

# C#程序设计及应用

唐大仕

dstang2000@263.net

北京大学

Copyright © by ARTCOM PT All rights reserved.



# 第7章 文本处理与正则表达式

唐大仕

[dstang2000@263.net](mailto:dstang2000@263.net)

<http://www.dstang.com>

# 内容提要



- 基于文本的应用
- 文本处理常用的几个类
- 正则表达式



# 基于文本的应用

01000100010010011001010111010100101001001010010010010111010100100101101001110110110



# 1 基于文本的应用

- 控制台应用程序
- Main()函数的参数—命令行参数
  - Main()函数可以带 string[]参数
  - Main()函数可以有返回值(int),也可以为void

# 使用Environment类



- .CommandLine .CommandLineArgs
- .MachineName . OSVersion
- .UserDomainName . UserName
- . GetEnvironmentVariables
- . CurrentDirectory . SystemDirectory
- .GetFolderPath(Environment.SpecialFolder.System)



# 文本处理常用的几个类

- Console类
  - Write WriteLine ReadLine
- String 类
- StringBuilder 类
- System.Text.Encoding类
  - .Default .UTF8 .GetEncoding
  - .GetBytes(str) .GetString(byte[])



# 正则表达式

01000100010010011001010111010100101001001010010010010111010100100101101001110110110





# 正则表达式

- 正则表达式 ( Regular Expression)
- 用来表示匹配某类文本
- 如：
  - `[0-9]{2,4}`
  - `^[a-zA-Z]+$`



- 正则表达式中几个主要要素:

- 字符 `ab \t \[ \u0020 \x20`

- `[a-zA-z]` 即 `\w`
    - `[0-9]` 即 `\d`    `[^0-9]` 即 `\D`
    - `\s` 即空白    `\S` 即非空白
    - `.` 表任意

- 次数 `{2,6}`

- `*` 即 `{0,}`    `+` 即 `{1,}`    `?` 即 `{0,1}`
    - `*?` `+`? `??` 表示lazy匹配

- `|`表示或者    `()`表示成组



## □位置限定

- ^ (首) xxxxxx\$ (尾)
- \b 单词边界 \B 非单词边界

## □(?<名称>xxxxxxxx) 表示对分组进行命名

- 在替换时, 使用 \${名称}
- 若不命名, 则为\$1, \$2等等 而\$0 表示整个匹配

## □(?i:xxxxxxxx) 表示选项(i表示忽略大小写)



# 正则表达式的选项

- 忽略大小写
- Multiline (影响 ^ 的含义)
- Singleline (影响 . 的含义)
- 预编译



# 使用Exprsso工具



- 参考:
- [http://msdn.microsoft.com/en-us/library/az24scfc\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/az24scfc(VS.71).aspx)



- `using System.Text.RegularExpressions;`
- Static方法:
  - ▣ `Regex.IsMatch( s, pattern )`
- 使用对象:
  - ▣ `Regex`对象
    - `.IsMatch(s)` `.Match(s)` `.Matches(s)`
    - `.Replace(s, r)`
  - ▣ `Match`对象
    - `.Groups` 各组 `.Value` 原串 `.Result(xxxxx)` 替换



- lrcPlayer 播放歌词
  - ▣ 使用文件及正则表达式