



Střední škola pedagogická, hotelnictví a služeb, Litoměřice, příspěvková organizace

Předmět:	Vývoj aplikací
Téma:	Visual Studio
Vyučující:	Ing. Milan Káža
Třída:	EK3
Hodina:	25-27
Číslo:	V/5

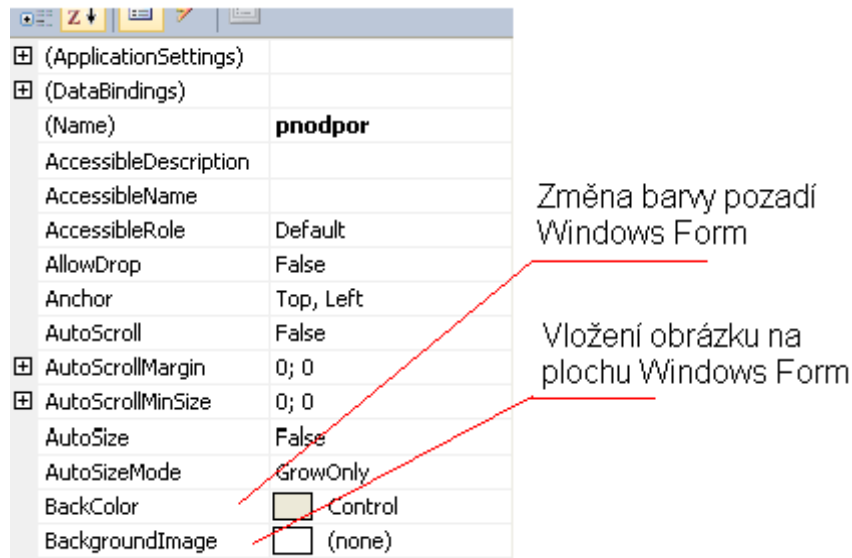
Programování v jazyce C a C#

Visual Studio

8. Hraní s barvami a další možnosti Visual Studia

V této části budou objasněny některé efekty, které dodávají programu jeho „krásku“ a mnohé lidi zaujmou, některé jsou ale důležité, proto se s nimi nyní seznámíme.

8.1. Barva pozadí a obrázky na pozadí



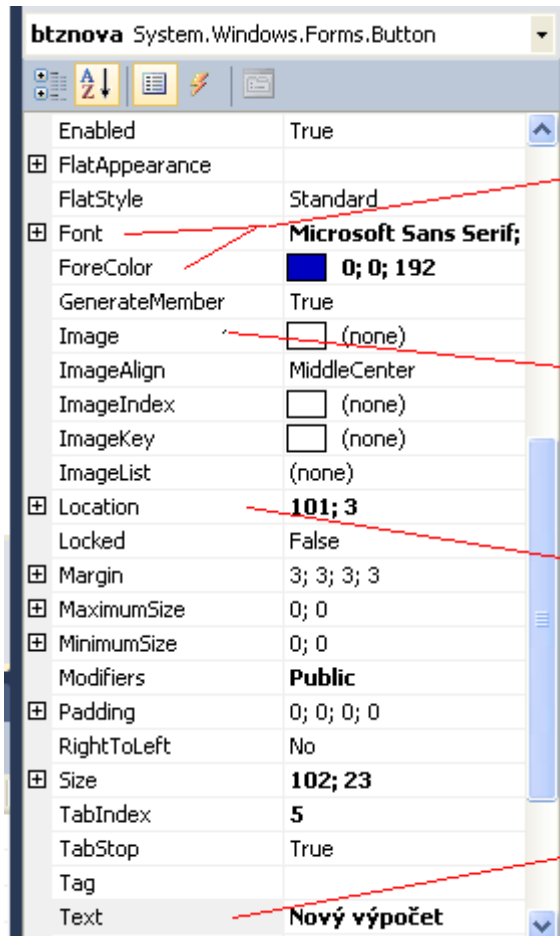
Záložka BackColor umožňuje změnit pozadí Windows Formy otevřením této složky se rozbalí nabídka barev a jejím stisknutím se změní pozadí.

Záložka BackgroundImage je pro vložení obrázku, který programátor si programátor vytvořil a vloží ho místo pozadí. Otevřením složky BackgroundImage se otevře okno s nabídkou, kde je možné si vybrat obrázek a potvrzením tlačítkem OK vložit na pozadí.

Otevřete Visual Studio a připravte si obrázek, který vložíte na pozadí. Otevřete si záložku BackgroundImage a vložte obrázek. Záložka je zobrazena na obrázku.

8.2. Velikost a barva písma

Stejně jako je možné měnit písmo v produktech Microsoft Office je to možné dělat i ve Visual Studiu. Při práci ve



Font pro písmo a jeho barvu

Vložení obrázku

Souřadnice X,Y

Popis ovládacích prvků

Windows Formu je otevřená záložka Properties, většinou je vpravo z pohledu programátora, kde je možné provádět úpravy a nastavení písma, podkladu atd.

Písmo se nastavuje v záložce Font, zde je možné nastavit velikost, typ písma, barvu písma.

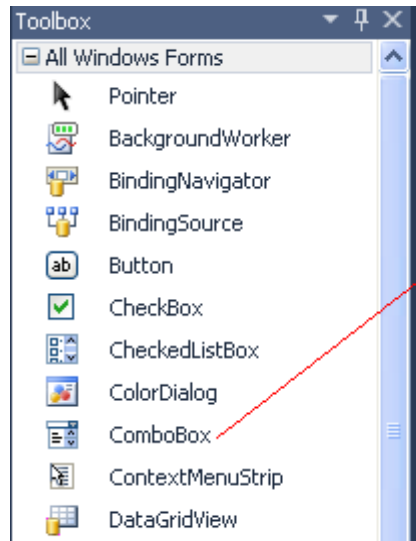
Do Windows Form vložte Label nebo Button nastavte myš na Font a otevřete vyberte si písmo, barvu a nastavte tak aby se vám nápis líbil.

Pokud je požadavek vložit do Buttonu obrázek, pak se použije záložka Image tím se otevře nabídka vybere se obrázek a ten se načte do daného Buttonu.

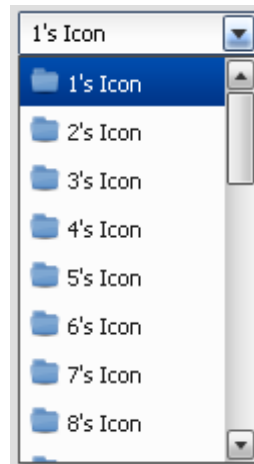
Pozor: Toto neplatí pro nahrání obrázku na celou plochu.

Je možné určit i kde se daný text zobrazí k tomu se používá záložka Location a popis je v záložce Text, ale s tím je většina již obeznámena.

8.3. ComboBox



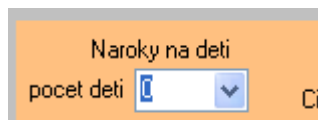
ComboBox je ovládací prvek pro výběr ze seznamu, kde je možné zobrazit seznam rozbalením a z něj provést výběr. Na obrázku je znázorněna záložka a v ní ComboBox.



Na tomto obrázku je ukázán otevřený ComboBox se seznamem, pomocí šipky se otevře seznam a z něj se může vybrat vložený údaj.

ComboBox se využívá při otvírání nabídky půjčky, nebo při zadání počtu dětí, při výběru města apod. Zadání ComboBoxu do programového kódu.

Vaším úkolem je vložit do programového kódu ComboBox s nabídkou počtu dětí, kterých je max.8.



*Do Windows Formy vložíte ComboBox, vybraný z nabídky a aktivujete ho, v programovém kódu se zobrazí: **private void cobpocet_SelectedIndexChanged(object sender, EventArgs e)** Nyní si vytvořte nové pole tak ja je ukázáno v proceduře Pocet_Deti().*

Zde je zadán počet dětí, který se pak objeví v nabídce pro zadání do počtu dětí při výpočtu daní.

```
private void Pocet_Deti ()
{
    this.cobpocet.Text = "0";
    this.cobpocet.Items.AddRange(new object[] { "0", "1", "2", "3", "4", "5", "6", "7", "8" });
}
```

Odkaz **this** provádí volání na aktuální instanci a je možné ho použít ke kvalifikaci datových složek, jako je využito v příkladu.

*Nyní vytvořte ovládání ComboBoxu, napište příkaz `if(cobpocet.Text=="1")`, který provádí porovnání kolik dětí má dotyčná osoba, výsledek je vložen do celkového výpočtu. Počet dětí je zadán v poli a deklarován jako **new object**. Jak je vidět ze zadání aplikace ComboBoxu dochází k porovnání mezi zadanou položkou a položkou vloženou v poli.*

```
private void cobpocet_SelectedIndexChanged(object sender, EventArgs e)
{
    double nadite;
    nadite = odecet_deti;

    if (cobpocet.Text == "1")
    {
        odecet_deti = nadite;
        lbdeti.Text = "pocet deti:1, narok na odecet " + "" + odecet_deti;
    }

    else if (cobpocet.Text == "2")
    {
        odecet_deti = nadite * 2;
        lbdeti.Text = "pocet deti:2, narok na odecet "+ "" + odecet_deti;
    }
}
```

```
}  
  
else if (cobpocet.Text == "3")  
{  
    odecet_deti = nadite * 3;  
    lbdeti.Text = "pocet deti:3, narok na odecet " + "" + odecet_deti;  
}  
  
else if (cobpocet.Text == "4")  
{  
    odecet_deti = nadite * 4;  
    lbdeti.Text = "pocet deti:4, narok na odecet " + "" + odecet_deti;  
}  
  
    else if(cobpocet.Text=="5")  
    {  
        odecet_deti=nadite*5;  
        lbdeti.Text = "pocet deti:5, narok na odecet " + "" + odecet_deti;  
    }  
  
else if (cobpocet.Text == "6")  
{  
    odecet_deti = nadite * 6;  
    lbdeti.Text = "pocet deti:6, narok na odecet " + "" + odecet_deti;  
}  
else if (cobpocet.Text == "7")  
{  
    odecet_deti = nadite * 7;  
    lbdeti.Text = "pocet deti:7, narok na odecet " + "" + odecet_deti;
```

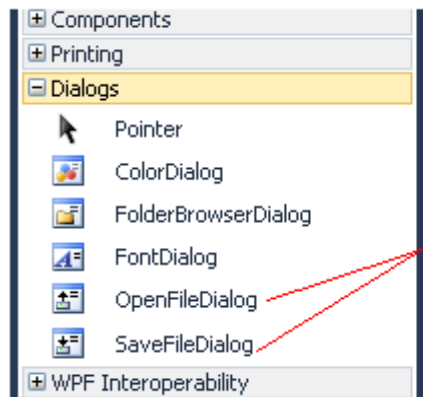
```
    }  
    else if (cobpocet.Text=="8")  
    {  
        odecet_deti = nadite * 8;  
        lbdeti.Text = "pocet deti:7, narok na odecet " + "" + odecet_deti;  
    }  
else  
    {  
        odecet_deti = nadite * 0;  
        lbdeti.Text ="bezdetne" + "0";  
    }  
    lbpocdeti.Text = "" + lbdeti.Text;  
}
```

Název třídy `Pocet_Deti()` vložte do `public Form1()` odtud je zavolaná třída a současně se zobrazí základní nabídka **ComboBoxu** v tomto případě 0. Celý zápis pak bud vypydat takto

```
public Form1()  
{  
    InitializeComponent();  
    Pocet_Deti();  
}
```

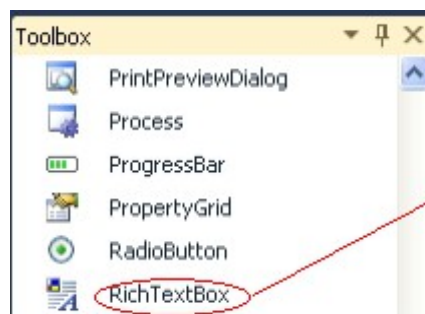
Při spuštění programu se objeví základní nabídka a stisknutím šipky se rozbalí nabídka. Zkuste si sami provést nabídku pomocí **Combo Boxu**.

8.4. Otevření souboru a jeho uložení



Otevření a uložení souboru

Na tomto obrázku jsou znázorněné komponenty, s jejichž pomocí se otevírá a ukládá soubor. Pomocí této komponenty se uživateli otevře nabídka (okno), a umožní uživateli vybrat soubor, který si chce otevřít. Komponenty **OpenFileDialog** a **SaveFileDialog** se nezobrazují v hlavní nabídce **Windows Form**, ale pod ní a volají se až při otevírání a ukládání souborů.

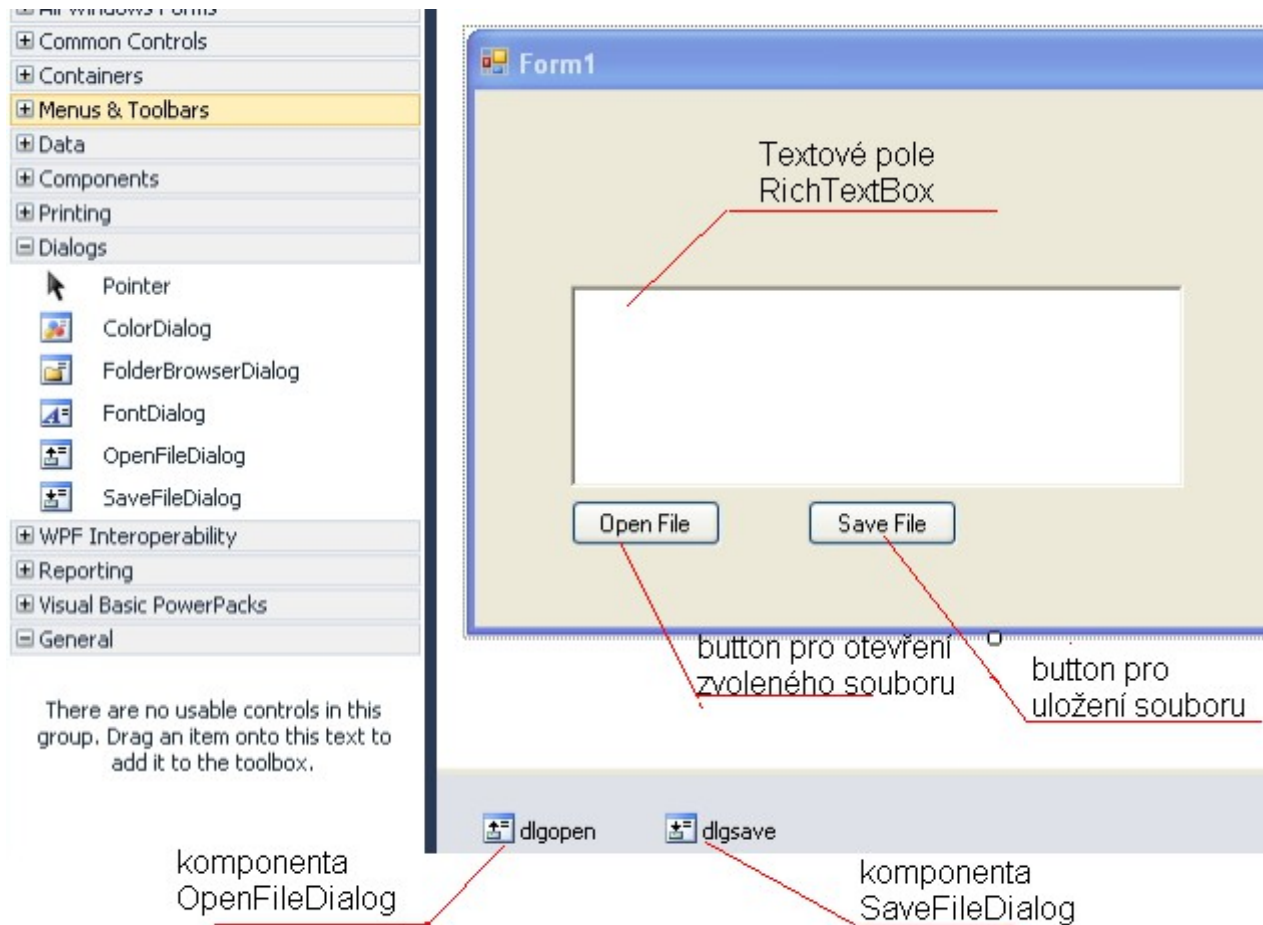


záložka RichTextBox

Aby bylo možné zobrazit obsah souboru je nutné vložit do **Windows Formu** záložku **RichTextBox**, do kterého se zobrazí obsah zvoleného souboru. Záložka je v nabídce komponent jak je zobrazena na obrázku. Záložka **RichTextBox** není jediná pro zobrazení obsahu souboru, další možnost je **ListBox** nebo **TextBox**, **RichTextBox** má tu výhodu, že je možné zavést příkaz pro vyhledávání **FindDialog**, v **ListBoxu** se tento příkaz vložit nedá.

8.4.1. Program otevření a uložení souboru

Založte ve Visual Studio nový projekt. Do WindowsFormu vložte komponenty RichTextBox, a dva Buttony, dále vložte komponenty OpenFileDialog a SaveFileDialog a všechny komponenty pojmenujte. Výsledek by měl vypadat tak je vidíte na obrázku.



Na obrázku jsou znázorněny všechny komponenty pro práci se soubory. Zatím nic neaktivujte, zde je pouze příprava pro vytvoření programového kódu. Nyní budete vytvářet programový kód.

Otevřete si prostředí pro vytvoření kódu a vložte knihovnu pro práci se soubory `using System.IO`; bez této knihovny

nebude podpora pro práci se soubory.

Nyní aktivujte komponentu `OpenFileDialog` a v prostředí programového kódu se vytvoří tato záložka:

```
private void dlgopen_FileOk(object sender, CancelEventArgs e) //komponeta pro
//otevření souboru
{
```

do záložky vložte tyto příkazy a proměnné.

Proměnná `name_file` deklarovaná jako `string` načte jméno vybraného souboru z nabídky.

```
string name_file = dlgopen.FileName;
```

`FileInfo` poskytuje vlastnosti a metody pro vytváření, kopírování a zobrazení souboru udaj je proměnná. Zde zajistí vložení souboru.

```
FileInfo udaj = new FileInfo(name_file);
```

`TextReader` představuje reader umožňující čtení sekvenční posloupnosti znaků. `StreamReader` zajišťuje čtení znaků z proudu bajtů v určitém kódování a vložte do příkazu `using`.

Vložte proměnnou `radka` a deklaruje ji jako `string` dále vložte příkaz `while` a vložte do závorek uvedené příkazy. Ty zajistí čtení souboru dokud se nebude `null` a vše se zobrazí v `RichTextBoxu`.

```
using (TextReader read = new StreamReader(name_file))
{
    string radka;
    while ((radka = read.ReadLine()) != null)
    {
        rtxzobraz.Text+=radka+ "\n";
    }
}
```

Nyní je vytvořená aplikace pro otevření souboru aktivujte `Button`, který zajistí otevření souboru a do něj vložte tento příkaz:

```
dlgopen.ShowDialog();
```

Ten otevře okno nabídky souborů.

Nyní vytvořte aplikaci pro uložení souboru, jak vidíte na příkladu, pokud vynecháte příkazy `if (dlgsave.ShowDialog() != DialogResult.Cancel)` nebude se zobrazovat upozornění, že soubor již existuje a zda ho chcete přepsat. Soubor se ukládá z `RichTextBoxu`, proto je velmi důležité napsat poslední řádek.

```
private void Uloz_Soubor()  
{  
    string name_file = dlgopen.FileName;  
    FileInfo data = new FileInfo(name_file);  
    dlgsave.FileName = name_file;  
  
    if (dlgsave.ShowDialog() != DialogResult.Cancel)  
        name_file = dlgsave.FileName;  
    rtxzobraz.SaveFile(name_file, RichTextBoxStreamType.PlainText);  
  
}
```

Nakonec aktivujte druhý Button pro uložení souboru.

Prostudujte programový kód a vytvořte program pro otevření a uložení souboru.

Zdrojový kód programu

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

namespace PrjFilesy
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
//*****

        private void dlgopen_FileOk(object sender, CancelEventArgs e)
        {
            string name_file = dlgopen.FileName;
            FileInfo udaj = new FileInfo(name_file);

            using (TextReader read = new StreamReader(name_file))
            {
```

```
        string radka;
        while ((radka = read.ReadLine()) != null)
        {
            rtxzobraz.Text+=radka+ "\n";
        }
    }
}
//*****
private void Uloz_Soubor()
{
    string name_file = dlgopen.FileName;
    FileInfo data = new FileInfo(name_file);
    dlgsave.FileName = name_file;

    if (dlgsave.ShowDialog() != DialogResult.Cancel)
        name_file = dlgsave.FileName;
    rtxzobraz.SaveFile(name_file, RichTextBoxStreamType.PlainText);
}

//*****/

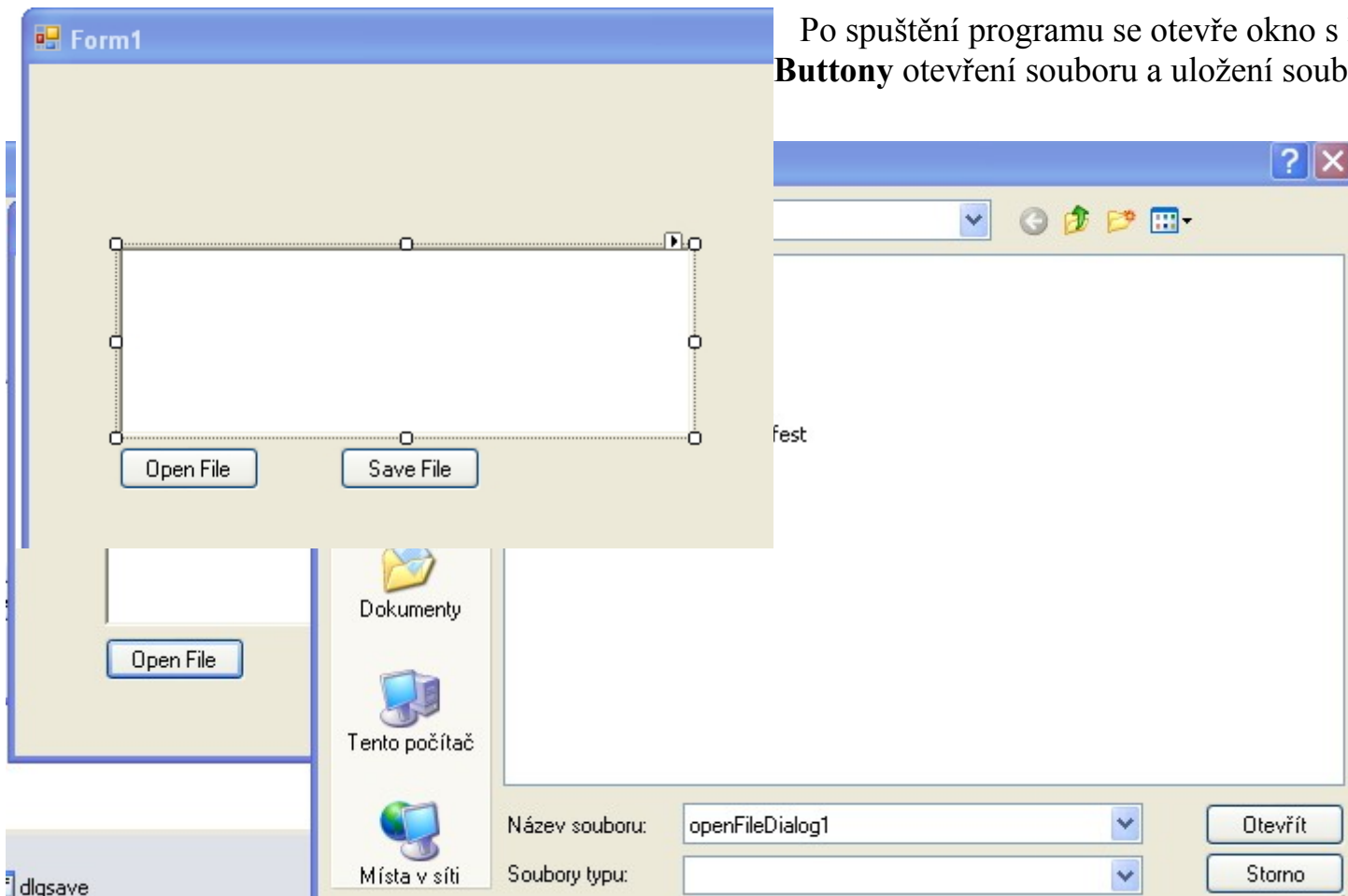
private void btopen_Click(object sender, EventArgs e)
{
    dlgopen.ShowDialog();
}
```

```
private void btnSave_Click(object sender, EventArgs e)
{
    Uloz_Soubor();
}

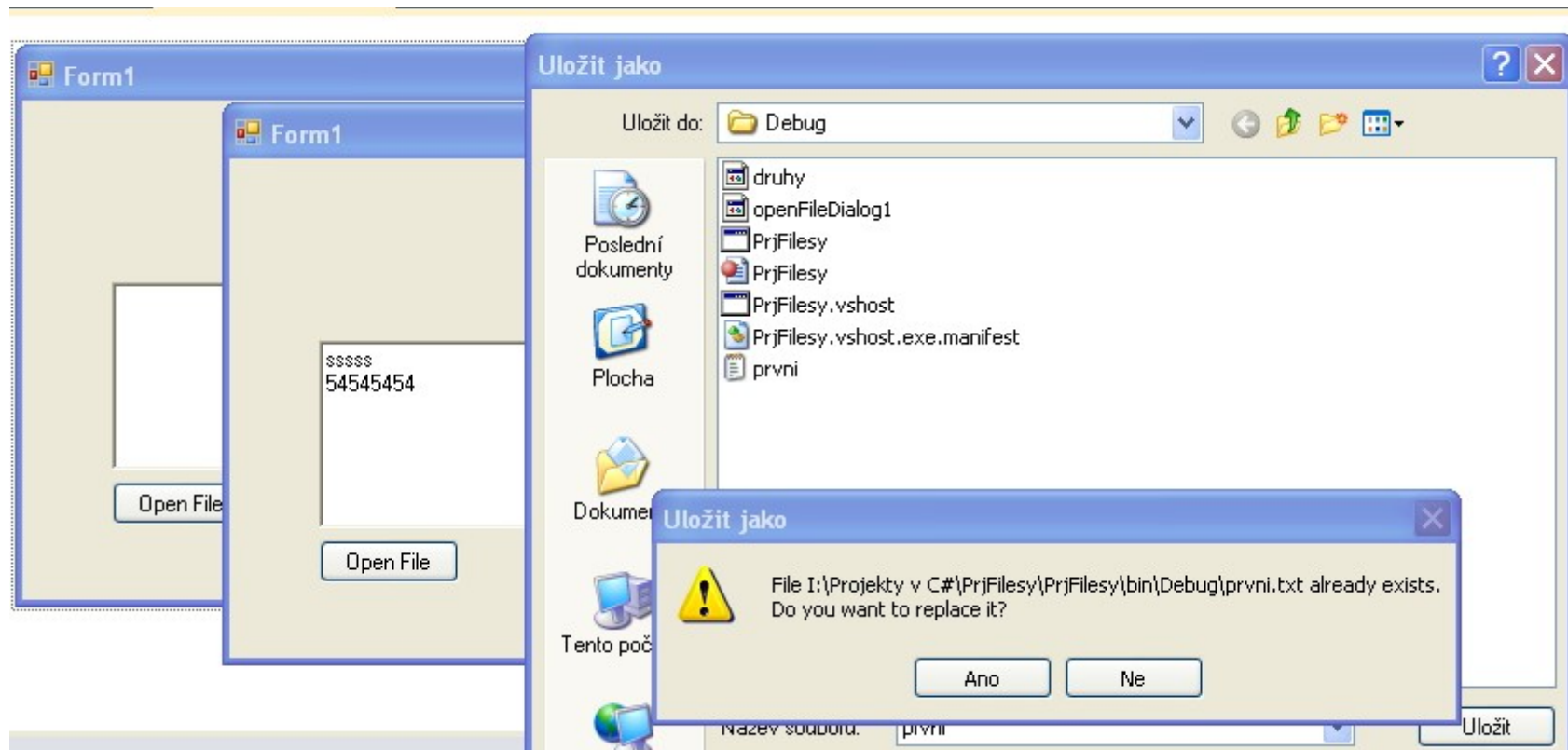
}

}
```

Po spuštění programu se otevře okno s **RichTextBoxem** a dvěma **Buttony** otevření souboru a uložení souboru, tak jak vidíte na obrázku.



Na tomto obrázku je nabídka souboru pro otevření. Uživatel může zvolit jaký soubor chce otevřít. Při nastavení programu je možné navolit filtr, pro přípony, které se mají otevírat, tak jak je známé z aplikací Windows.



Na tomto obrázku je ukázané upozornění, že soubor již existuje a zda ho chce uživatel přepsat, při zvolení Ano se soubor přepíše, pokud se zvolí Ne zadá uživatel nové jméno souboru.

Kontrolní otázky

1. Vysvětlíte ComboBox a jak ho využijete, napište program s ComboBoxem
2. Vytvořte program pro otevření a uložení souboru a vložte do něj filtr souboru pro přípony .txt a .dat

Použitá literatura:

- Miroslav Virius: od C++ k C#, 1.vydání, KOOP České Budějovice, 2002*
Eric Gunnerson: Začínáme programovat v C#, 1. vydání, Computer Press, 2001
John Sharp, Jon Jagger: Microsoft Visual C# .NET krok za krokem, 1. vydání, Knihy.iDnes
Miroslav Virius: C# Hotová řešení, 1. vydání, Computer Press, 2006
Amadeo Mareš: 1001 tipů a triků pro C#