



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

<b>Předmět:</b>	<b>Vývoj aplikací</b>
<b>Téma:</b>	<b>Soubory</b>
<b>Vyučující:</b>	<b>Ing. Milan Káža</b>
<b>Třída:</b>	<b>EK3</b>
<b>Hodina:</b>	<b>11</b>
<b>Číslo:</b>	<b>V/5</b>

## Programování v jazyce C a C#

### 10. Soubory a práce s nimi

Soubor v informatice označuje pojmenovanou sadu dat uloženou na datovém médiu, se kterou lze pracovat jako s jedním celkem.

Obsahem souboru mohou být různá data. Mnoho souborů reprezentuje jediný objekt a obsahuje jenom jeden druh dat (například textový dokument, zvukový záznam), ale existují i soubory složené (archivní soubory, knihovny), které obsahují množství dílčích objektů a souborů; také kancelářské dokumenty mohou obsahovat vložené objekty (např. obrázky a tabulky), u multimediálních i jiných souborů mluvíme o kontejnerech, které mají vlastní formát a mohou obsahovat několik objektů (obraz, zvuk, titulky) různých formátů, přičemž formát jednotlivých složek není příliš závislý na formátu kontejneru. Podle toho, jak má být obsah souboru interpretován lze soubory rozdělit na textové a binární.

Soubory, které se používají při programování jsou hlavičkové, které načítají vložené funkce, to může být např. matematické funkce (sinus, cosinus, PI, odmocnina atd), dále funkce pro práci se soubory (např. otevření souboru, uložení atd.). Další využití hlavičkových souborů je při programování mikroprocesorů, kdy na začátku uvádí typ mikroprocesoru pro načtení registrů, vstupně-výstupních bran atd. Samozřejmě by se nemusely používat tyto hlavičkové soubory, ale programátor by musel všechno napsat do programu, proto hlavičkové soubory usnadňují práci programátorům.

Programy pracují nejčastěji s textovými soubory, do nich uživatelé ukládají data a načítají z nich data, speciálním typem textového souboru je databáze. Samozřejmě že s dalšími typy souborů je možné v programu pracovat, pokud je tak nastaven.

Může nastat otázka, proč v dnešní době, kdy jsou k dispozici prostředky Microsoft Office jako Word, Excel apod. se využívá stále textových „jednoduchých“ souborů. Odpověď je jednoduchá např. možnost kódování dat je daleko jednodušší v textovém souboru než ve Wordu, dále soubory jsou malé a snadno přenositelné což soubor vytvořený ve Wordu se neotevře na každém počítači.

## 10.1. Otevření souboru

Soubor se otevírá pro čtení, zápis a pro čtení a zápis. Pokud je nastavené, že soubor je možné otevřít jen pro čtení tak si uživatel přečte pouze obsah souboru, ale nemá možnost ho nijak upravit, pokud je nastavení souboru pro zápis, otevře se uživateli nová nepopsaná plocha a on udělá zápis, ten se může uložit pod jménem např. datum. V případě nastavení souboru pro čtení a zápis to znamená, že uživatel může si data přečíst a do souboru, většinou na konec vložit nová data.

### Otevření souboru pro čtení

V jazyce C se vloží hlavičkový soubor `stdio.h`, dále se vloží datový typ `FILE` a napíše se jméno proměné s hvězdičkou, což značí v C ukazatele. Dále se napíše příkaz k otevření souboru a ověří se zda byl soubor otevřen a potom se vloží příkaz cyklu zde je vyžit `while` dokud není nalezen konec souboru a soubor se zobrazí.

```
#include <stdio.h>    //hlavičkový soubor

FILE *filename;      //ukazatel na soubor
int znaky;           //proměnná znaky typu integer

main() {
    if ((filename=fopen ("data.txt", "r"))==NULL) //ověření otevření souboru data.txt
    {
        printf("soubor se nepodarilo otevrit");
        exit (1);
    }
    while((znaky=fgetc(filename), feof(filename))==NULL) { //čte znaky dokud není konec
        putchar(znaky);
    }
}
```

Tento příklad je uveden pro jazyk C ještě v DOS-vském provedení může se napsat i v dalších produktech C++ jak pod Windows

tak pod Linux.

V jazyce C a vyšších se udává jak se soubor použije. Pokud je vloženo písmeno „r“ jak je v příkazu otevření je soubor určen jen pro čtení, pokud by bylo vloženo písmeno „w“ je určen soubor pro zápis a pokud písmeno „a“ je soubor určen pro zápis a čtení.

Při otevírání binárního souboru se zápis uvádí takto: pro čtení „rb“  
pro zápis „wb“  
pro čtení a zápis „ab“

V jazyce C# pro otevření souboru pro čtení využívá metodu **StreamReader**, která je odvozena od **System.IO.File**, a instanci třídy **FileStream**. Pro otevření se zavádí příkaz **FileMode.Open** na začátek je nutné deklarovat **using System.IO**, tato deklarace zajišťuje otevření, čtení a zápis do souboru.

```
using System;
using System.IO;    //deklarace System.IO podpora pro otevření souboru

static string jméno = "data.txt"; //jméno souboru, který se bude načítat
static void Main(string[] args)
{
    FileStream fs = new FileStream(jméno, FileMode.Open); //funkce pro
                                                         // otevření souboru

    StreamReader sr = new StreamReader(fs);
    string s = sr.ReadLine();                          //čte do konce souboru
    while(s != null)
    {
```

```
int i = Int32.Parse(s);  
Console.WriteLine(i);  
s = sr.ReadLine();  
}
```

Tento příklad je platný pro C# a je možné ho aplikovat pouze pod Windows. Tak jako v jazyce C se zadává jak se souborem bude pracovat tak je tomu stejně v jazyce C#. Pro práci se soubory je určena třída `System.IO.File` a z ní se odvozují další třídy a podpora pro práci se soubory, několik typů z této třídy bude zde uvedeno, všechny ale ne, zbytek je možné dohledat na internetu a nebo v použité literatuře.

#### Třída `System.IO.FileMode` (způsob otevření souboru)

Konstanta	Význam
<code>Open</code>	Otevře existující soubor
<code>Truncate</code>	Otevře existující soubor a vymaže jeho obsah
<code>Create</code>	Vytvoří nový soubor a pokud existuje bude přemazán
<code>CreateNew</code>	Vytvoří nový soubor, pokud existuje vzniká výjimka <b><code>IOException</code></b>
<code>OpenOrCreate</code>	Otevře existující soubor a pokud neexistuje tak je vytvořen
<code>Append</code>	Otevře soubor s možností vpisování na konec

**Třída System.IO.FileAccess (přístup k souboru)**

Konstanta	Význam
<b>Read</b>	Povoluje čtení
<b>Write</b>	Povoluje zápis
<b>ReadWrite</b>	Povoluje čtení a zápis

**Třída System.IO.FileShare (možnost sdílení s více programy)**

Konstanta	Význam
<b>None</b>	Zakazuje sdílení
<b>Read</b>	Povoluje čtení ze sdíleného souboru
<b>Write</b>	Povoluje zápis dosdíleného souboru
<b>ReadWrite</b>	Povoluje čtení a zápis ze sdíleného souboru

**Otevření souboru pro zápis**

Řešení pro jazyk C

```
#include <stdio.h>
FILE *fw; //ukazatel na soubor
int i; //proměnná i
main()
{
```

```
fw=fopen("pokus.txt","w"); //otevření souboru pro zápis
  for(i=1;i<=10;i++)      //zápis čísel každé na novou řádku
    fprintf(fw,"%d\n",i);
fclose(fw);              //uzavření souboru
}
```

Na tomto příkladu je řešení pro jazyk C, pokud soubor neexistuje tak je vytvořen v případě, že existuje je přepsán a po ukončení zůstanou nová data, ale při každém otevření se soubor automaticky vymaže.

Řešení pro jazyk C#

```
using System.IO;

static string jmeno = "data.txt"; //jméno souboru, který se bude otevírat
static void Main(string[] args)
{
  FileStream fs = new FileStream(jmeno, FileMode.Create); //otevření souboru pro zápis
  StreamWriter sw = new StreamWriter(fs);
  for(int i = 0; i < text.GetLenght; i++) //načítání napsaného textu
    sw.WriteLine(i);
  sw.Close(); //uzavření souboru
}
```

I v tomto případě se bude soubor otevírat tak, že původní data vymaže a v případě, že nebude existovat tak se soubor vytvoří a uloží pod jménem **data.txt**. Do souboru se ukládají znaky a čte se jejich délka.

Práce se soubory je velmi důležitá proto dbejte na jejich vyzkoušení.

### Kontrolní otázky

1. Co je to soubor a jak se soubory rozdělují.
2. Vytvořte příklad pro čtení ze souboru.
3. Vytvořte příklad pro zápis do souboru.

### *Použitá literatura:*

*Miroslav Virius: C# pro zelenáče, 1.vydání, Neocortex, s.r.o Praha, 2002*

*Pavel Herout: Učebnice jazyka C, 3.vydání, KOOP České Budějovice, 1997*

*Miroslav Virius: od C++ k C#, 1.vydání, KOOP České Budějovice, 2002*