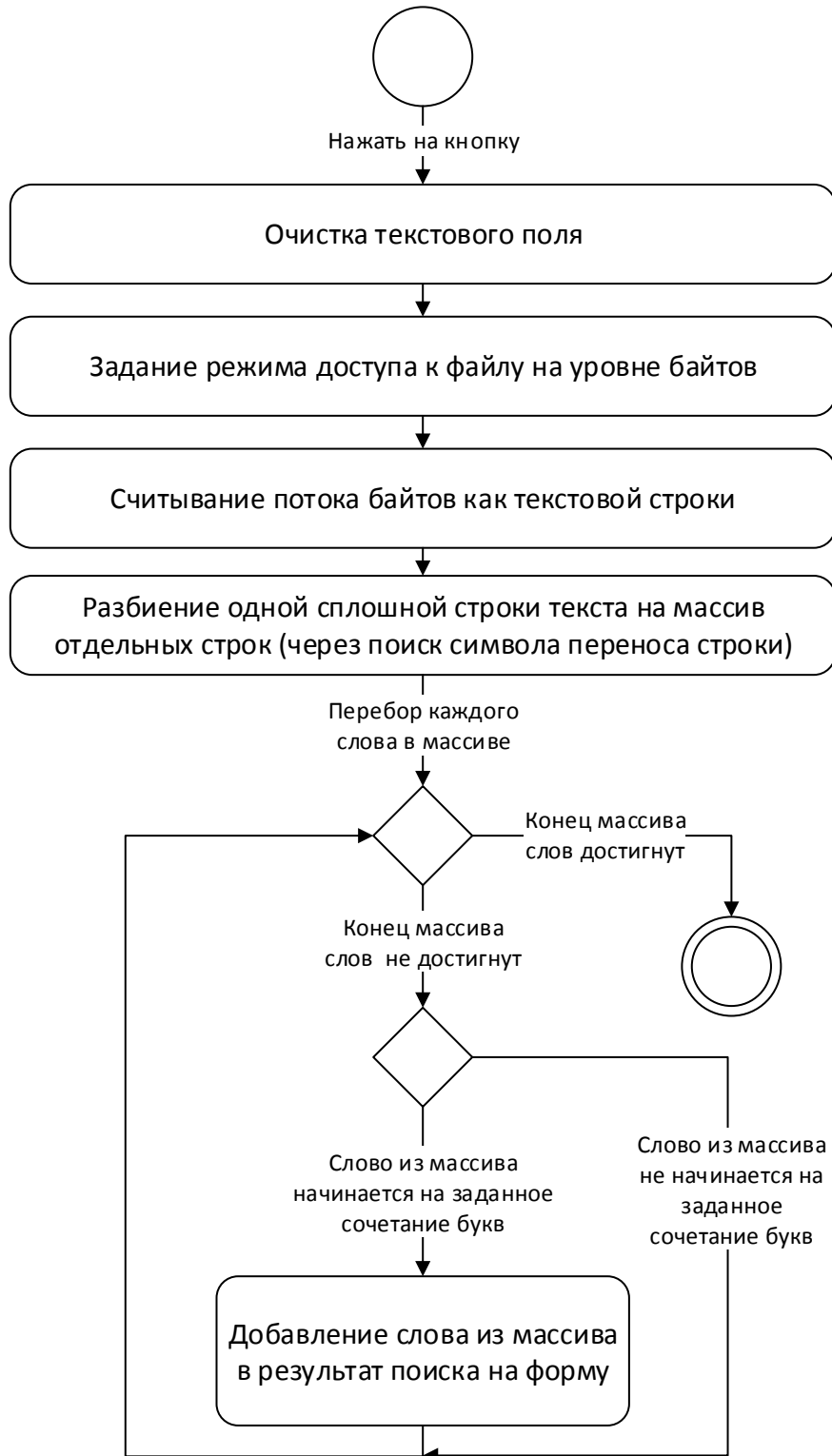


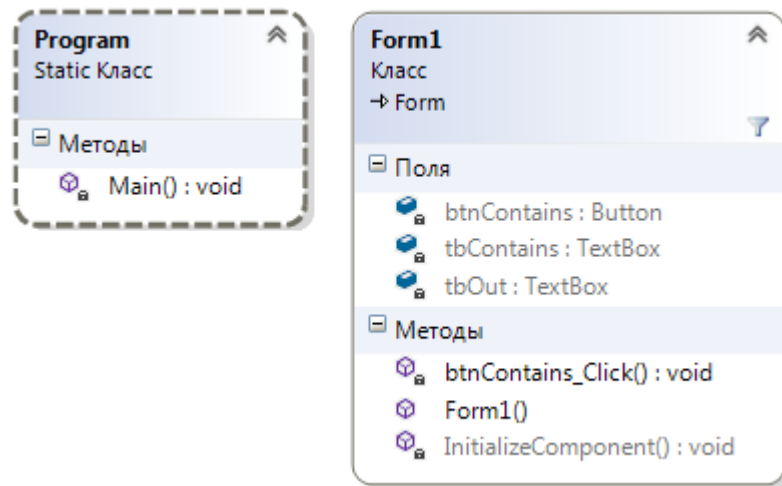
# Лабораторная работа № 2

## «Поиск слов в файле по первым буквам»

### UML-диаграмма действий



## Диаграмма классов



1. Создаём проект: **Файл – Создать – Проект.**

В открывшемся окне:

Шаблоны – **Visual C#**

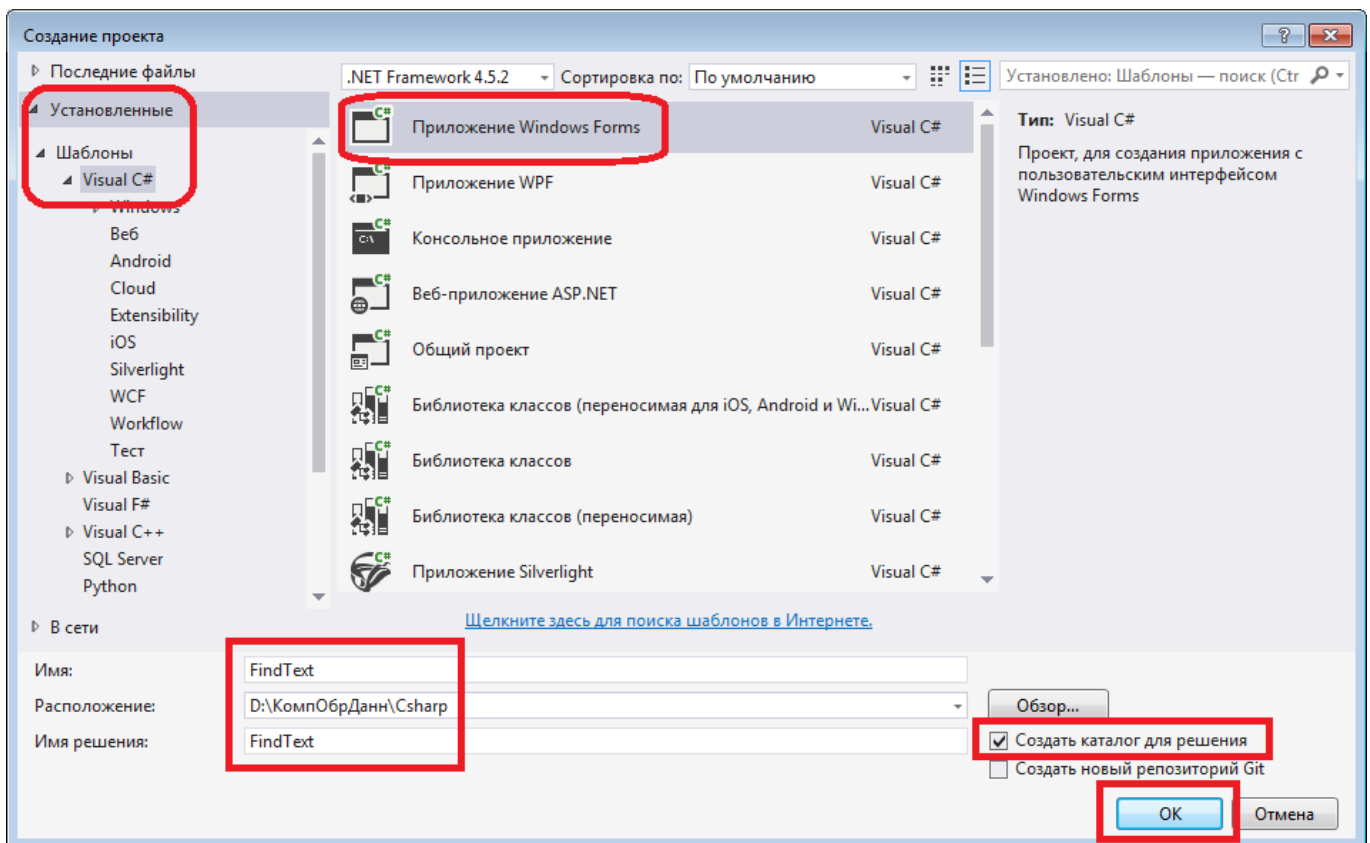
Шаблон – **Приложение Windows Forms**

Имя – **FindText**

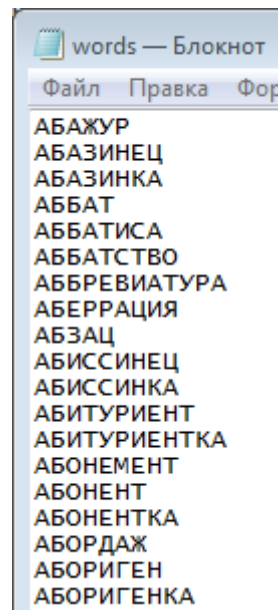
Расположение - **d:\КомпОбрДанн\Csharp**

Создать каталог для решения – поставить галочку

Нажать **ОК.**

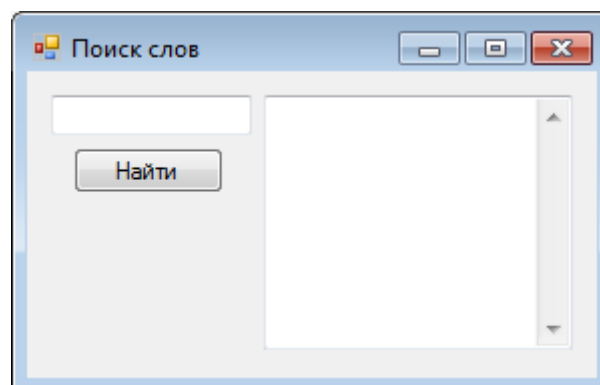


2. Перейти в папку **d:\КомпОбрДанн\Csharp\** и скопировать в неё готовый текстовый файл **words.txt** с набором слов, или создать новый и записать в него 15 – 20 слов. Все слова в этом файле должны быть в верхнем регистре, каждое слово – с новой строки!



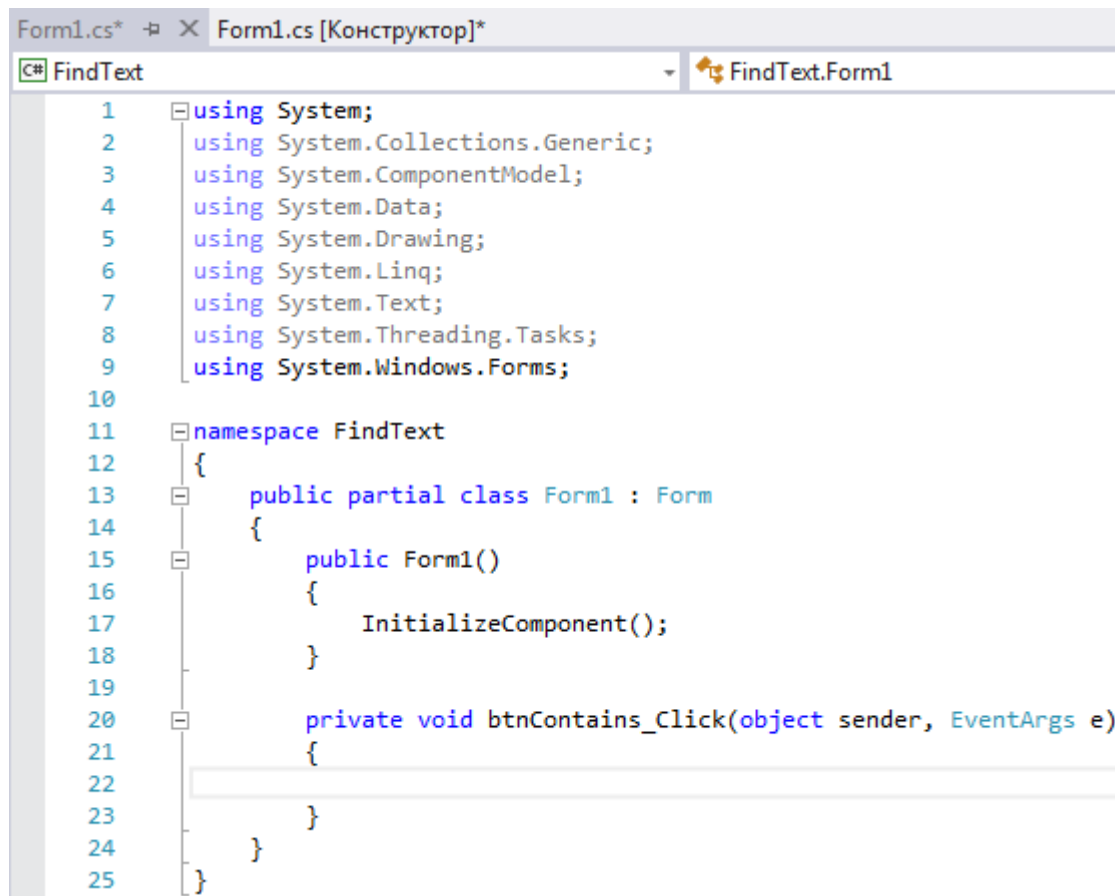
АБАЖУР  
АББАТ  
АББРЕВИАТУРА  
АБЕРРАЦИЯ  
АБЗАЦ  
БИАТЛОН  
БИБЛИОГРАФ  
ЗАПРЕТ  
ЗАПРОС  
ПАРКОВКА  
ПАРЛАМЕНТАРИЙ  
ПАРНОКОПЫТНЫЕ  
ПАРОВОЗ

3. Открыть **Панель элементов** (через меню **Вид – Панель элементов**). Поместить на форму 2 компонента **TextBox** и один компонент **Button**.



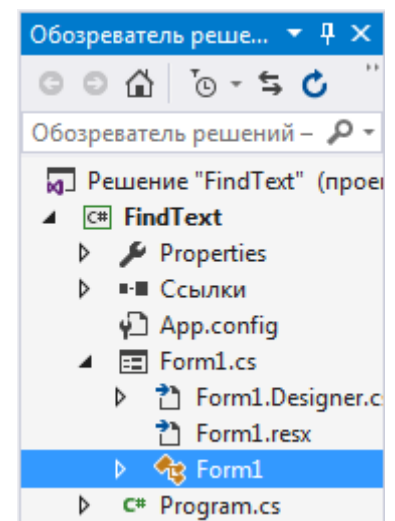
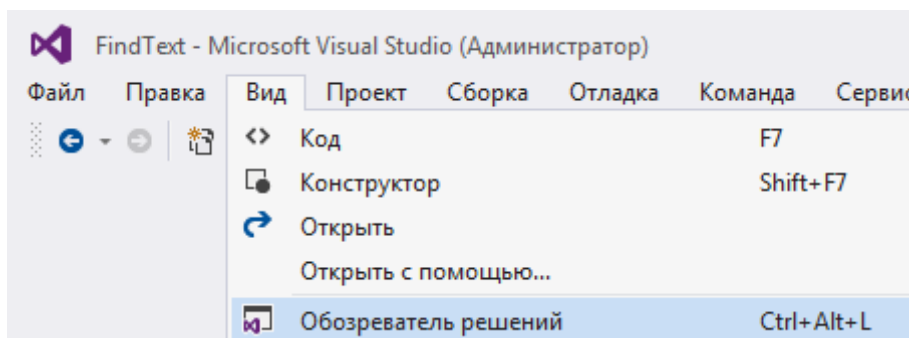
Правый клик по кнопке – **Свойства (Properties)** – переходим в окно свойств. Находим свойство **(Name)**, и меняем значение **button1** на **btnContains**. Аналогично меняем у **textBox1** свойство **(Name)** с **textBox1** на **tbContains**. Меняем у **textBox2** свойство **(Name)** с **textBox2** на **tbOut**, свойство **ScrollBars** с **None** на **Vertical**.

4. Двойной клик по кнопке **btnContains** автоматически генерирует метод **btnContains\_Click()**, который обрабатывает нажатие по кнопке **btnContains**.  
Открывается окно **Form1.cs** с автоматически сгенерированным кодом.



```
Form1.cs*  Form1.cs [Конструктор]*
C# FindText FindText.Form1
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace FindText
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void btnContains_Click(object sender, EventArgs e)
21         {
22
23         }
24     }
25 }
```

5. В строке меню нажать **Вид – Обзорщик решений**.  
Автоматически сгенерированный код находится на ветке **FindText – Form1.cs – Form1**.



6. В окне **Form1.cs** в разделе **using** добавить пространство имен **System.IO**.  
Пространство имен **System.IO** содержит типы, позволяющие осуществлять чтение и запись в файлы и потоки данных, а также типы для базовой поддержки файлов и папок.

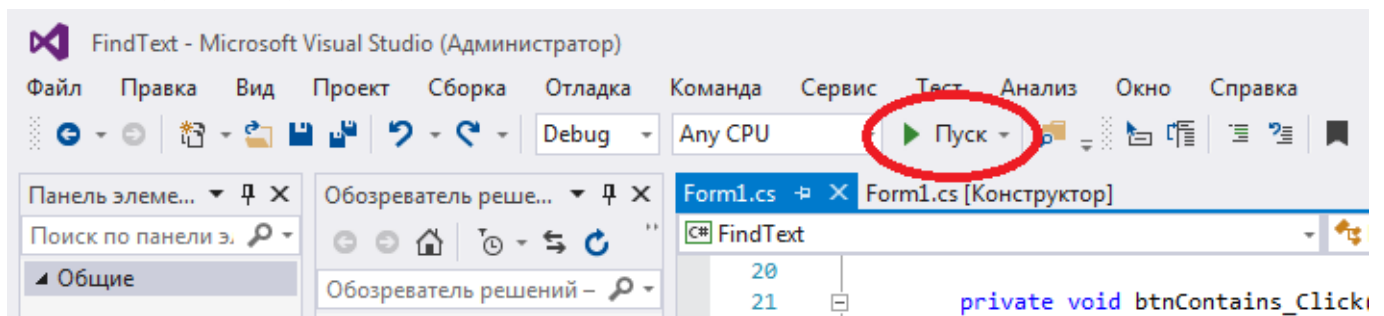
```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.IO;
11
12 namespace FindText
13 {
14     public partial class Form1 : Form
15     {
16         public Form1()
17         {
18             InitializeComponent();
19         }
20     }
21 }
```

7. Заполнить метод **btnContains\_Click()** следующим кодом:

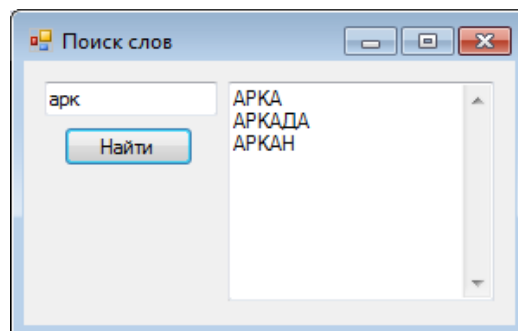
```
private void btnContains_Click(object sender, EventArgs e)
{
    // очищаем текст в компоненте tbOut
    tbOut.Text = String.Empty;
    // задаём поток байтов stream для открытия и способа обращения к файлу
    Stream stream = new FileStream(@"d:\КомпОбрДанн\Csharp\words.txt",
        FileMode.Open, FileAccess.Read);
    // задаём переменную reader для считывания байтов из потока байтов stream
    StreamReader reader = new StreamReader(stream);
    // задаём переменную words для списка строк, с доступом к нему по индексу
    List<string> words = new List<string>();
    // считываем в text все символы с текущей позиции до конца потока stream
    string text = reader.ReadToEnd();

    // Split('\n') - метод возвращает строковый массив, с элементами,
    // разделёнными знаком переноса строки
    // AddRange - метод добавляет элементы коллекции в конец списка
    // (формирует из потока байтов слова, разделённые знаком переноса строки)
    words.AddRange(text.Split('\n'));
    // для каждого элемента из списка (для каждого слова в файле)
    foreach (string word in words)
    {
        // если слово из файла начинается на сочетание букв, записанных
        // в tbContains, то добавляем с новой строки это слово в tbOut
        if (word.StartsWith(tbContains.Text.ToUpper()))
        {
            tbOut.Text += word + '\n';
        }
    }
    // закрываем основной поток и освобождаем ресурсы, связанные с устройством чтения
    reader.Close();
    // закрывает текущий поток и отключает ресурсы, связанные с текущим потоком
    stream.Close();
}
```

## 8. Запустить программу



## 9. Результат выполнения



## 10. Как понятно из названия метода **StartsWith**

```
if (word.StartsWith(tbContains.Text.ToUpper()))
```

данный метод ищет только слова, которые начинаются на заданную букву или сочетание букв, и не ищет слова, которые содержат поисковый запрос в произвольном месте внутри слова.

Также путь к файлу со словами («базе данных») жёстко задан в коде, и нет возможности выбирать файл с произвольным расположением на диске.

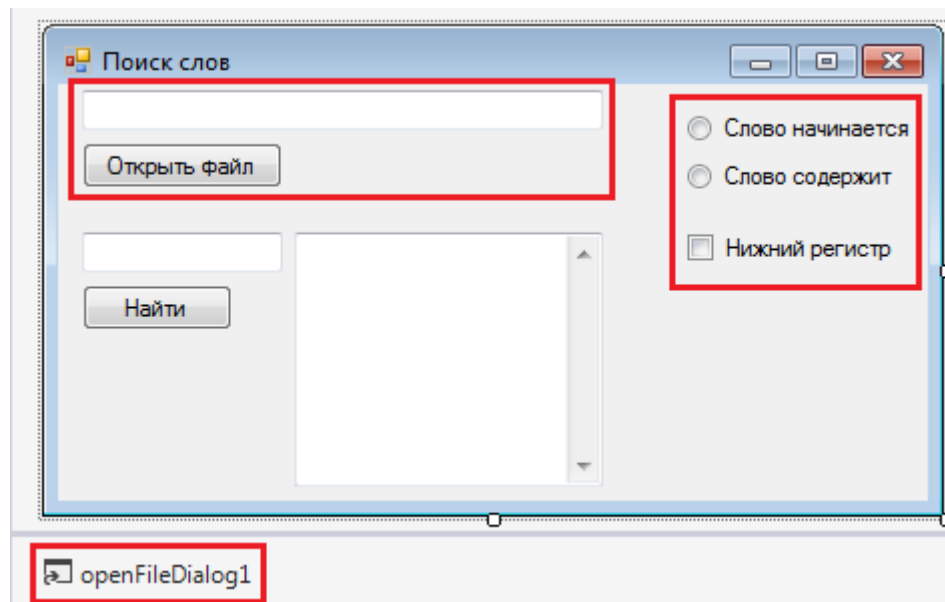
Дополним программу отсутствующим функционалом.

## Улучшение программы

## 11. Открыть **Панель элементов** (через меню **Вид – Панель элементов**).

Дополнительно к существующим элементам поместить на форму один компонент **TextBox**, один компонент **Button**, два компонента **RadioButton**, один компонент **CheckBox**, один компонент **OpenFileDialog**.

Примечание: компонент **OpenFileDialog** необходимо поместить непосредственно на саму форму, а Visual Studio автоматически разместит его ниже окна формы, т.к. он является «невидимым» компонентом при работе программы. Т.е. хотя компонент **OpenFileDialog** визуальное и располагается ниже формы, но фактически всё равно расположен на форме; поэтому для работы с ним, его визуализация отделена от визуализации самой формы для исключения ошибок наложения элементов друг на друга.



Правый клик по новой кнопке – **Свойства (Properties)** – переходим в окно свойств на вкладку **Свойства**. Меняем:

- у компонента **Button** свойство **(Name)** с **button1** на **btnOpenFile**, свойство **Text** с **button1** на **Открыть файл**;
- у компонента **TextBox** свойство **(Name)** с **textBox1** на **tbOpenFile**;
- у компонента **radioButton1** свойство **(Name)** с **radioButton1** на **rbStart**, свойство **Text** с **radioButton1** на **Слово начинается**;
- у компонента **radioButton2** свойство **(Name)** с **radioButton2** на **rbContains**, свойство **Text** с **radioButton2** на **Слово содержит**;
- у компонента **CheckBox** свойство **Text** с **checkBox1** на **Нижний регистр**;

12. Добавим условие, чтобы при запуске программы кнопка **btnContains** была недоступна для нажатия. Это можно сделать двумя способами:

- или через код;
- или через окно свойств (свойство **Enabled** задать как **False**).

В обоих случаях необходимо событие, которое в дальнейшем сделает кнопку доступной для нажатия. В данном случае таким событием будет выбор одной из радиокнопок **rbStart** (**Слово начинается**) или **rbContains** (**Слово содержит**). Сделаем кнопку **btnContains** недоступной через код.

В конструкторе **Form1()** класса **Form1** дописать одну строку:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        // сразу после создания формы сделать кнопку поиска недоступной
        btnContains.Enabled = false;
    }
}
```

### 13.Изменить цикл **foreach** в классе **Form1**:

```
// для каждого элемента из списка (для каждого слова в файле)
foreach (string word in words)
{
    // если нажата радиокнопка rbStart
    if (rbStart.Checked == true)
    {
        // перевод букв из окна поиска tbContains в верхний регистр
        // если слово в текстовом файле начинается с букв из окна поиска
        if (word.StartsWith(tbContains.Text.ToUpper()))
        {
            // если галочка на чекбоксе активна
            if (checkBox1.Checked == true)
            {
                // выводим слова в нижнем регистре
                tbOut.Text += word.ToLower() + '\n';
            }
            // если галочки на чекбоксе нет
            else
            {
                // выводим слова в верхнем регистре
                tbOut.Text += word + '\n';
            }
        }
    }
    // если нажата радиокнопка rbContains
    if (rbContains.Checked == true)
    {
        // перевод букв из окна поиска tbContains в верхний регистр
        // если слово в текстовом файле содержит буквы из окна поиска
        if (word.Contains(tbContains.Text.ToUpper()))
        {
            // если галочка на чекбоксе активна
            if (checkBox1.Checked == true)
            {
                // выводим слова в нижнем регистре
                tbOut.Text += word.ToLower() + '\n';
            }
            // если галочки на чекбоксе нет
            else
            {
                // выводим слова в верхнем регистре
                tbOut.Text += word + '\n';
            }
        }
    }
}
```

### 14.Перейти на окно с формой, дважды кликнуть по кнопке **btnOpenFile** (Открыть файл). В окне кода автоматически создастся метод **btnOpenFile\_Click**. В этом методе дописать код:

```
private void btnOpenFile_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK) //показываем диалог открытия
        tbOpenFile.Text = openFileDialog1.FileName; //в текстбокс записываем путь к файлу
}
```

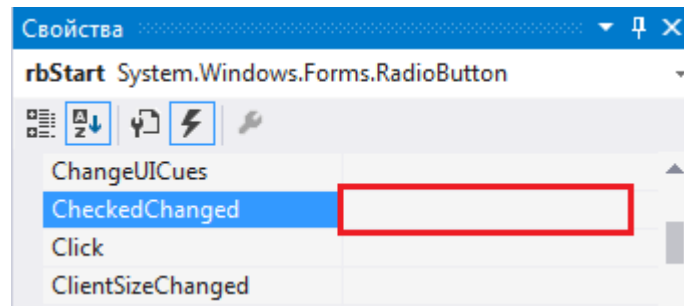


15.Перейти на окно с формой, выделить радиокнопку **rbStart**, в окне свойств

перейти на вкладку **События**    .

Найти событие **CheckedChanged** и дважды кликнуть мышкой по полю справа.

В окне кода автоматически сгенерируется метод **rbStart\_CheckedChanged**.



16.Сделать аналогичную операцию для радиокнопки **rbContains** и метода **rbContains\_CheckedChanged**.

17.Дописать код в тело обоих методов:

```
private void rbStart_CheckedChanged(object sender, EventArgs e)
{
    // если есть галочка на радиокнопке rbStart
    if (rbStart.Checked == true)
    {
        // сделать кнопку поиска btnContains доступной для нажатия
        btnContains.Enabled = true;
    }
}

private void rbContains_CheckedChanged(object sender, EventArgs e)
{
    // если есть галочка на радиокнопке rbContains
    if (rbContains.Checked == true)
    {
        // сделать кнопку поиска btnContains доступной для нажатия
        btnContains.Enabled = true;
    }
}
```

18.Запустить программу на выполнение:

