

# PROGRAMOVÁNÍ

<b>2</b>	<b>SPŠ Chomutov</b>	<b>Beránek</b>
<b>11.11.2022</b>	<b>Textový editor</b>	<b>V4</b>

## Zadání:

Vytvořit textový editor pomocí objektově orientovaného programování

Musí umět:

Upravovat text

Ukládat a otevírat soubory

Zvýraznění kódu

## Teorie:

Objektově orientované programování se zaměřuje více na to, jak program vidí člověk spíše než stroj. Využívá práce s objekty, které mají vlastní atributy a metody. Protože je všechno uspořádané pod objekty je kód přehlednější, a hlavně čitelnější, na první pohled je mnohem jasnější co má program dělat. Objekty se vytváří ze tříd. Ty slouží jako jakési šablony pro vytváření objektů. Z jedné třídy můžeme vytvořit několik objektů, které budou sdílet stejné atributy a metody, ale budou mít jiná data.

## Popis programu:

Při jakémkoliv pokusu o vytvoření nového souboru, otevření jiného souboru nebo ukončení aplikace se otevře modální dialogové okno, které se zeptá uživatele jestli chce soubor uložit, neukládat nebo zrušit, podle rozhodnutí uživatele vrátí form string s příslušným textem rozhodnutí, na základě kterého hlavní program pokračuje příslušnou akcí

Základně proměnná „filepath“ obsahuje prázdné uvozovky, čímž program ví že uživatel doteď soubor neukládal. Ve chvíli, kdy se ho pokusí uložit, se objeví SaveFileDialog, kde si uživatel zvolí kam chce soubor uložit. Tuto cestu si program uloží do proměnné „filepath“ a při jakémkoliv dalším uložení automaticky ukládá do stejného souboru. „Uložit jako“ ovšem vždycky otevře SaveFileDialog a po vybrání souboru přepíše „filepath“ na aktuální cestu. Stejně tak se děje při otevření souboru. Při vytvoření nového souboru se „filepath“ přepíše zpět na prázdné uvozovky.

Program při uložení nebo načtení souboru kontroluje, jestli cesta obsahuje „.txt“ nebo „.rtf“ (jiné soubory program nepodporuje a nedovolí s nimi pracovat). Pokud se jedná o soubor typu txt, uloží do souboru čistě text pomocí StreamWriteru a nebere v potaz jakoukoliv jeho úpravu (což zajišťuje kompatibilitu s jinými programy které nepracují s úpravou textu). Pokud se jedná o soubor typu rtf, program ukládá text i s veškerou jeho úpravou pomocí SaveFile funkce RichTextBoxu. Podobně to funguje i při otevírání a načítání souborů.

Pokud si uživatel zapne zvýraznění syntaxe C# kódu, program prvně označí všechn text a změní jeho styl na základní a barvu na černou. Následně nastaví hodnotu proměnné „pozice“ na nulu a oddělí funkcí Split všechna slova. Jako oddělovače funkce používá mezeru, \n, \t a všechny druhy závorek. Nyní zkontroluje jestli se první slovo z pole všech těchto slov shoduje s nějakým ze slov v listu slov, které se mají upravovat. Pokud najde shodu, kurzor se přesune na hodnotu „pozice“, označí se počet znaků podle délky slova a označený text se přebarví. Poté program přičte k proměnné „pozice“ délku aktuálního slova (posune kurzor na konec slova). Nyní si označí jeden znak za aktuální pozicí a pokud se toto označení shoduje s jedním z oddělovačů použitých u funkce Split, přidá k hodnotě „pozice“ ještě +1. Nyní se proces opakuje pro další slovo v poli.

## Seznam funkcí a proměnných:

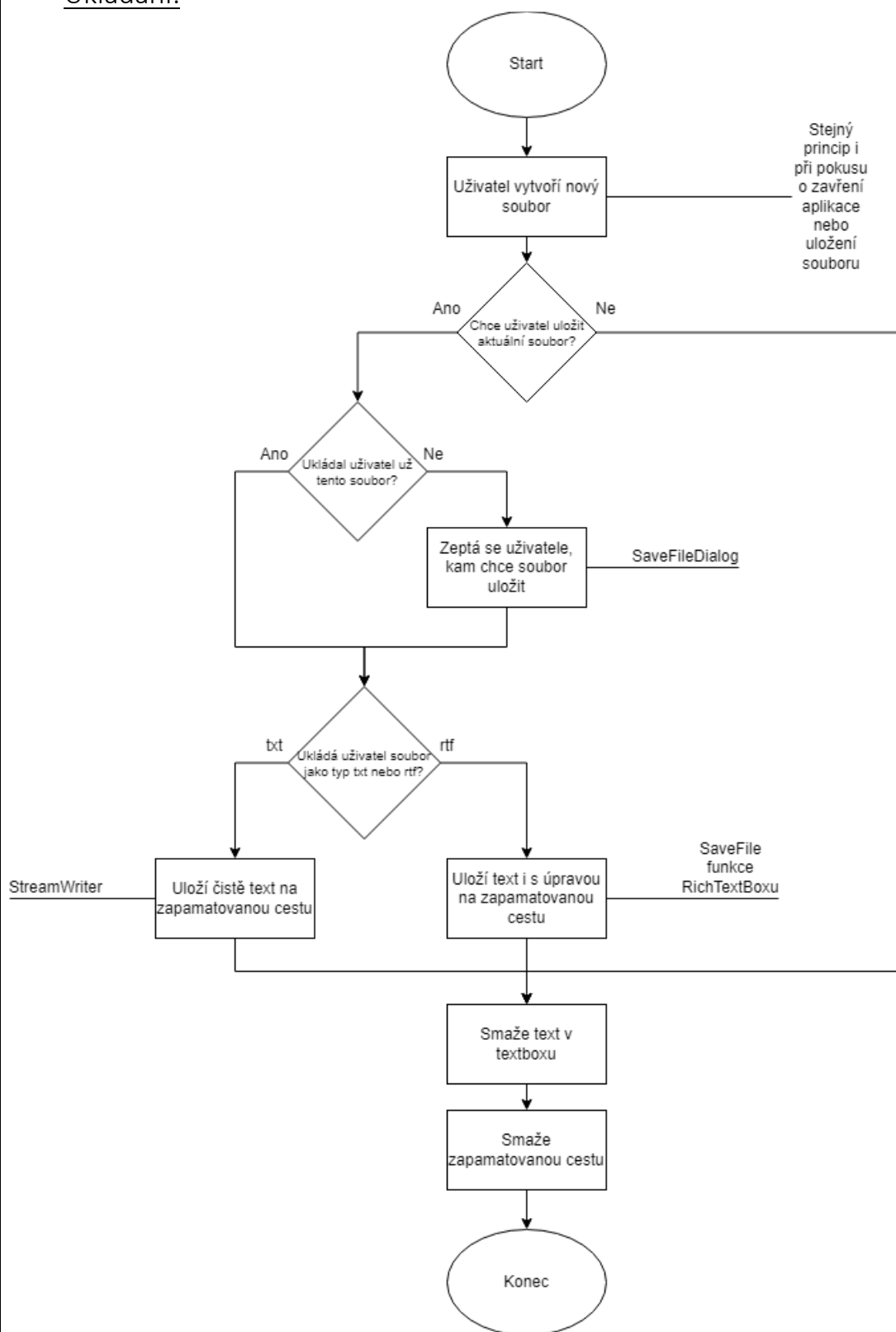
Typ	Proměnná	Účel
<b>Class Form1</b>		
string	filepath	Cesta k aktuálně upravovanému souboru
<b>Class Form_ulozit</b>		
string	result	Výsledek rozhodnutí uživatele
<b>Class Soubor</b>		
string	text	Obsahuje aktuálně načtený řádek textu
<b>Class Text</b>		
List<string>	fialova	Slova, která se mají přebarvit fialově
List<string>	modra	Slova, která se mají přebarvit modře
List<string>	znaky	Oddělovací znaky pro funkci Split
int	pozice	Aktuální pozice kurzoru

Typ	Argumenty	Metoda	Účel
<b>Class Form1</b>			
void	sender, e	soubor_otevrit_click	Umožní uživateli vybrat soubor a načte ho
void	sender, e	soubor_novy_click	Vytvoří nový soubor
void	sender, e	soubor_ulozit_click	Uloží aktuálně otevřený soubor
void	sender, e	soubor_ulozitjako_click	Uloží aktuálně otevřený soubor jako nový soubor
void	sender, e	form_zavren	Při pokusu o ukončení programu se uživatele zeptá jestli chce uložit soubor
void	sender, e	text_zarovnat_doleva	Zarovná text doleva
void	sender, e	text_zarovnat_nastred	Zarovná text na střed
void	sender, e	text_zarovnat_doprava	Zarovná text doprava
void	sender, e	text_barva_cerna	Změní barvu textu
void	sender, e	text_barva_bila	Změní barvu textu
void	sender, e	text_barva_modra	Změní barvu textu
void	sender, e	text_barva_zelena	Změní barvu textu
void	sender, e	text_barva_zluta	Změní barvu textu
void	sender, e	text_barva_cervena	Změní barvu textu
void	sender, e	text_barva_vlastni	Otevře ColorDialog a změní barvu textu
void	sender, e	text_font_calibri	Změní font textu
void	sender, e	text_font_comicans	Změní font textu
void	sender, e	text_font_papyrus	Změní font textu
void	sender, e	text_font_wingdings	Změní font textu
void	sender, e	font_text_vlastni	Otevře FontDialog a změní font textu
void	sender, e	text_velikost_8	Změní velikost textu
void	sender, e	text_velikost_12	Změní velikost textu
void	sender, e	text_velikost_16	Změní velikost textu
void	sender, e	text_velikost_20	Změní velikost textu
void	sender, e	text_velikost_72	Změní velikost textu
void	sender, e	text_velikost_vlastni	Změní velikost textu
void	sender, e	text_uprava_siroke	Upraví styl textu
void	sender, e	text_uprava_sikme	Upraví styl textu
void	sender, e	text_uprava_preskrtnute	Upraví styl textu
void	sender, e	text_uprava_podtrzene	Upraví styl textu
void	sender, e	prg_csharp	Zapne označování syntaxe C# kódu
void	sender, e	ostatni_info	Otevře informační okno
void	sender, e	txb_KeyPress	Pokud je zaplé zvýraznění syntaxe, aktualizuje toto zvýrazňování
<b>Class Form_ulozit</b>			
void	sender, e	form_nacten	Při otevření formu vyresetuje proměnnou result
void	sender, e	btn_save_Click	Zaznamená rozhodnutí uživatele
void	sender, e	btn_notsave_Click	Zaznamená rozhodnutí uživatele
void	sender, e	btn_cancel_Click	Zaznamená rozhodnutí uživatele
void	sender, e	form_zavren	Při pokusu o zavření formu zaznamená rozhodnutí uživatele
<b>Class Soubor</b>			
void	filepath, txb	otevrittxt	Načte ze souboru čistě text
void	filepath, txb	otevritrtf	Načte ze souboru text s úpravou (typ rtf)

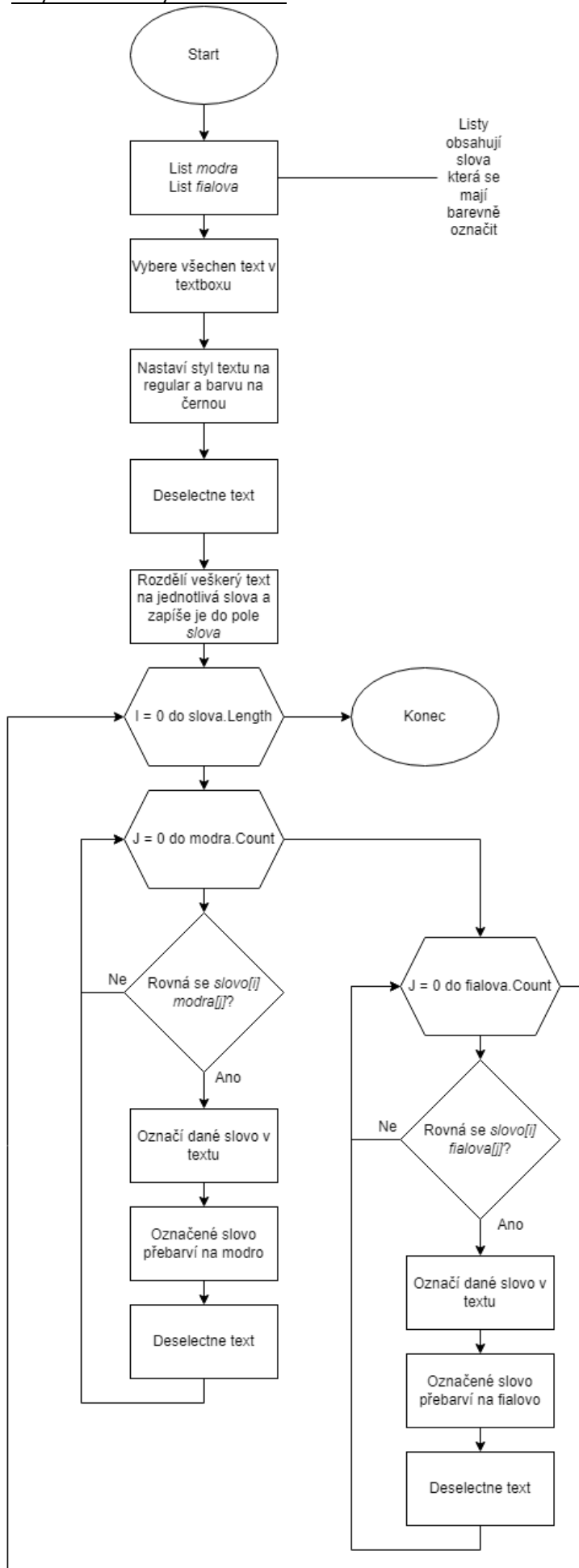
void	filepath, txb	ulozittxt	Uloží do souboru čistě text
void	filepath, txb	ulozitrtf	Uloží do souboru text i s jeho úpravou (typ rtf)
<b>Class Text</b>			
void	algn, txb	zarovnat	Zarovná text doleva, na střed nebo doprava
void	clr, txb	barva	Změní barvu textu
void	fnt, txb	font	Změní font textu
void	txb	csharp	Zvýrazňuje syntax C# kódu

## Vývojový diagram:

### Ukládání:



## Zvýraznění syntaxe C#:



## Zdrojový kód:

### Ukládání a načítání souboru:

```
private void soubor_otevrit_click(object sender, EventArgs e)
{
    form_ulozit.ShowDialog(); // zeptá se uživatele jestli chce uložit aktuální soubor
    switch (form_ulozit.result)
    {
        case "save":
        {
            if (filepath == "") // Pokud uživatel doposud soubor neukládal, program se
            zeptá kam má být obsah uložen
            {
                if (saveFileDialog.ShowDialog() == DialogResult.OK)
                {
                    filepath = saveFileDialog.FileName;
                    if (filepath.Contains(".txt") == true)
                    {
                        soubor.ulozittxt(filepath, txb);
                    }
                    if (filepath.Contains(".rtf") == true)
                        soubor.ulozitrtf(filepath, txb);
                }
                else return;
            }
            if (filepath.Contains(".txt") == true) // pokud uživatel ukládá soubor jako
            .txt, uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
            {
                soubor.ulozittxt(filepath, txb);
            }
            if (filepath.Contains(".rtf") == true) // pokud uživatel ukládá soubor jako
            .rtf, uloží text včetně všech jeho úprav
                soubor.ulozitrtf(filepath, txb);
            break;
        }
        case "cancel":
        {
            return;
        }
    }
    txb.Text = "";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        filepath = openFileDialog.FileName;
        if (filepath.Contains(".txt") == true) // pokud je soubor .txt, načte pouze čistý
        text
        {
            soubor.otevrittxt(filepath, txb);
        }
        if (filepath.Contains(".rtf") == true) // pokud je soubor .rtf, načte text i s
        úpravami
            soubor.otevritrtf(filepath, txb);
    }
    else return;
    this.Text = filepath + " - GatoPad--";
}

private void soubor_novy_click(object sender, EventArgs e)
{
    form_ulozit.ShowDialog(); // zeptá se uživatele jestli chce uložit aktuální soubor
    switch (form_ulozit.result)
    {
        case "save":
        {
            if (filepath == "") // Pokud uživatel doposud soubor neukládal, program se
            zeptá kam má být obsah uložen
            {
```

```

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            filepath = saveFileDialog.FileName;
            if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá
soubor jako .txt, uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
            {
                soubor.ulozittxt(filepath, txb);
            }
            if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá
soubor jako .rtf, uloží text včetně všech jeho úprav
            {
                soubor.ulozitrtf(filepath, txb);
            }
            else return;
        }
        txb.Text = "";
        filepath = ""; // Vytváří se nový soubor, který se bude ukládat jinak
        this.Text = "GatoPad--";
        break;
    }
    case "notsave":
    {
        txb.Text = "";
        filepath = ""; // Vytváří se nový soubor, který se bude ukládat jinak
        this.Text = "GatoPad--";
        break;
    }
}

private void soubor_ulozit_click(object sender, EventArgs e)
{
    if (filepath == "") // Pokud uživatel doposud soubor neukládal, program se zeptá kam má
být obsah uložen
    {
        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            filepath = saveFileDialog.FileName;
            if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá soubor jako
.txt, uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
            {
                soubor.ulozittxt(filepath, txb);
            }
            if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá soubor jako
.rtf, uloží text včetně všech jeho úprav
            {
                soubor.ulozitrtf(filepath, txb);
            }
            else return;
        }
        if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá soubor jako .txt, uloží
se pouze čistě text, což zajistí kompatibilitu s jinými programy
        {
            soubor.ulozittxt(filepath, txb);
        }
        if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá soubor jako .rtf, uloží
text včetně všech jeho úprav
        {
            soubor.ulozitrtf(filepath, txb);
            this.Text = filepath + " - GatoPad--";
        }
    }

    private void soubor_ulozitjako_click(object sender, EventArgs e)
    {
        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            filepath = saveFileDialog.FileName;
            if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá soubor jako .txt,
uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
            {

```



```

        soubor.ulozitrtf(filepath, txb);
    }
    if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá soubor jako .rtf,
uloží text včetně všech jeho úprav
        soubor.ulozitrtf(filepath, txb);
    }
    else return;
    this.Text = filepath + " - GatoPad--";
}

private void form_zavren(object sender, FormClosingEventArgs e)
{
    form_ulozit.ShowDialog();
    switch(form_ulozit.result)
    {
        case "save": // zeptá se uživatele jestli chce uložit aktuální soubor
        {
            if (filepath == "") // Pokud uživatel doposud soubor neukládal, program se
zeptá kam má být obsah uložen
            {
                if (saveFileDialog.ShowDialog() == DialogResult.OK)
                {
                    filepath = saveFileDialog.FileName;
                    if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá
soubor jako .txt, uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
                    {
                        soubor.ulozittxt(filepath, txb);
                    }
                    if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá
soubor jako .rtf, uloží text včetně všech jeho úprav
                    {
                        soubor.ulozitrtf(filepath, txb);
                    }
                }
                else return;
            }
            if (filepath.Contains(".txt") == true) // pokavaď uživatel ukládá soubor jako
.txt, uloží se pouze čistě text, což zajistí kompatibilitu s jinými programy
            {
                soubor.ulozittxt(filepath, txb);
            }
            if (filepath.Contains(".rtf") == true) // pokavaď uživatel ukládá soubor jako
.rtf, uloží text včetně všech jeho úprav
            {
                soubor.ulozitrtf(filepath, txb);
            }
            break;
        }
        case "notsave":
        {
            return;
        }
        case "cancel":
        {
            e.Cancel = true;
            return;
        }
    }
}

```

#### Zvýraznění syntaxe:

```

public void csharp(RichTextBox txb)
{
    List<string> fialova = new List<string>() // seznam slov upravovaných funkcí csharp
    {
        "if", "else", "switch", "case", "return", "break", "while", "for", "try", "catch"
    };

    List<string> modra = new List<string>() // seznam slov upravovaných funkcí csharp
    {
        "public", "static", "private", "string", "char", "int", "bool", "double", "float",
        "void"
    };
}

```

```

};

List<string> znaky = new List<string> // list oddělovačů textu
{
    " ", "\n", "\t", "(", ")", "{", "}", "[", "]"
};

int pozice = 0; // pozice kurzoru

txb.SelectAll();
txb.SelectionFont = new Font(txb.SelectionFont, FontStyle.Regular);
txb.SelectionColor = Color.Black;
txb.DeselectAll();
pozice = 0; // kurzor je na začátku textu
string[] slova = txb.Text.Split(' ', '\n', '\t', '(', ')', '{', '}', '[', ']'); // slova
se rozdělí a zapíše do pole
for(int i = 0; i < slova.Length; i++)
{
    for(int j = 0; j < modra.Count; j++)
    {
        if (slova[i] == modra[j])
        {
            txb.SelectionStart = pozice;
            txb.SelectionLength = modra[j].Length;
            txb.SelectionColor = Color.Blue;
            txb.DeselectAll();
        }
    }
    for (int j = 0; j < fialova.Count; j++)
    {
        if (slova[i] == fialova[j])
        {
            txb.SelectionStart = pozice;
            txb.SelectionLength = fialova[j].Length;
            txb.SelectionColor = Color.Purple;
            txb.DeselectAll();
        }
    }
    pozice = pozice + slova[i].Length; // posune kurzor na konec slova
    txb.Select(pozice, 1); // označí další znak za aktuálním slovem
    for (int j = 0; j < znaky.Count; j++)
    {
        if (txb.SelectedText == znaky[j]) // přičte k pozici 1 pokud další znak je
oddělovač
        {
            pozice++;
        }
    }
    txb.DeselectAll();
}
}

```

## Závěr:

Programu stále chybí několik podstatných funkcí pro textový editor, jako například možnost tisku. Také zvýraznění syntaxe je pouze základní, dalo by se rozšířit o detailnější zvýrazňování nebo třeba další programovací jazyky. Jinak ale program funguje dobře a je i velice slušně ošetřený proti chybám.