

Упражнения: Стандартни потоци







Problem 1. Копиране на двоичен файл

Напишете програма, която копира съдържанието на двоичен файл (например изображение, видео и т.н.) в друг с помощта на **FileStream**. **Не ви е позволено** да използвате класа **File** или други подобни помощни класове.







Problem 2. Разделяне на файл на части

Напишете програма, която взима произволен файл и го нарязва на **n** части. Разпишете следните методи:

- **Slice(string sourceFile, string destinationDirectory, int parts)** - разделя дадения **sourceFile** на **n** части и ги записва в **destinationDirectory**.

Входен файл	Изходяща директория
<p>parts = 5</p>  <p>SOLID-Logger.avi 00:32:06 678 MB</p>	<div><p>Part-0.avi 135 MB</p></div> <div><p>Part-2.avi 135 MB</p></div> <div><p>Part-4.avi 135 MB</p></div> <div><p>Part-1.avi 135 MB</p></div> <div><p>Part-3.avi 135 MB</p></div>

- **Assemble(List<string> files, string destinationDirectory)** - комбинира всички файлове в един, в реда, в който са подадени, и записва резултата в **destinationDirectory**.

Входни файлове	Изходяща директория
<div><p>Part-0.avi 135 MB</p></div> <div><p>Part-2.avi 135 MB</p></div> <div><p>Part-4.avi 135 MB</p></div> <div><p>Part-1.avi 135 MB</p></div> <div><p>Part-3.avi 135 MB</p></div>	 <p>assembled.avi 00:32:06 678 MB</p>




Използвайте **FileStream**. **Не ви е позволено** да използвате класа **Файл** или други подобни помощни класове.

Problem 3. Компресиране на нарязаните файлове

Променете вашата предходна програма, така че тя също така да **компресира** байтовете докато ги разделя на части и да ги **разкомпресира** когато ги обединява пак в **оригиналния** файл. Използвайте **GzipStream**.

Съвет: Когато взимате файловете от папката, подсигурете се, че взимате само тези с **.gz** разширение (защото може например да има и скрити файлове).

Source File	Compressed & Sliced	Decompressed & Assembled
-------------	---------------------	--------------------------

parts = 5  SOLID-Logger.avi 00:32:06 678 MB	 Part-0.gz GZ File 115 MB  Part-2.gz GZ File 113 MB  Part-4.gz GZ File 111 MB	 Part-1.gz GZ File 112 MB  Part-3.gz GZ File 114 MB	 assembled.avi 00:32:06 678 MB
---	---	--	--

Problem 4. Претърсване на директория

Претърсете дадена **директория** за **всички файлове** с дадено **разширение**. Търсете **само в първото ниво** на **папката** и опишете информацията за всеки **намерен файл** във **report.txt**.

Файловете трябва да бъдат **групирани** според своето **разширение**. **Разширенията** трябва да са **подредени** според **броя** на файловете, които притежават такова разширение, в **намаляващ ред**, и по **името** в **азбучен ред**.

Файловете с едно и също разширение трябва да бъдат **подредени** според своя **размер**.

report.txt трябва да бъде записан **на работния плот**. Подсигурете се, че винаги използвате валидния път към тази папка, независимо кой е логнатия потребител.

Вход	Изглед към папката	report.txt
.	<div> <div>Name</div> <div> <div>bin</div> <div>obj</div> <div>Properties</div> <div>01. Writing-To-Files.csproj</div> <div>App.config</div> <div>backup.txt</div> <div>controller.js</div> <div>log.txt</div> <div>Mecanismo.cs</div> <div>model.php</div> <div>Nashmat.cs</div> <div>Program - Copy.cs</div> <div>Program.cs</div> <div>Salimur.cs</div> <div>script.asm</div> <div>Wedding.cs</div> </div> </div>	<div> <div>.CS</div> <div>--Mecanismo.cs - 0.994kb</div> <div>--Program.cs - 1.108kb</div> <div>--Nashmat.cs - 3.967kb</div> <div>--Wedding.cs - 23.787kb</div> <div>--Program - Copy.cs - 35.679kb</div> <div>--Salimur.cs - 588.657kb</div> <div>.txt</div> <div>--backup.txt - 0.028kb</div> <div>--log.txt - 6.72kb</div> <div>.asm</div> <div>--script.asm - 0.028kb</div> <div>.config</div> <div>--App.config - 0.187kb</div> <div>.csproj</div> <div>--01. Writing-To-Files.csproj - 2.57kb</div> <div>.js</div> <div>--controller.js - 1635.143kb</div> <div>.php</div> </div>

Problem 5. * Пълно претърсване на директория

Променете предходната си програма, така че тя да **обхожда рекурсивно** и всички **поддиректории** на първоначалната директория.

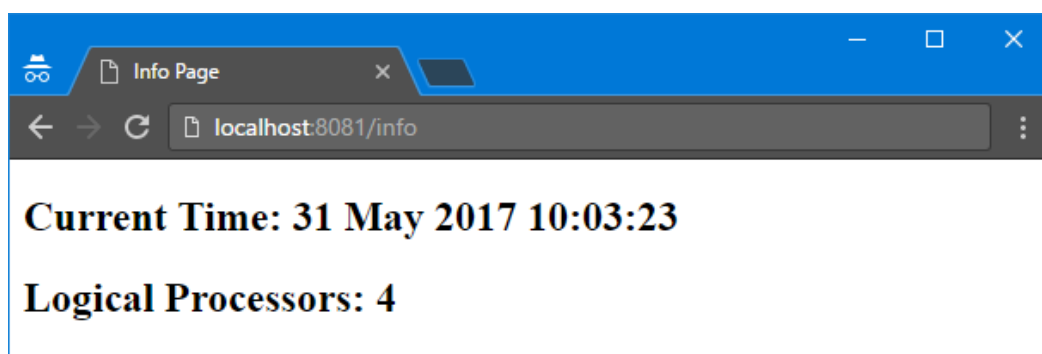
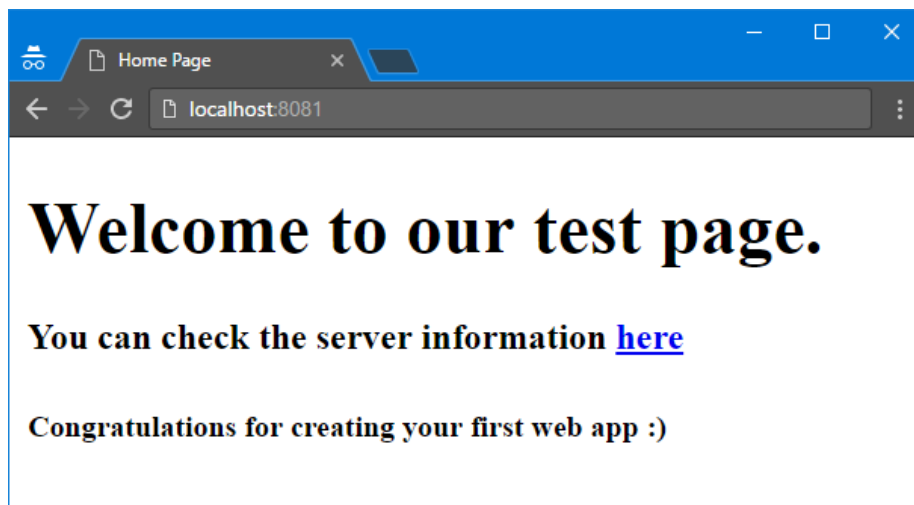
Problem 6. ** HTTP сървър

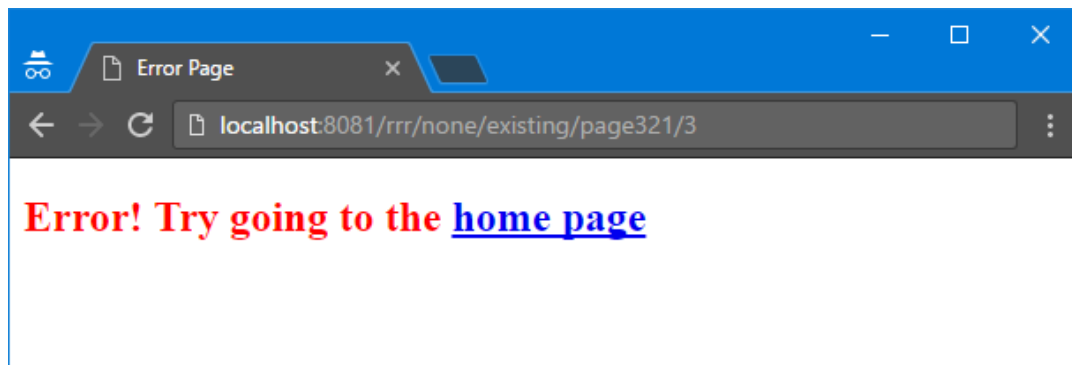
Създайте прост HTTP сървър, който ще може да **получава заявки** и да **връща подходящи отговори** според **заявления път**. **Прочетете** малко повече в Интернет как трябва да изглеждат HTTP заявка и нейния отговор. Създайте сайт с 3 страници:

- **1^{вата}** страница трябва да е достъпна на адрес **localhost:{port}/** - (това е основната директория). Тази страница съдържа само **приветствено съобщение** и **препратка** към втората страница
- **2^{рата}** страница трябва да е достъпна на адрес **localhost:{port}/info** – тази страница показва **текущото време** и **броят на логическите процесори** в компютъра
- **3^{тата}** трябва да бъде **страница за грешки** - ако потребителят се опита да достъпи каква да е друга страница върнете като отговор тази страница за грешки.

Ще ви бъдат предоставени за улеснение HTML файлове на страниците, които ни трябва. Можете да ги промените както прецените или дори да създадете ваши собствени HTML файлове.

Примери





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".



- Курсът е базиран на учебно съдържание и методика, предоставени от фондация "Софтуерен университет" и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

