

Programování

Soubory a složky



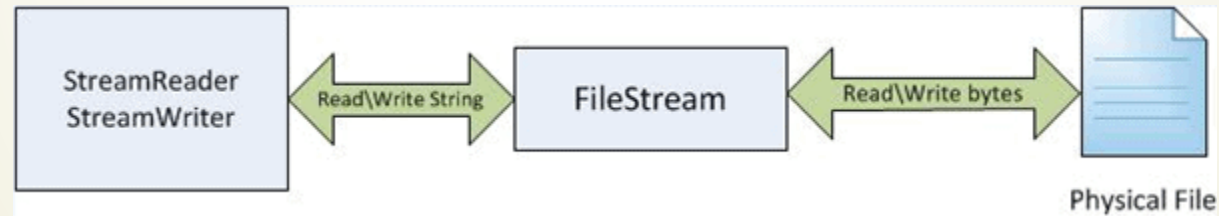


Stream

- ▶ Součást knihovny System.IO
- ▶ Vytváří standard pro přenos souborů v datových proudech
- ▶ Abstraktní třída umožňující čtení/zápis z různých zdrojů (soubor, paměť, ...)
- ▶ Následující třídy implementují třídu Stream podle konkrétního zdroje
- ▶ FileStream – fyzické soubory
- ▶ MemoryStream – operační paměť
- ▶ BufferedStream – jiné proudy (streams)
- ▶ NetworkStream – síťové sockety
- ▶ PipeStream - procesy

StreamReader, StreamWriter

- Obě třídy slouží jako pomoc pro snazší čtení a zápis dat
- Konvertují čtené datové bytes do znaků případně řetězců
- Konvertují zapisované znaky a řetězce do datových bytes
- Lze je používat pro různé typy streamů



Příklad čtení a zápisu do textového

Výpis souboru do konzole

```
string path = "c:\\myFile.txt";  
using (StreamReader sr = new StreamReader(path))  
{  
    while (sr.EndOfStream != true) {  
        Console.WriteLine(sr.ReadLine());  
    }  
    sr.Close();  
}
```

Zápis do soubor

```
string path = "myfile.txt";  
using (StreamWriter sw = new StreamWriter(path, true))  
{  
    sw.WriteLine("ABC");  
    sw.WriteLine("DEF");  
    sw.Close();  
}
```

Neexistující soubor pro zápis

Try - catch

```
string path = "c:\\file.txt";
try
{
    using (StreamReader sr = new StreamReader(path))
    {
        while (!sr.EndOfStream)
        {
            Console.WriteLine(sr.ReadLine());
        }
        sr.Close();
    }
}
catch (FileNotFoundException e)
{
    Console.WriteLine(e.Message);
}
```

File

```
string path = "c:\\file.txt";
if (File.Exists(path))
{
    using (StreamReader sr = new StreamReader(path))
    {
        while (!sr.EndOfStream)
        {
            Console.WriteLine(sr.ReadLine());
        }
        sr.Close();
    }
}
else
{
    //Co dělat v případě, že soubor neexistuje
}
```

File

- Statická třída pro práci s textovým souborem
- Pro práci s třídou File není třeba ustanovovat datové proudy
- Pokud chceme mít větší kontrolu nad prací se souborem využijeme třídu **FileInfo**

```
//kontrola, zda soubor existuje  
bool isFileExists = File.Exists("C:\\ soubor.txt");  
  
//vytvoření kopie souboru  
File.Copy("C:\\soubor.txt", "D:\\novySoubor.txt");  
  
//zobrazit poslední datum přístupu  
DateTime lastAccessTime = File.GetLastAccessTime("C:\\soubor.txt");
```

Directory

- Statická třída pro práci s adresáři
- Pokud chceme mít větší kontrolu nad prací s adresářem využijeme třídu **DirectoryInfo**

```
string sourceDirectory = @"C:\current";
string archiveDirectory = @"C:\archive";

try
{
    IEnumerable<string> txtFiles;
    txtFiles = Directory.EnumerateFiles(sourceDirectory, "*.txt");

    foreach (string currentFile in txtFiles)
    {
        string fileName = currentFile.Substring(sourceDirectory.Length + 1);
        Directory.Move(currentFile, Path.Combine(archiveDirectory, fileName));
    }
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
```