

Стандартни потоци в .NET



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. FileStream
2. MemoryStream
3. NetworkStream
4. Буферирани потоци
5. Други потоци (Crypto, Gzip и др.)



Класът FileStream

- Наследява класът **Stream** и поддържа всичките му методи и свойства
 - Поддържа четене, запис, позициониране и т.н.
- Конструкторът има параметри за:
 - Името на файла
 - Режимът на отваряне на файла
 - Режимът на достъп до файла
 - Дали файлът е заключен или се позволява едновременно достъп до него и от други приложения

Класът FileStream (2)

```
FileStream fs = new FileStream(string fileName, FileMode  
[,FileAccess [, FileShare]]);
```

Опционални параметри

- **FileMode** – режим на отваряне
 - Open, Append, Create, CreateNew, OpenOrCreate, Truncate
- **FileAccess** – режим на опериране с файла
 - Read, Write, ReadWrite
- **FileShare** – права за достъп за другите потребители докато файлът е отворен
 - None, Read, Write, ReadWrite

Запис на текст във файл – пример


```
string text = "Кирилица";  
var fileStream = new FileStream("../log.txt",  
                                FileMode.Create);  
try  
{  
    byte[] bytes = Encoding.UTF8.GetBytes(text);  
    fileStream.Write(bytes, 0, bytes.Length);  
}  
finally  
{  
    fileStream.Close();  
}
```

try-finally гарантира, че
потокът винаги ще се затвори

Encoding.UTF8.GetBytes()
връща прилежащите байтове
за символите в текста

Копиране на файл – пример

```
using (var source = new FileStream(SheepImagePath, FileMode.Open))
{
    using (var destination =
        new FileStream(DestinationPath, FileMode.Create))
    {
        while (true)
        {
            int readBytes = source.Read(buffer, 0, buffer.Length);
            if (readBytes == 0)
                break;
            destination.Write(buffer, 0, readBytes);
        }
    }
}
```



using автоматично затваря потока

Четене на низ в паметта – пример

```
string text = "In-memory text.";
byte[] bytes = Encoding.UTF8.GetBytes(text);
using (var memoryStream = new MemoryStream(bytes))
{
    while (true)
    {
        int readByte = memoryStream.ReadByte();
        if (readByte == -1)
            break;
        Console.WriteLine((char) readByte);
    }
}
```

Прост веб сървър – пример

```
var tcpListener = new TcpListener(IPAddress.Any, PortNumber);
tcpListener.Start();
Console.WriteLine("Listening on port {0}...", PortNumber);

while (true)
{
    using (NetworkStream stream = tcpListener.AcceptTcpClient().GetStream())
    {
        byte[] request = new byte[4096];
        stream.Read(request, 0, 4096);
        Console.WriteLine(Encoding.UTF8.GetString(request));

        string html = string.Format("{0}{1}{2}{3} - {4}{2}{1}{0}",
            "<html>", "<body>", "<h1>", "Welcome to my awesome site!", DateTime.Now);
        byte[] htmlBytes = Encoding.UTF8.GetBytes(html);
        stream.Write(htmlBytes, 0, htmlBytes.Length);
    }
}
```

Получава потока

Чете заявката

Записва отговора

Буферирани потоци

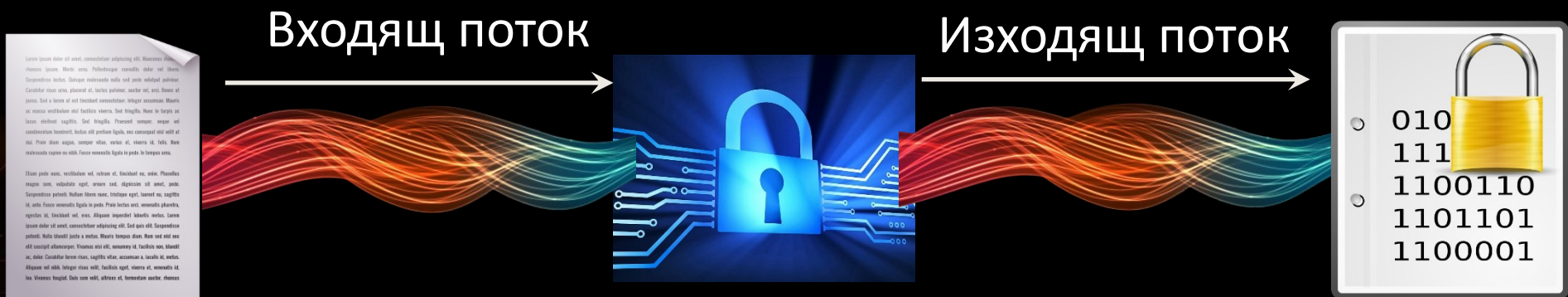
- Буферират данните и ефективно увеличават производителността
- Заявка за прочитане на дори 1 байт води до прочитане на още килобайти в аванс
 - Потокът ги пази във вътрешен буфер
- Следващото четене връща данни от вътрешния буфер
 - Много бърза операция

Буферирани потоци (2)

- Записаните данни се съхраняват във вътрешния буфер
 - Това е много бърза операция
- Когато буферът **се препълни**:
 - Се извиква **Flush()**
 - Данните се изпращат по назначение
- В .NET се използва класът **System.IO.BufferedStream**

Други потоци

- .NET поддържа специални потоци
 - Те работят като обичайните, но предоставят още функции
 - **CryptoStream** криптира при запис и декриптира при четене
 - **GzipStream** компресира и разкомпресира данните
 - **PipedStream** е за четене/запис на данни през няколко процеса



Обобщение

- **FileStream** позволява четене и запис във файлове
- **MemoryStream** е за четене и запис в паметта
- **NetworkStream** служи за мрежова комуникация
- **Буферираните потоци** подобряват производителността при четене и запис
- Другите видове потоци (**Crypto, Gzip** и др.) добавят допълнителна функционалност: криптиране, компресия, комуникация между процеси и т.н.



Стандартни потоци в .NET



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

