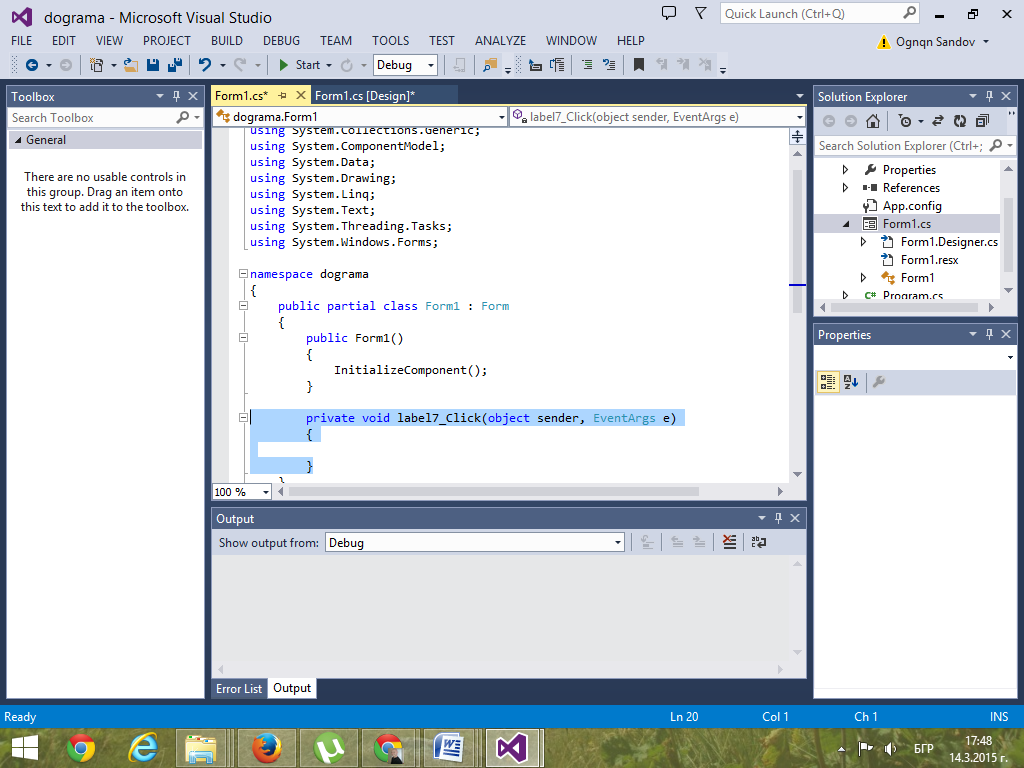
Ако е необходимо както в случая и повече.



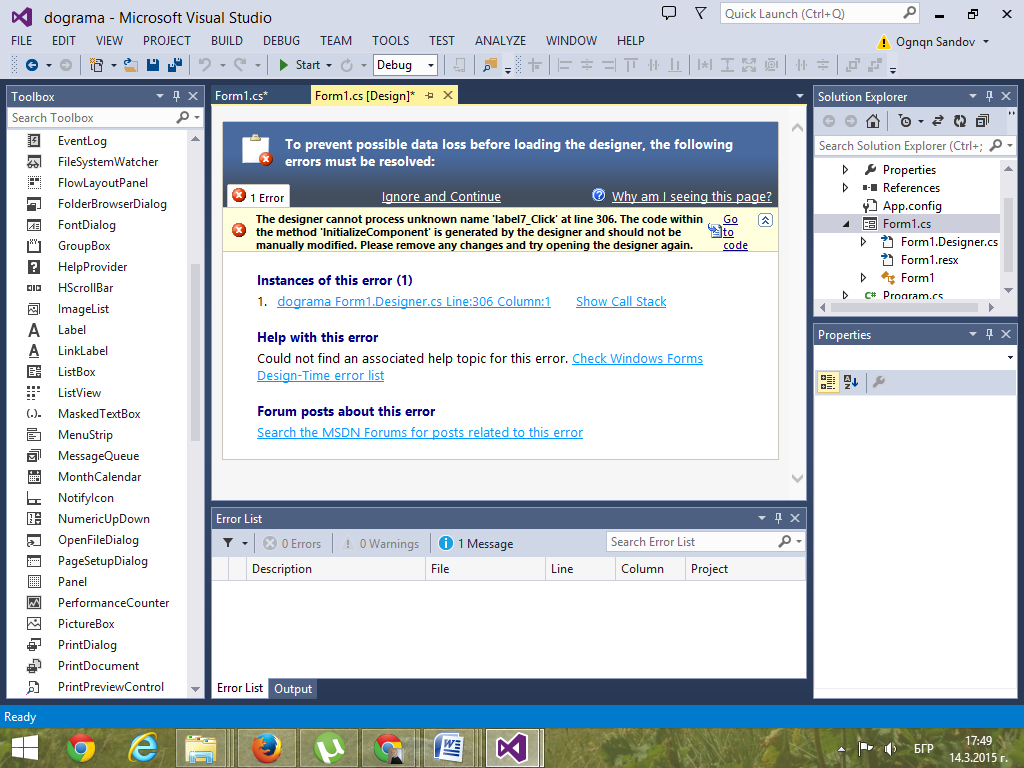
14. Контрол ***Label*** изобразява текст във формата. Посредством свойството ***Text*** на контрола от ***Properties*** прозореца се задава какъв текст ще изобразява ***Label*** контролата.



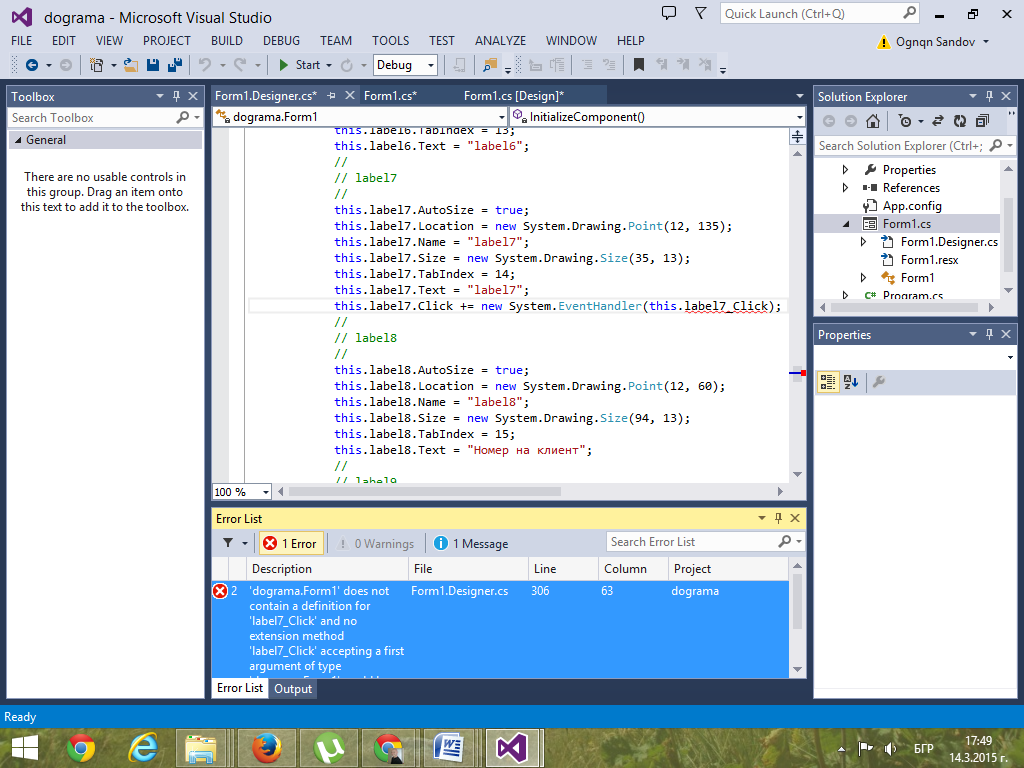
При случаен двоен клик върху някоя контрола се генерира код за обработване на събитие ***Click***.

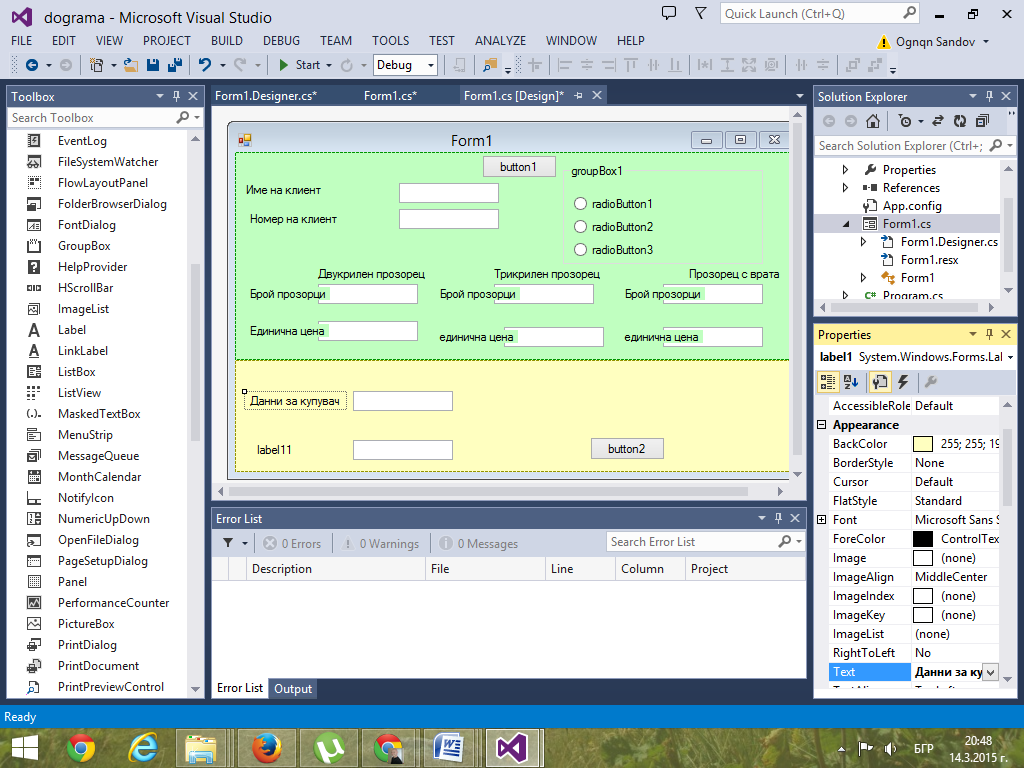


След изтриване на този код при опит за връщане към дизайна ще се генерира следната грешка.



При клик върху линка към грешката се отваря долният прозорец, където трябва да бъде изтрит целия ред с грешката.

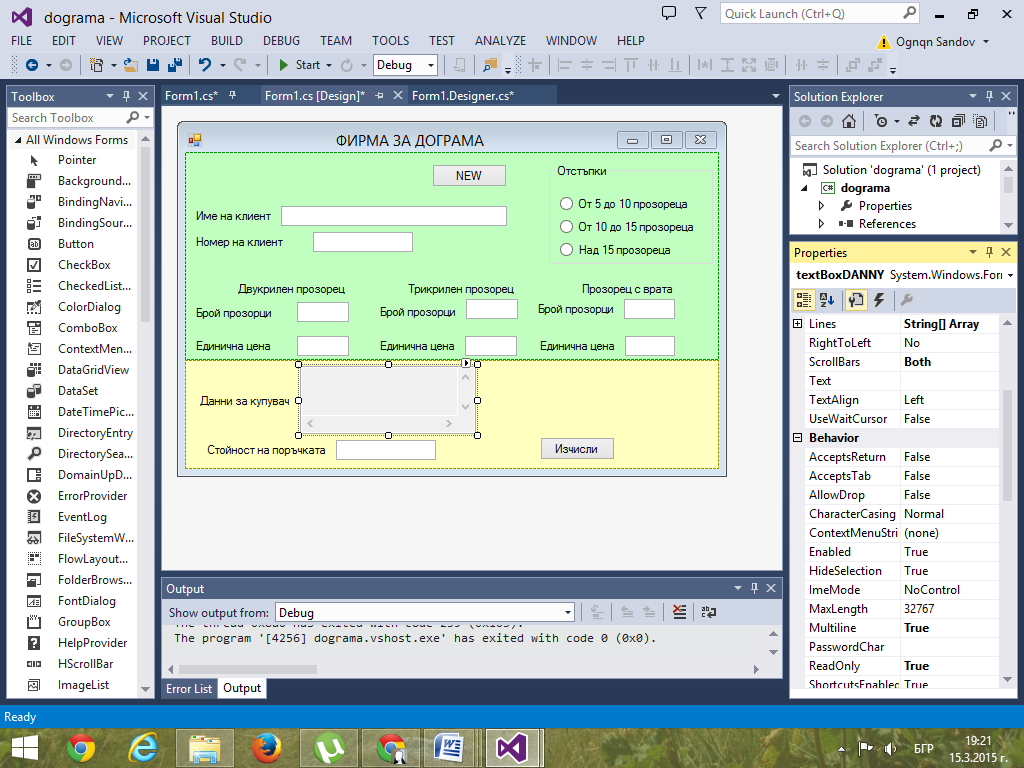


  
Чрез свойството текст задаваме текста на всички контроли.



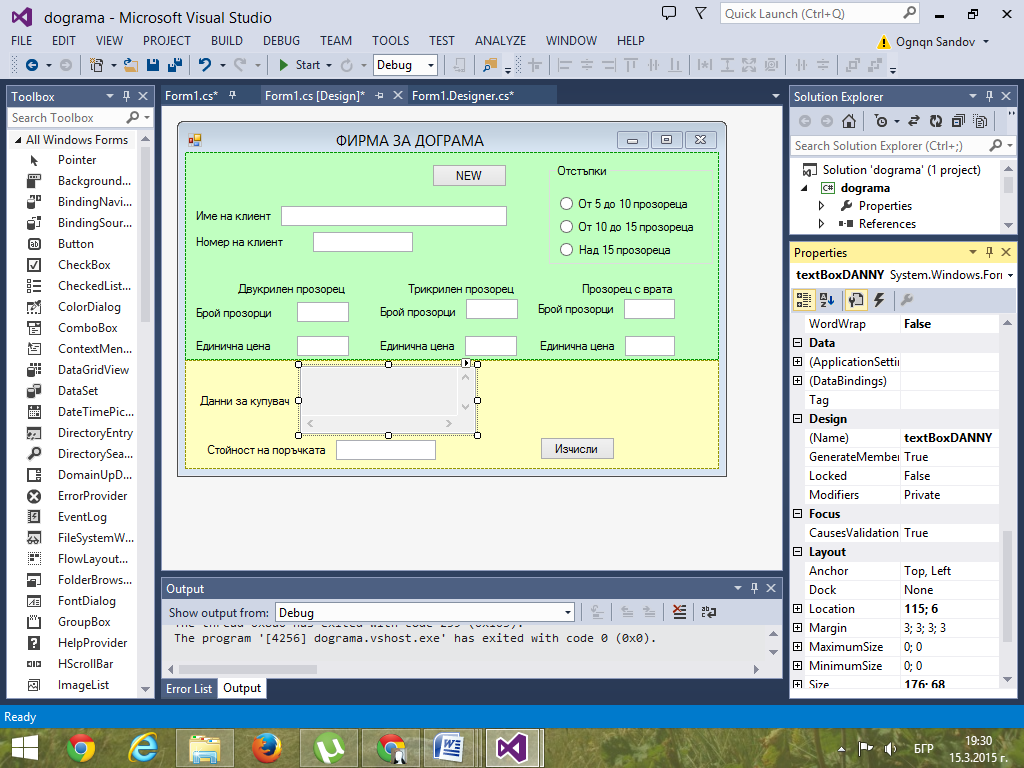
15.За ***TextBox*** контрол за Данни за купувача променяме свойството ***Multiline*** от ***False*** на ***True*** за да го променим от едноредов на многоредов .След това се разширява височината на контрола, което може да се реализира по два начина: чрез изтегляне на вертикалния размер с помощта на мишката или чрез задаване на конкретна стойност на свойството ***Size.Height***. Свойството ***ReadOnly*** се установява в режим ***True***, за да стане недостъпно въвеждането на текст от потребителя.

Свойството ***Scrollbars*** се установява в режим ***Both,*** в резултат на което се появяват хоризонтален и вертикален скролбар.

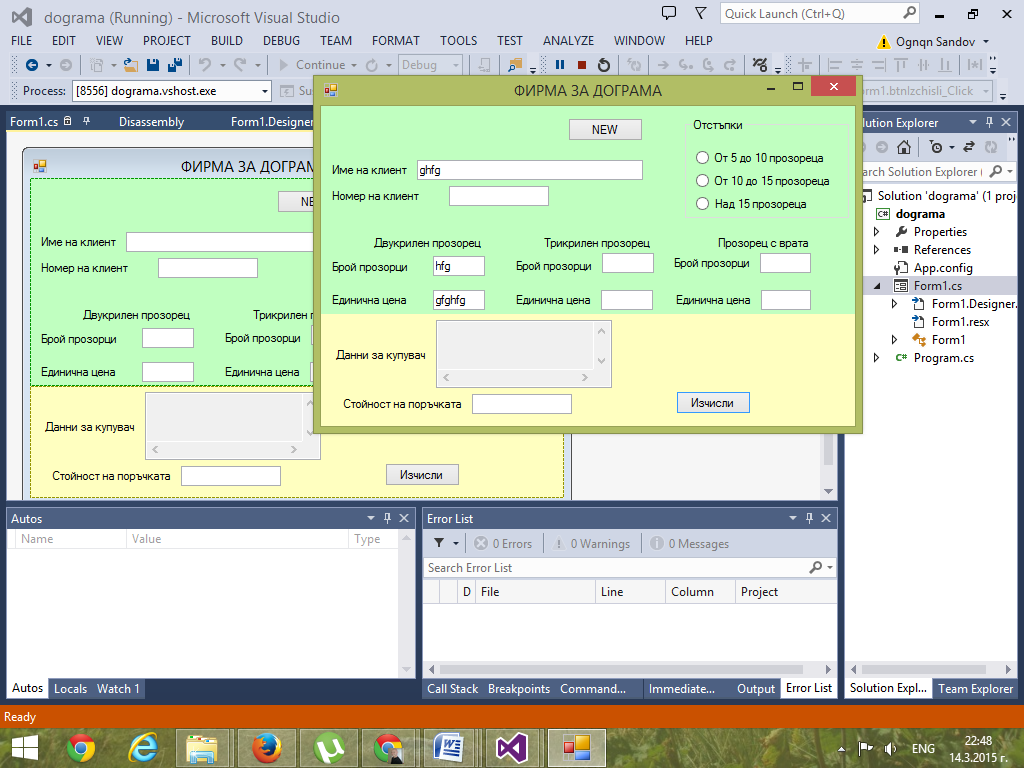


Свойството ***ReadOnly*** се установява в режим ***True,*** за да стане недостъпно въвеждането на текст от потребителя и за ***TextBox*** контрола за Стойност на поръчката.

16. Посредством свойството (***Name***) се дават подходящи (не много дълги но да подсказват за какво точно се отнасят) имена на всички контроли.



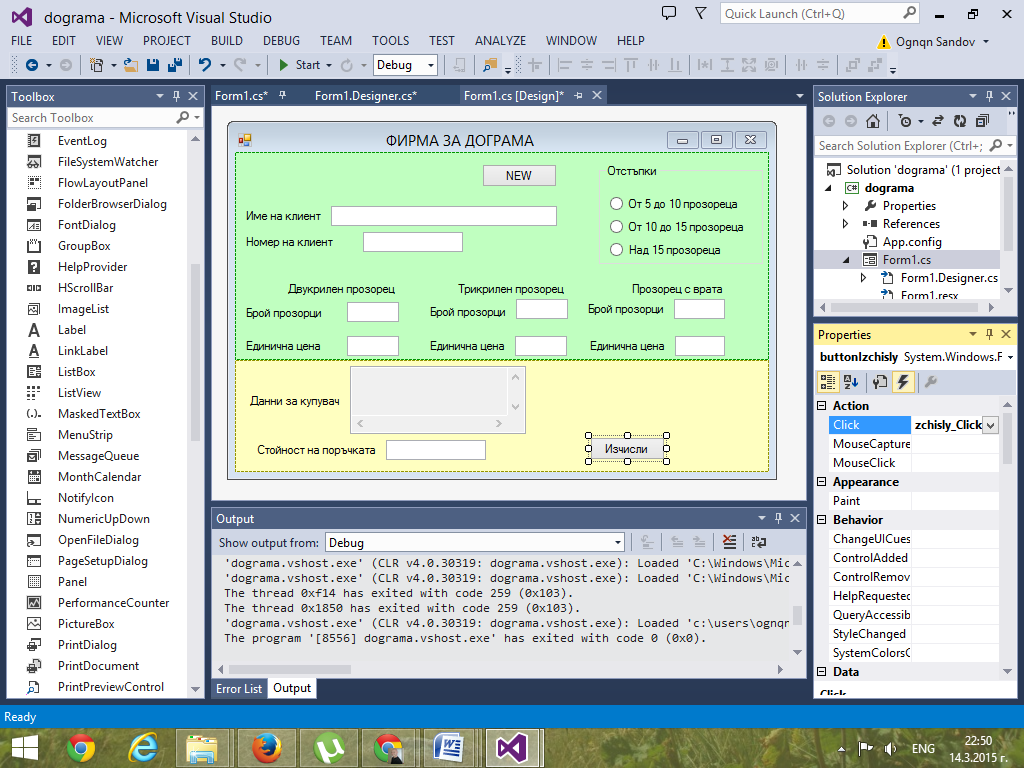
След тези операции при старт трябва да се получи долната картинка.Така формата е завършена



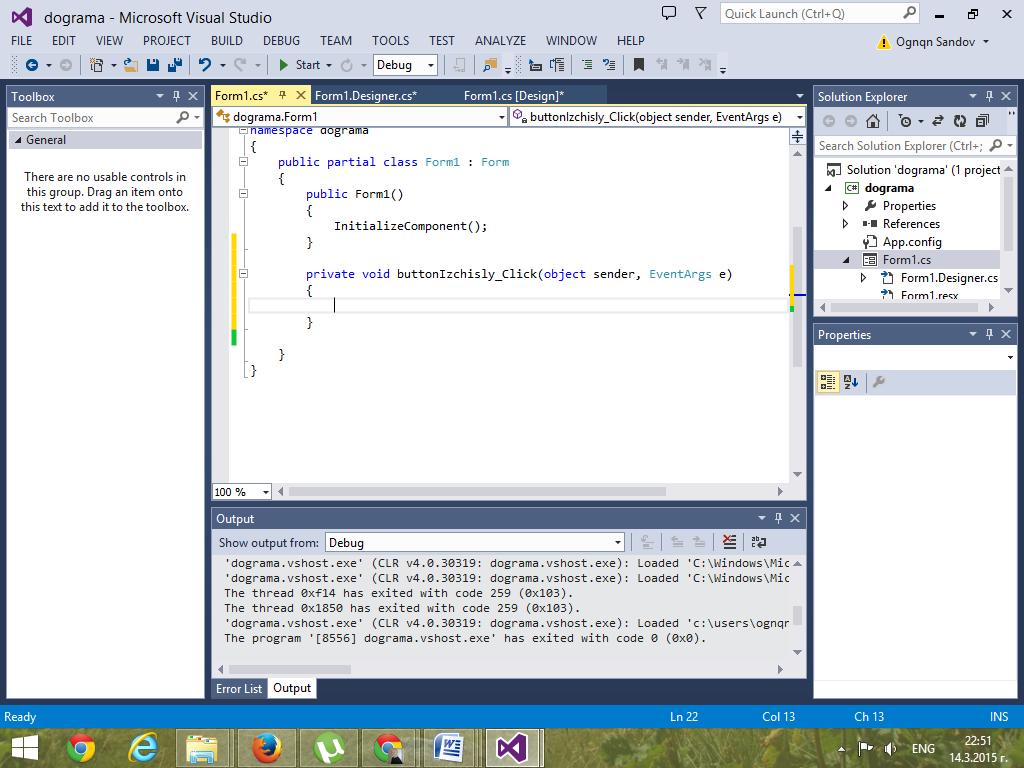
17. Добавяне на методи за обработване на събития.

Бутон изчисление-натискането на който да води до извеждане данните за клиентите в една текстова кутия, а стойностите на неговата поръчка в друга.

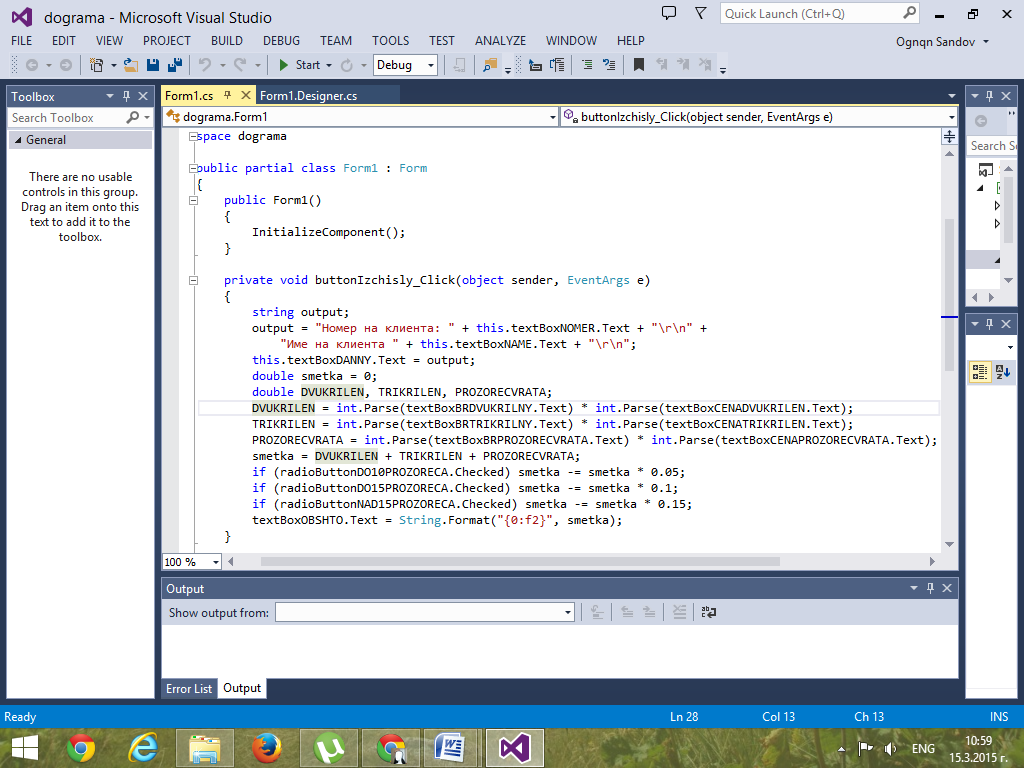
За добавяне на метода Click към даден бутон е необходимо да се щракне два пъти върху бутона или да се абонира за събитието в прозореца Events.



Това довежда до генериране на метода ***buttonIzchisly\_Click***().



За да изпълни условието на задачата към метода трябва да се добави следния код.



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

{

string output;

output = "Номер на клиента: " + this.textBoxNOMER.Text + "\r\n" +

"Име на клиента " + this.textBoxNAME.Text + "\r\n";

this.textBoxDANNY.Text = output;

double smetka = 0;

double DVUKRILEN, TRIKRILEN, PROZORECVRATA;

DVUKRILEN = int.Parse(textBoxBRDVUKRILNY.Text) \* int.Parse(textBoxCENADVUKRILEN.Text);

TRIKRILEN = int.Parse(textBoxBRTRIKRILNY.Text) \* int.Parse(textBoxCENATRIKRILEN.Text);

PROZORECVRATA = int.Parse(textBoxBRPROZORECVRATA.Text) \* int.Parse(textBoxCENAPROZORECVRATA.Text);

smetka = DVUKRILEN + TRIKRILEN + PROZORECVRATA;

if (radioButtonDO10PROZORECA.Checked) smetka -= smetka \* 0.05;

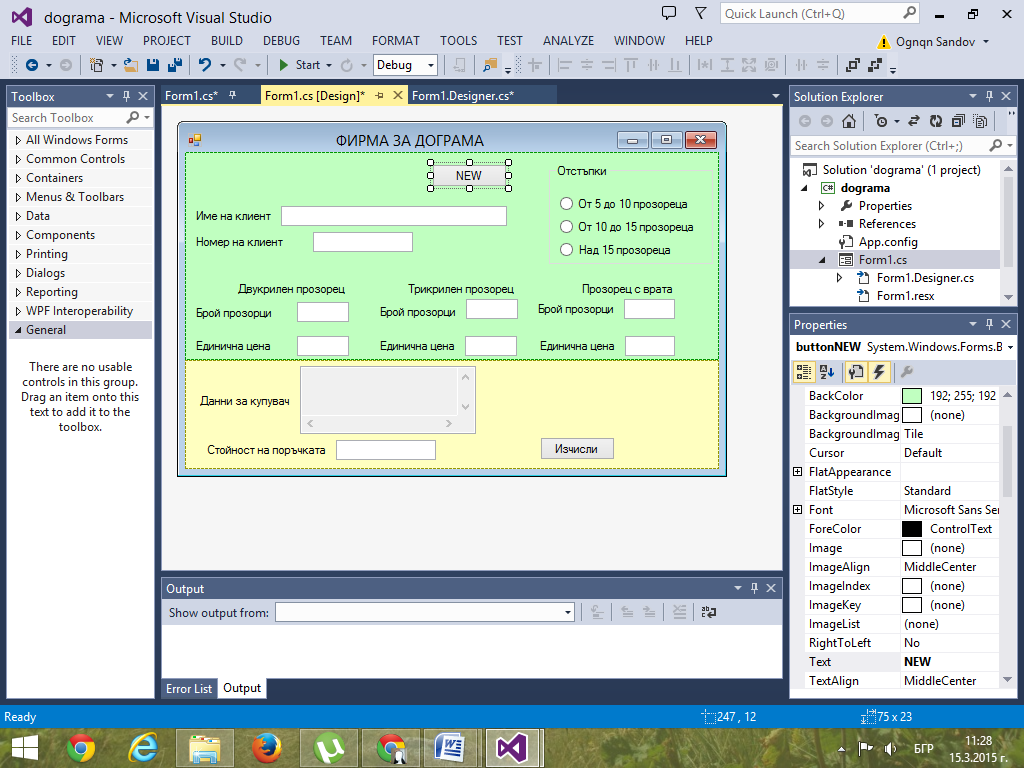
if (radioButtonDO15PROZORECA.Checked) smetka -= smetka \* 0.1;

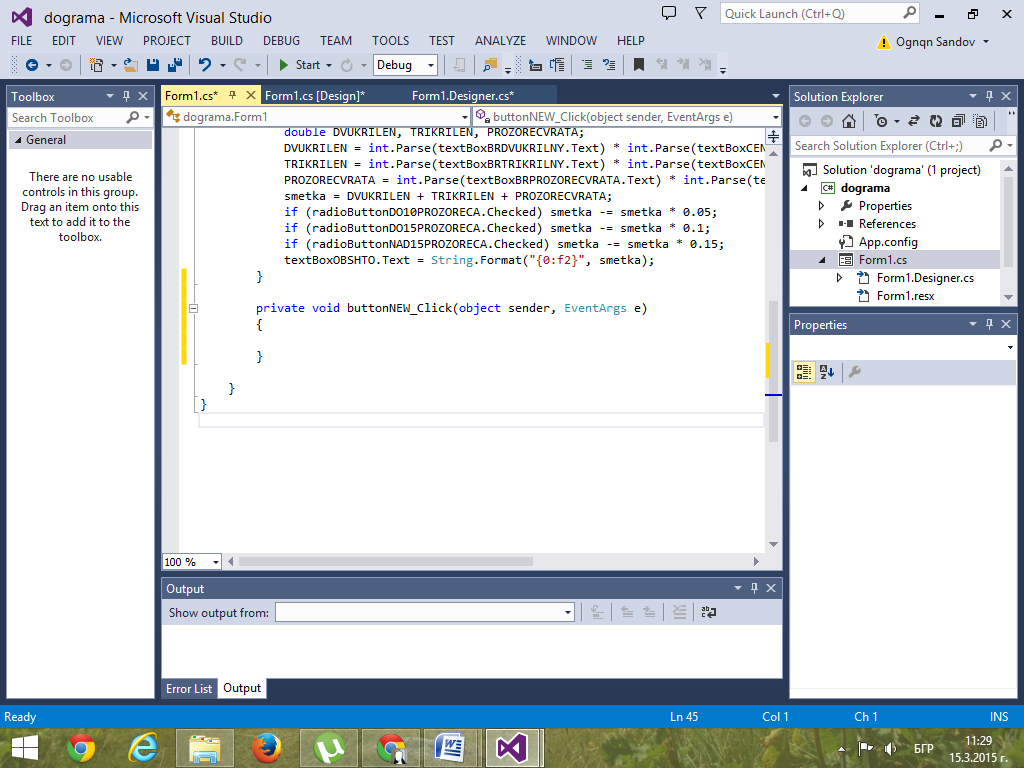
if (radioButtonNAD15PROZORECA.Checked) smetka -= smetka \* 0.15;

textBoxOBSHTO.Text = String.Format("{0:f2}", smetka);

}

18. Добавяне на метод ***Click*** *и за* ***buttonNEW***





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

{

TextBox txt = sender as TextBox;

foreach (Control ctrl in panel1.Controls)

{

txt = ctrl as TextBox;

if (txt != null)

{

txt.Text = "";

txt.BackColor = SystemColors.Window;

txt.Tag = false;

}

}

textBoxDANNY.Clear();

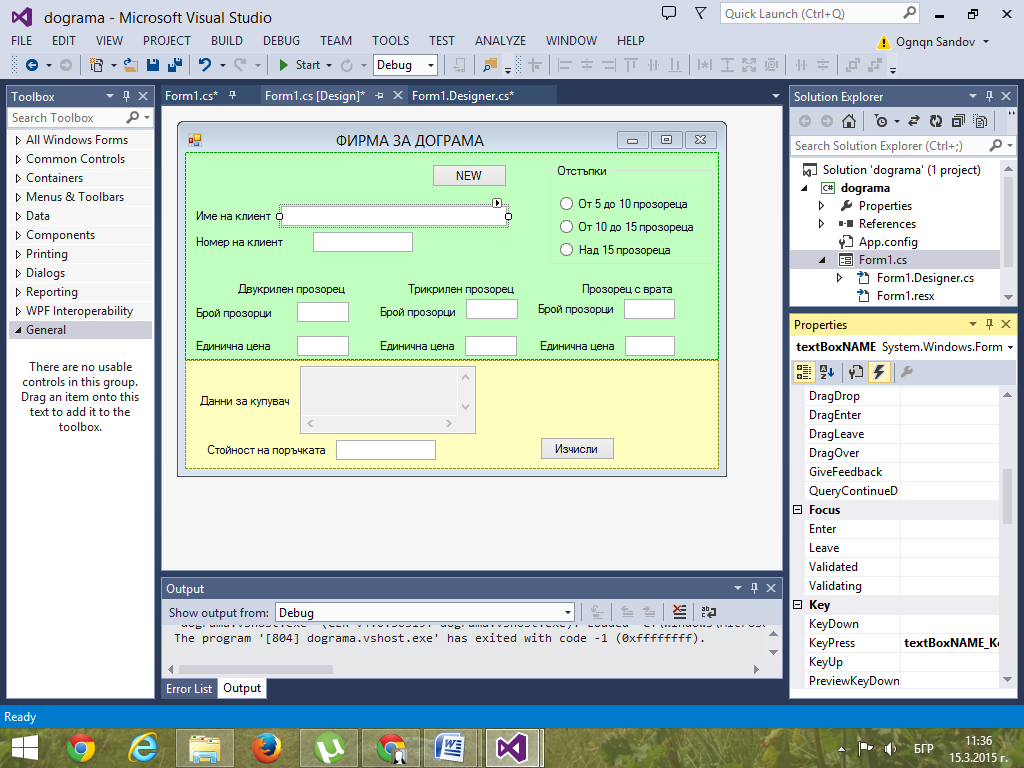
textBoxOBSHTO.Clear();

}

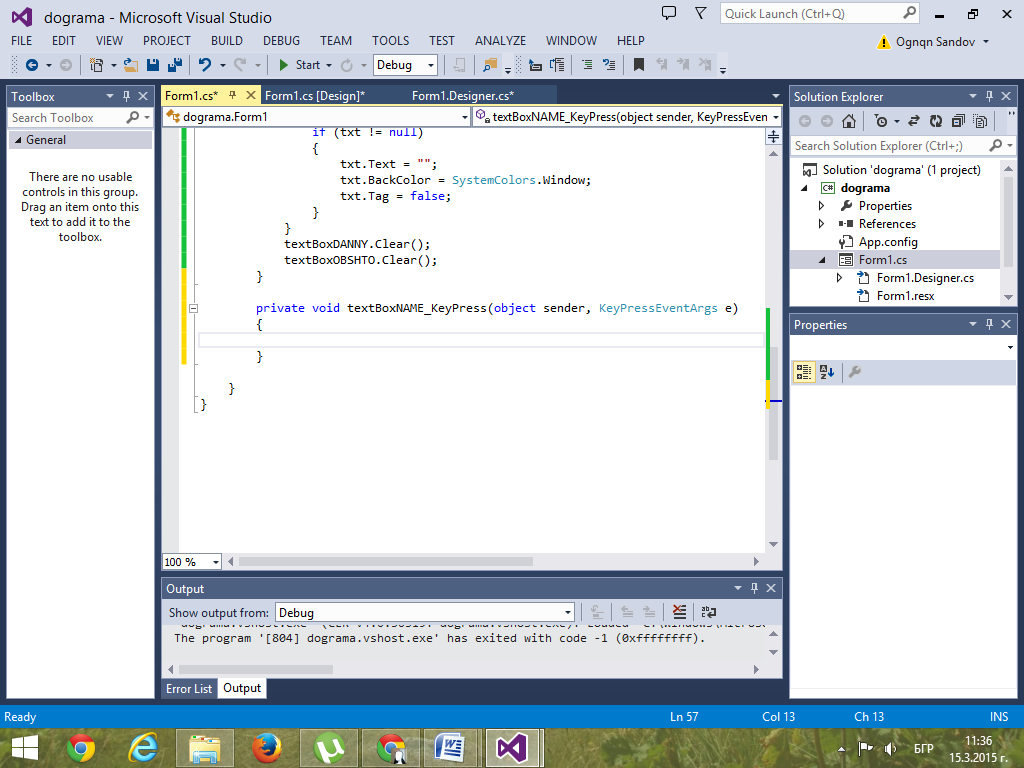
19. Създаване на метод ***KeyPress*** за валидация на полетата.

* метод ***KeyPress*** за текстовото поле ***textBoxNAME***

Във формата избираме текстовото поле.След това се извършва абониране за събитието посредством двойно щракване върху събитието ***KeyPress*** в списъка ***Events*** на прозореца ***Properties*** .



В метода за валидиране



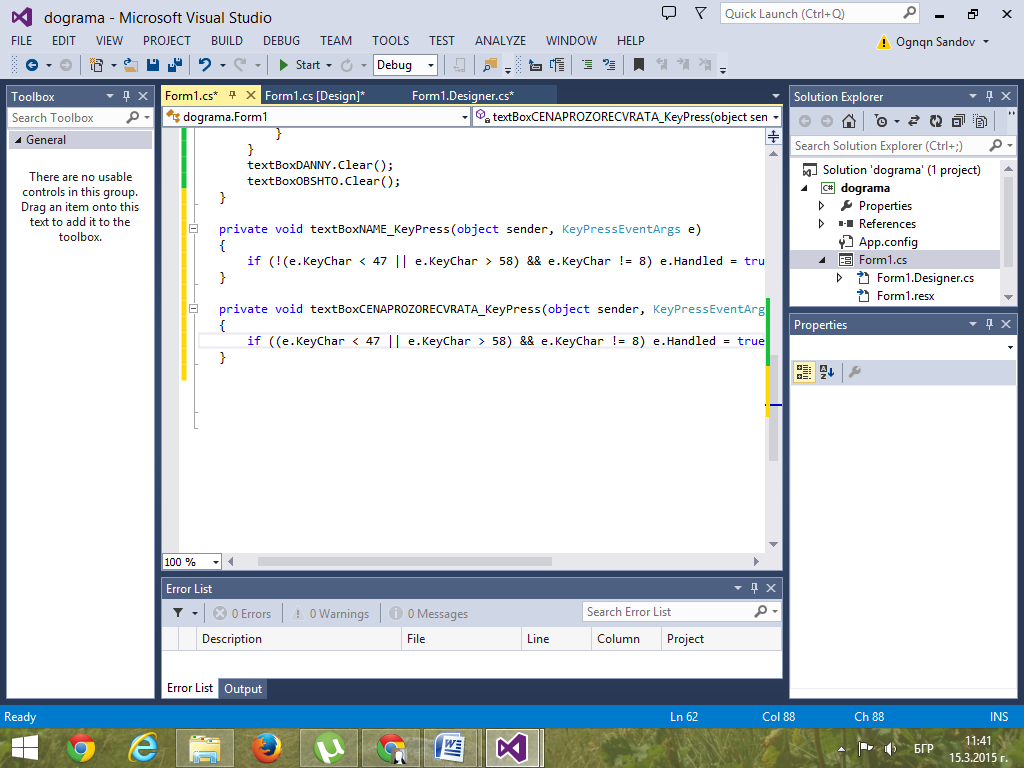
се добавя следния код чрез който се забранява въвеждането на цифри в textBoxNAME

private void textBoxNAME\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar < 47 || e.KeyChar > 58) && e.KeyChar != 8) e.Handled = true;

}



* метод ***KeyPress*** за валидиране на полетата за въвеждане на брой прозорци единична цена и номер на клиента.

Във формата посредством бутон Ctrl от клавиатурата избираме съответните текстови полета.След това се извършва абониране за събитието посредством двойно щракване върху събитието ***KeyPress*** в списъка ***Events*** на прозореца ***Properties*** . В метода ще се изпише само името на първия избран елемент. Добавя се следния код чрез който се забранява въвеждането на знаци различни от цифри и запетаики(десетична запетайка).

private void textBoxCENAPROZORECVRATA\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar < 48 || e.KeyChar > 58) && e.KeyChar != 8 && e.KeyChar != 44) e.Handled = true;

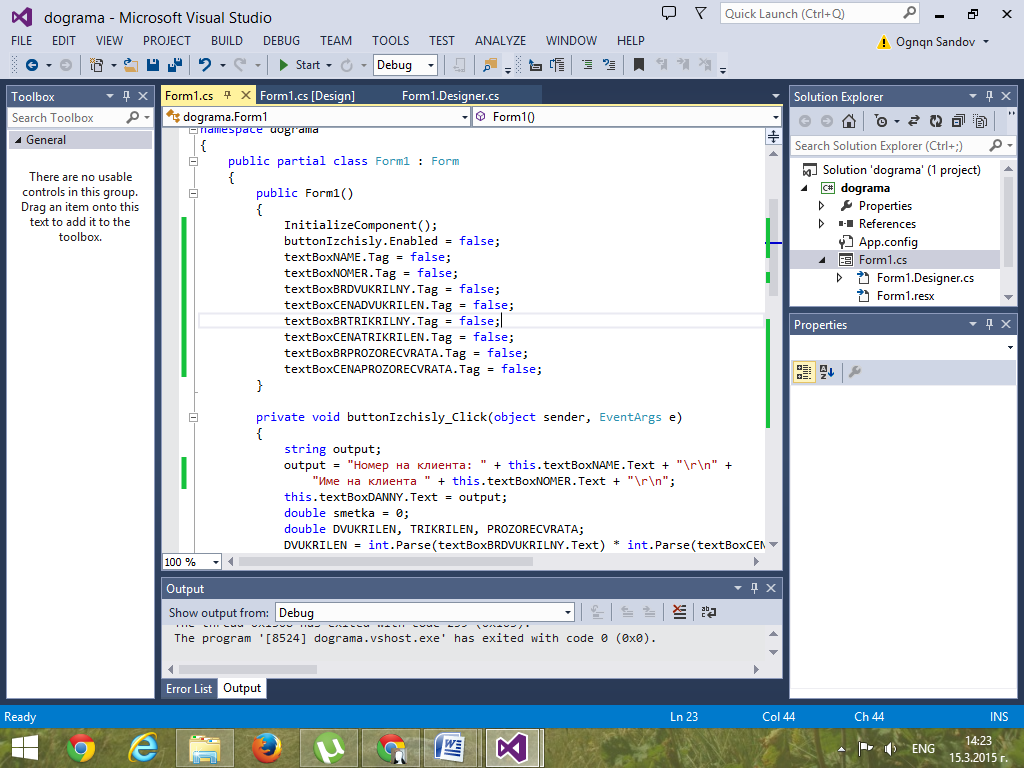
}

20. За да не се допуска изпълнение на метода ***buttonIzchisly\_Click()*** преди въвеждане на необходимите данни след метода ***InitializeComponent() се въвежда следния код.***



21. Създаване на метод ***Validating()***

За определяне състоянието на контролите ще се използва свойството Tag. В конструктора се добавят следните редове:



public Form1()

{

InitializeComponent();

buttonIzchisly.Enabled = false;

textBoxNAME.Tag = false;

textBoxNOMER.Tag = false;

textBoxBRDVUKRILNY.Tag = false;

textBoxCENADVUKRILEN.Tag = false;

textBoxBRTRIKRILNY.Tag = false;

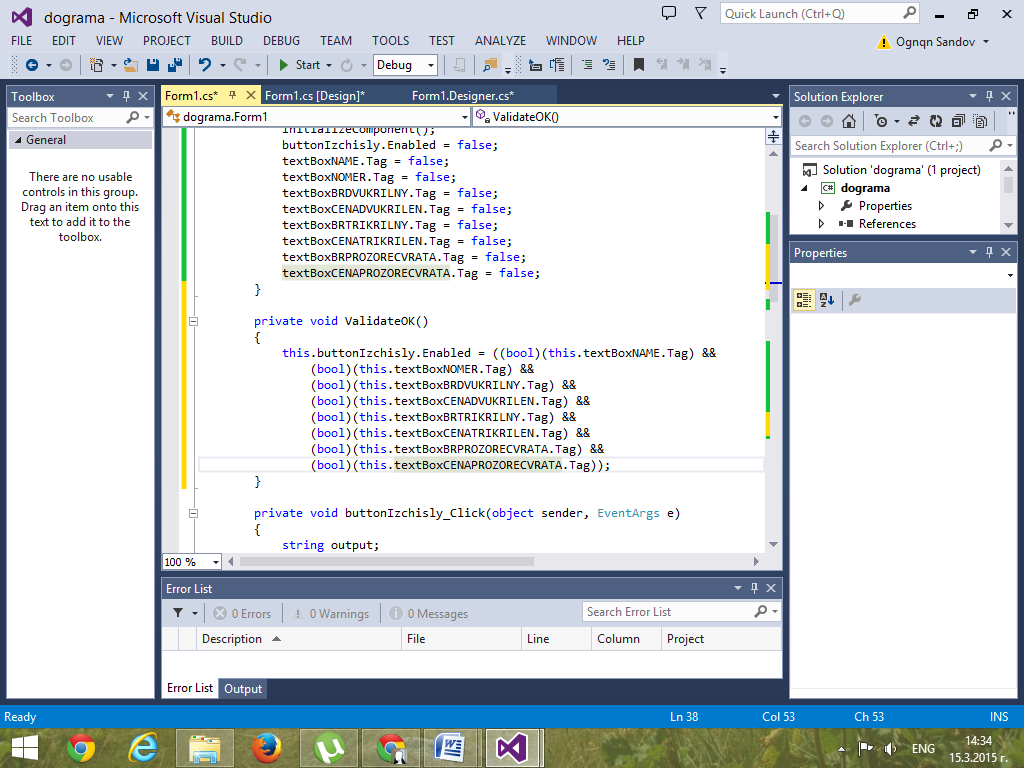
textBoxCENATRIKRILEN.Tag = false;

textBoxBRPROZORECVRATA.Tag = false;

textBoxCENAPROZORECVRATA.Tag = false;

}

Метод който ще проверява състоянията за свойството Tag:



private void ValidateOK()

{

this.buttonIzchisly.Enabled = ((bool)(this.textBoxNAME.Tag) &&

(bool)(this.textBoxNOMER.Tag) &&

(bool)(this.textBoxBRDVUKRILNY.Tag) &&

(bool)(this.textBoxCENADVUKRILEN.Tag) &&

(bool)(this.textBoxBRTRIKRILNY.Tag) &&

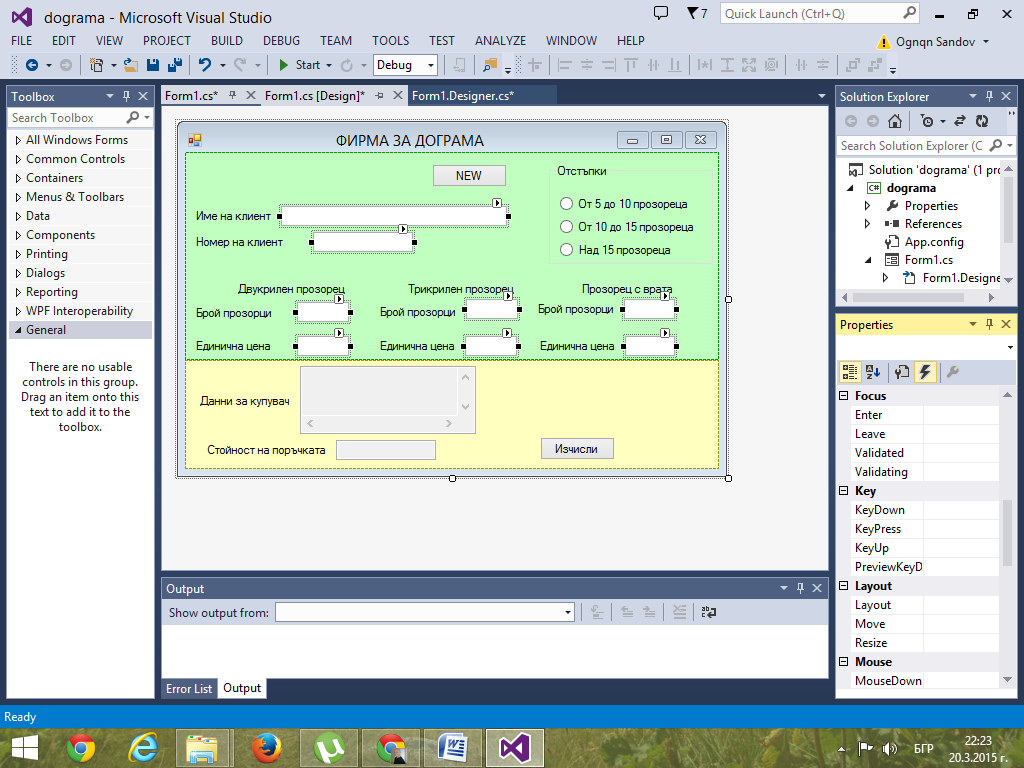
(bool)(this.textBoxCENATRIKRILEN.Tag) &&

(bool)(this.textBoxBRPROZORECVRATA.Tag) &&

(bool)(this.textBoxCENAPROZORECVRATA.Tag));

}

Маркират се всички текстови полета в които ще се въвеждат данни и посредством двойно щракване на събитието ***Validating*** списъка ***Events*** на прозореца ***Properties се*** абонират за събитието.



В генерирания метод добавяме следния код:

private void textBoxNAME\_Validating(object sender, CancelEventArgs e)

{

TextBox tb = (TextBox)sender;

if (tb.Text.Length == 0)

{

tb.BackColor = Color.Red;

tb.Tag = false;

}

else

{

tb.BackColor = System.Drawing.SystemColors.Window;

tb.Tag = true;

}

ValidateOK();

}

С което задачата е изпълнена.

ЦЕЛИЯ КОД

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 dograma

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

buttonIzchisly.Enabled = false;//деактивиране на бутон Изчисли

//инициализиране на свойството Tag, предназначено запроверка допустимоста на данните

textBoxNAME.Tag = false;

textBoxNOMER.Tag = false;

textBoxBRDVUKRILNY.Tag = false;

textBoxCENADVUKRILEN.Tag = false;

textBoxBRTRIKRILNY.Tag = false;

textBoxCENATRIKRILEN.Tag = false;

textBoxBRPROZORECVRATA.Tag = false;

textBoxCENAPROZORECVRATA.Tag = false;

}

private void ValidateOK()

{

//активира бутона Изчисли ако стойностите на всички свойства Tag са true

this.buttonIzchisly.Enabled = ((bool)(this.textBoxNAME.Tag) &&

(bool)(this.textBoxNOMER.Tag) &&

(bool)(this.textBoxBRDVUKRILNY.Tag) &&

(bool)(this.textBoxCENADVUKRILEN.Tag) &&

(bool)(this.textBoxBRTRIKRILNY.Tag) &&

(bool)(this.textBoxCENATRIKRILEN.Tag) &&

(bool)(this.textBoxBRPROZORECVRATA.Tag) &&

(bool)(this.textBoxCENAPROZORECVRATA.Tag));

}

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

{

string output; // Деклариране на променлива от тип string в която да се заредят данните за клиента.

output = "Номер на клиента: " + this.textBoxNAME.Text + "\r\n" +

"Име на клиента " + this.textBoxNOMER.Text + "\r\n";//конкатенация на текстовите

//стойности от полетата за име и номер

this.textBoxDANNY.Text = output;//поставяне на текста зареден в променливата output в полето textBoxDANNY

double smetka = 0;// Деклариране на променлива от тип double в която да се зареди резултата от аритметичните операции

double DVUKRILEN, TRIKRILEN, PROZORECVRATA;// Деклариране на променливи от тип double

//в следващите три реда се вземат данните от TextBox полетата за брой и цена преобразуват се съответно в тип int и double и се умножават.

DVUKRILEN = int.Parse(textBoxBRDVUKRILNY.Text) \* double.Parse(textBoxCENADVUKRILEN.Text);

TRIKRILEN = int.Parse(textBoxBRTRIKRILNY.Text) \* double.Parse(textBoxCENATRIKRILEN.Text);

PROZORECVRATA = int.Parse(textBoxBRPROZORECVRATA.Text) \* double.Parse(textBoxCENAPROZORECVRATA.Text);

smetka = DVUKRILEN + TRIKRILEN + PROZORECVRATA;//сметката преди отстъпката

//следващите три реда проверяват кой радиобутон е избран и от сметката се изважда съответната сума

if (radioButtonDO10PROZORECA.Checked) smetka -= smetka \* 0.05;

if (radioButtonDO15PROZORECA.Checked) smetka -= smetka \* 0.1;

if (radioButtonNAD15PROZORECA.Checked) smetka -= smetka \* 0.15;

textBoxOBSHTO.Text = String.Format("{0:f2}", smetka);//изписване на стойноста заредена в променливата smetka в полето textBoxOBSHTO

}

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

{

TextBox txt = sender as TextBox;

foreach (Control ctrl in panel1.Controls)

{//този метод обхожда всички TextBox контроли във входната част на формата

//и изчиства записите в тях и възстановява цвета ако е променен в червен

txt = ctrl as TextBox;

if (txt != null)

{

txt.Text = "";

txt.BackColor = SystemColors.Window;

txt.Tag = false;

}

}

//изтриване на данните от textBoxDANNY и textBoxOBSHTO

textBoxDANNY.Clear();

textBoxOBSHTO.Clear();

}

private void textBoxNAME\_KeyPress(object sender, KeyPressEventArgs e)

{//метод за валидиране чрез който се забранява въвеждането на цифри в textBoxNAME

if (!(e.KeyChar < 47 || e.KeyChar > 58) && e.KeyChar != 8) e.Handled = true;

}

private void textBoxNOMER\_KeyPress(object sender, KeyPressEventArgs e)

{//метод за валидиране чрез който се забранява въвеждането на други символи освен цифри и запетая в textBoxNOMER

if ((e.KeyChar < 48 || e.KeyChar > 58) && e.KeyChar != 8 && e.KeyChar != 44) e.Handled = true;

}

private void textBoxNAME\_Validating(object sender, CancelEventArgs e)

{

//изпращач е обекта TextBox по тази причина същия се привежда към този тип

TextBox tb = (TextBox)sender;

//ако липсва текст се установява червен цвят нафона на контрола TextBox

// за указване на допустимост на данните се използва свойството Tag

if (tb.Text.Length == 0)

{

tb.BackColor = Color.Red;

tb.Tag = false;

//в този случай се обработва по-нататъшно обработване

}

else

{

tb.BackColor = System.Drawing.SystemColors.Window;

tb.Tag = true;

}

//извиква се функцията ValidateOK(),която задава стойноста на свойството Enabled на buttonIzchisly

ValidateOK();

}

}

}