io

Jon Sporring

November 21, 2019

1 Lærervejledningn

Emne Input/output

Sværhedsgrad Let

2 Introduktion

3 Opgave(r)

1. Write a program `myFirstCommandLineArg` which takes an arbitrary number of arguments from the command line and writes each argument as, e.g.,

```
$ mono myFirstCommandLineArg.exe a sequence of args
4 arguments received:
0: "a"
1: "sequence"
2: "of"
3: "args"
```

   The program must exit with the status value 0.

2. Make a program `myFirstReadKey` which continuously reads from the keyboard using the `System.Console.ReadKey()` function. The following key-presses must result in the following:

   'a' writes "left\n" to the screen
   's' writes "right\n" to the screen
   'w' writes "up\n" to the screen
   'z' writes "down\n" to the screen
   shift+'q' quits the program

   All other key-presses must be ignored. When the program exits, then the exit status must be 0.

3. Make a program `myFirstReadFile` which

   (a) opens the text file "myFirstReadFile.fsx" as a stream using the `System.IO.File.OpenText` function,

   (b) reads each individual character using the `System.IO.StreamReader.Read` function,

   (c) writes each character to the screen using the `printf` function, and

   (d) closes the stream using `System.IO.FileStream.Close`.

   The program's exist status must be 1 or 0 depending on whether there was an error or not.

4. Make a program `myFirstWriteFile` which

   (a) opens a new text file "newFile.txt" as a stream using the `System.IO.File.CreateText` function,

   (b) writes the characters 'a' ... 'z' one at a time to the file to using `System.IO.StreamReader.Write`, and

   (c) closes the stream using `System.IO.FileStream.Close` function.

   The program's exist status must be 1 or 0 depending on whether there was an error or not.

5. Make a program with the function,

   ```
   printFile : unit -> unit
   ```

   which initiates a dialogue with the user. The function must ask the user for the name of a file, and if it exists, then the content is to be printed to the screen. The program must return 1 or 0 depending on whether the specified file exists or not.

6. Make a program with the function,

   ```
   printWebPage : url:string -> string
   ```

   which reads the content of the internet page `url` and returns its content as a string option.

7. Lav en lommeregner,

   ```
   simpleCalc : unit -> unit
   ```

   som starter en uendelig dialog med en bruger. Brugeren skal kunne indtaste simple regnestykker på postive heltal, og hvert regnestykke må kun bestå af en enkelt af følgende binære operatorer: `+`, `-`, `*`, `/`. Resultatet skal kunne genbruges i den efterfølgende beregning med navnet `ans`.

8. Make a program with a function,

   ```
   fileReplace :
     filename:string -> needle:string -> replace:string -> unit
   ```

   which replaces all occurrences of the string `needle` with the string `replace` in the file `filename`. Your solution must use the `System.IO.File.OpenText`, `ReadLine`, and `WriteLine` functions.

9. I html-standarden angives links med `<a></a>` tags, f.eks. kunne et link til Googles hjemmeside skrives som `<a href="https://google.com">Tryk her for Google</a>`. Der skal laves et program

```
countLinks : url:string -> int
```

som henter internetsiden angivet med argument `url` og som tæller, hvor mange links der er på siden ved at tælle antallet af `<a` delstrenge.

Bemærk: Langt de fleste internetsider kræver et gyldigt certifikat for at dit program kan læse siden, og som udgangspunkt har mono ingen certifikater installeret. For at installere et nyttigt sæt certifikater kan du bruge `mozroots`, som er en del af Mono pakken. På Linux/MacOS gør følgende fra Konsollen:

```
mozroots --import --sync
```

På Windows gør du følgende (på samme linje)

```
mono "C:\Program Files (x86)\Mono\lib\mono\4.5\mozroots.exe" --import
--sync
```

Ret evt. stien, hvis din installation af mozroots ligger et andetsted. Derefter kan du læse de fleste sider uden at blive afvist.

Til besvarelsen skal der laves en kort afprøvning, og en kort beskrivelse af løsningen med argumenter for større valg, der er foretaget, for at nå til den givne løsning.

10. Filen `storeClausLilleClaus.txt` indeholder H.C. Andersens eventyr "Store Claus og Lille Claus" fra 1835. I skal skrive et program, som indlæser filen og udskriver hyppigheden af alle de ord, som bruges i eventyret, til filen `hyppighed.txt`. Hyppigheden skal være sorteret fra mest til mindst hyppige. Med ord skal forstås tegnfølger, som ikke indeholder whitespaces eller tegnsætning, og hvor store bogstaver er konverteret til små.