

Namespace

- Namespace je jmenný prostor,
- syntakticky vypadá podobně jako třída,
- určuje území platnosti identifikátorů,
- lze vnořovat namespace do namespace,
- nelze vnořit namespace do třídy.

Namespace II

- U nelokálních identifikátorů je nutno říct, ve kterém namespace jsou,
- chceme-li používat identifikátory z konkrétního namespace častěji, můžeme použít `using`, například: `using System;`
- Příklad správně: `using System;`
`Console.WriteLine();`
- Příklad správně: `System.Console.WriteLine();`
- Příklad špatně: `using System.Console;`
`WriteLine();`
špatně, protože `Console` je statická třída, ne jmenný prostor.

Namespace III

- Pomocí `using` lze vytvářet aliasy tříd:
- `using <alias>=<třída>;`
- Příklad: `using c = System.Console;`
`c.WriteLine();`
- Nyní už rozumíme celému obrázku, který Visual Studio vygeneruje,
- až na to, že ne nutně víme, co nám jednotlivé namespaces umožňují,
- zájemci to ovšem mohou různými způsoby zjišťovat.

Vstup a výstup

Textový soubor

- Standardní vstup se řídí ze třídy `Console` v namespace `System`.
- Textové soubory se ovládají velmi podobně pomocí `StreamReader` a `StreamWriter`.
- `StreamReader` a `StreamWriter` jsou v namespace `System.IO`, tedy:
- ```
System.IO.StreamReader r= new
System.IO.StreamReader(@"c:\temp\file.txt");
```
- Instance třídy `StreamReader` resp. `StreamWriter` má stejný charakter jako v Pascalu proměnná typu `text` assignovaná ke konkrétnímu souboru.
- V Pascalu jsme jméno proměnné typu `text` předávali každé funkci, v C# voláme dotyčnému readeru (resp. writeru) metody

# Vstup a výstup II

Textový soubor některé metody tříd `StreamReader` a `StreamWriter`

- `void r.Close()` // zavře reader,
- `string r.ReadToEnd()` //přečte soubor do konce,
- `w.Write(co)` //vypíše do writeru,
- `r.Read()` // přečte znak z readeru,
- `r.EndOfStream` // atribut určující konec vstupního streamu
- `@"....."` // pevný řetězec – nefungují v něm sekvence `\n`, `\r`, ....

# Vstup a výstup II

## Příklad

```
Zkopírování souboru System.IO.StreamReader r=new
System.IO.StreamReader("soubor.txt");
System.IO.StreamWriter w=new
System.IO.StreamWriter("lacina_kopie.txt");
w.Write(r.ReadToEnd()); w.Close();
```

# Vstup a výstup III

## Příklad

```
Zkopírování souboru po znacích System.IO.StreamReader
r=new System.IO.StreamReader("soubor.txt");
System.IO.StreamWriter w=new
System.IO.StreamWriter("lacina_kopie.txt");
while(!r.EndOfStream) w.Write((char)r.Read());
w.Close();
```

## Poznámky

- Metody `ReadLine()`, resp. `WriteLine()`,
- kódování češtiny – lze předat jako druhý parametr konstruktoru `Readeru/Writeru`:

- Příklad:

```
Encoding e=Encoding.GetEncoding(1250);
Encoding f=Encoding.GetEncoding(852);
System.IO.StreamReader r=new
System.IO.StreamReader("soubor.txt",e);
System.IO.StreamWriter w=new
System.IO.StreamWriter("lacina_kopie.txt",f);
w.Write(r.ReadToEnd()); w.Close();
```