# Упражнения: Стриймове, файлове и директории

Можете да проверите решенията си тук: <https://judge.softuni.org/Contests/4233/10-Streams-Files-and-Directories>.

**Бележка**: За тези проблеми следвайте инструкциите за необходимите методи и класове. За всеки проблем изпратете компресирана папка на вашия проект без папките **"bin"** и **"obj"** в нея.

## Четни редове

Напишете програма, която чете **текстов файл** (например text.txt) и отпечатва на конзолата всички **четни редове**. Номерата на редовете започват от **0**. Използвайте **StreamReader.** Преди да ги отпечатате, заменете стойностите **{"-", ",", ".", "!", "?"}** с **"@"** и **обърнете реда на думите**.

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class EvenLines  {  static void Main()  {  string inputFilePath = @"..\..\..\text.txt";  Console.WriteLine(ProcessLines(inputFilePath));  }  public static string ProcessLines(string inputFilePath)  {  }  } |

### Примери

|  |  |
| --- | --- |
| **Входен файл: text.txt** | **Изход (на конзолата)** |
| -I was quick to judge him, but it wasn't his fault.  -Is this some kind of joke?! Is it?  -Quick, hide here. It is safer. | fault@ his wasn't it but him@ judge to quick was @I  safer@ is It here@ hide @Quick@ |

## Нечетни редове

Напишете програма, която чете файл (примерно input.txt) и записва всеки **нечетен** ред в друг файл. Броя на редовете **започват от 0**.

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class OddLines  {  static void Main()  {  string inputFilePath = @"..\..\..\Files\input.txt";  string outputFilePath = @"..\..\..\Files\output.txt";  ExtractOddLines(inputFilePath, outputFilePath);  }  public static void ExtractOddLines(string inputFilePath, string outputFilePath)  {  // TODO: напишете вашия код…  }  } |

### Примери

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| Two households, both alike in dignity, In fair Verona, where we lay our scene, From ancient grudge break to new mutiny, Where civil blood makes civil hands unclean. From forth the fatal loins of these two foes A pair of star-cross'd lovers take their life; Whose misadventured piteous overthrows Do with their death bury their parents' strife. | In fair Verona, where we lay our scene,  Where civil blood makes civil hands unclean.  A pair of star-cross’d lovers take their life;  Do with their death bury their parents’ strife |

## Номера на редове

Напишете програма, която чете **текстов файл** (например text.txt) и добавя **номера на редовете** и **брой букви** и **пунктуационни знаци**. Резултатът трябва да се напише на **друг текстов файл** (например output.txt). Използвайте статичния клас, за да **прочетете** и да **напишете** всички редове на входния и изходния файл.

**Бележка: използвайте следната структура**

|  |
| --- |
| public class LineNumbers  {  public static void ProcessLines(string inputFilePath, string outputFilePath)  {  }  } |

### Примери

|  |  |
| --- | --- |
| **text.txt** | **output.txt** |
| -I was quick to judge him, but it wasn't his fault.  -Is this some kind of joke?! Is it?  -Quick, hide here. It is safer. | Line 1: -I was quick to judge him, but it wasn't his fault. (37)(4)  Line 2: -Is this some kind of joke?! Is it? (24)(4)  Line 3: -Quick, hide here. It is safer. (22)(4) |

## Брой думи

Напишете програма, която чете списък от думи от даден файл (примерно words.txt) и намира колко пъти се повтаря всяка дума в друг файл (примерно text.txt). Съвпадението трябва да бъде **case-insensitive. Резултатът трябва да бъде написан в изходен файл (примерно output.txt). Сортирайте думите по повтаряне в низходящ ред.**

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class WordCount  {  public static void CalculateWordCounts(string wordsFilePath, string textFilePath, string outputFilePath)  {  // TODO: напишете вашия код…  }  } |

### Примери

|  |  |  |
| --- | --- | --- |
| **words.txt** | **text.txt** | **output.txt** |
| quick is fault | -I was quick to judge him, but it wasn't his fault.  -Is this some kind of joke?! Is it?  -Quick, hide here…It is safer. | is - 3  quick - 2  fault - 1 |

## Копиране на двоичен файл

Напишете програма, която **копира** съдържанието от двоичен файл (например **copyMe.png**) до друг двоичен файл (например **copyMe-copy.png**) чрез **FileStream**. Не ви е позволено да използвате класа **File** или други подобни класове.

**Бележка:** използвайте следната структура:

|  |
| --- |
| public class CopyBinaryFile  {  public static void CopyFile(string inputFilePath, string outputFilePath)  {  }  } |

## Обхождане на директория

Напишете програма, която **преминава през дадена директория** за всички файлове с дадено **разширение**. Трябва да търси през **първото ниво** на директорията. Напишете информация за всеки файл в report.txt. Файлът трябва да бъде **запаметен на десктопа**. Файловете трябва да бъдат **групирани по разширения**. Разширенията трябва да бъдат **сортирани** по броя на файловете в низходящ ред и по име във възходящ ред. Файловете на всяко разширение трябва да бъдат **сортирани** по техния размер. Уверете се, че пътят до директорията винаги ще бъде **валиден**.

**Бележка:** използвайте следната структура:

|  |
| --- |
| public class DirectoryTraversal  {  public static string TraverseDirectory(string inputFolderPath)  {  }  public static void WriteReportToDesktop(string textContent, string reportFileName)  {  }  } |

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Директория** | **report.txt** |
| . | directory | .cs  --Mecanismo.cs - 0.994kb  --Program.cs - 1.108kb  --Nashmat.cs - 3.967kb  --Wedding.cs - 23.787kb  --Program - Copy.cs - 35.679kb  --Salimur.cs - 588.657kb  .txt  --backup.txt - 0.028kb  --log.txt - 6.72kb  .asm  --script.asm - 0.028kb  .config  --App.config - 0.187kb  .csproj  --01. Writing-To-Files.csproj - 2.57kb  .js  --controller.js - 1635.143kb  .php  --model.php - 0kb |

## Копиране на съдържанието на директория

Напишете метод, който **копира директория с файлове** (без поддиректориите) в друга директория. Входната и изходната папка ще бъдат дадени като вход от конзолата. Ако изходната папка съществува, трябва да я изтриете (заедно със съдържанието в нея).

**Бележка:** използвайте следната структура:

|  |
| --- |
| public class CopyDirectory  {  public static void CopyAllFiles(string inputPath, string outputPath)  {  }  } |

## \* Зипване и извличане

Напишете програма, която създава **ZIP** файл (архив), като държи даден входен файл и извлича ZIP файла от архива в отделен изходен файл.

* Използвайте файла copyMe.png от ресурсите като входен файл в ZIP файл (например archive.zip)
* Извлечете файла от архива в нов файл по ваш избор (например extracted.png)

Ако вашият код е успешен, входният и изходният файл трябва да са еднакви.

**Бележка:** използвайте следната структура:

|  |
| --- |
| public class ZipAndExtract  {  public static void ZipFileToArchive(string inputFilePath, string zipArchiveFilePath)  {  }  public static void ExtractFileFromArchive(string zipArchiveFilePath, string fileName, string outputFilePath)  {  }  } |

### Насоки

* Използвайте класа **ZipFile**
* Записът в ZIP файла трябва да съдържа само името на файла без неговия път.

## Обединяване на текстови файлове

Напишете програма, която чете **съдържанието на два файла** (примерно input1.txt и input2.txt) и ги **съединява** в трети файл (примерно output.txt). Съединението трябва да бъде в следния формат:

* Ред 1 от input1.txt
* Ред 1 от input2.txt
* Ред 2 от input1.txt
* Ред 2 от input2.txt
* …

Ако някой от файловете има **повече редове** от другия, добавете **в края на изхода** редовете, които **не могат** да бъдат съпоставени с другия файл.

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class MergeFiles  {  public static void MergeTextFiles(string firstInputFilePath, string secondInputFilePath, string outputFilePath)  {  // TODO: напишете вашия код…  }  } |

### Примери

|  |  |  |
| --- | --- | --- |
| **input1.txt** | **input2.txt** | **output.txt** |
| 1  3  5 | 2  4  6  7 | 1  2  3  4  5  6  7 |

## Разделяне / Обединяване на двоични файлове

Даден ви е входен двоичен файл (примерно example.png). Напишете програма, която **разделя** файла на два с **еднакви** размери (примерно part-1.bin и part-2.bin). Когато размерът на входния файл е **нечетен**, първата част трябва да бъде с **1 байт** **повече** от втората.

След като разделите входния файл, съединете двата файла в **нов** (примерно example-joined.png). Полученият файл трябва да бъде същият като първоначалния.

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class SplitMergeBinaryFile  {  public static void SplitBinaryFile(string sourceFilePath, string partOneFilePath, string partTwoFilePath)  {  // TODO: напишете вашия код…  }  public static void MergeBinaryFiles(string partOneFilePath, string partTwoFilePath, string joinedFilePath)  {  // TODO: напишете вашия код…  }  } |

## Извличане на специални байтове

Даден ви е двоичен файл (примерно **example.png**) и текстов файл (примерно **bytes.txt**), съдържащ списък от битове в обхвата [0…255]. Напишете програма, която извлича всички **еднакви битове** в двата файла в изходен двоичен файл (примерно output.bin).

**Бележка:** използвайте следната структура:

|  |
| --- |
| public class ExtractBytes  {  public static void ExtractBytesFromBinaryFile(string binaryFilePath, string bytesFilePath, string outputPath)  {  // TODO: напишете вашия код…  }  } |

### Примери

|  |  |  |
| --- | --- | --- |
| **example.png** | **bytes.txt** | **output.bin** |
|  | 20  46  183  212 |  |

## Размер на папка

Дадена ви е папка във файловата система (примерно TestFolder). Изчислите **размера** на файловете в тази папка, **включително** **подпапките**. Резултатът трябва да бъде написан в друг файл (примерно оutput.txt). Размерът трябва да бъде в **килобайти**.

**Бележка**: използвайте следната структура:

|  |
| --- |
| public class FolderSize  {  public static void GetFolderSize(string folderPath, string outputFilePath)  {  // TODO: напишете вашия код…  }  } |

### Примери

|  |
| --- |
| **оutput.txt** |
| 0.0869140625 KB |