

# Programmering og Problemløsning

## Datalogisk Institut, Københavns Universitet

### Arbejdsseddel 9 - individuel opgave

Jon Sparring

25. november - 29. november.  
Afleveringsfrist: søndag d. 29. november kl. 17:00.

Emnerne for denne arbejdsseddel er:

- undtagelser (exceptions),
- input og output til skærm og tastatur,
- input og output til filer
- input fra internettet.

Opgaverne er delt i øve- og afleveringsopgaver. I denne periode skal I arbejde individuelt med jeres afleveringsopgaver. Regler for gruppe- og individuelle afleveringsopgaver er beskrevet i "Noter, links, software m.m." → "Generel information om opgaver".

## Øveopgaver

9ø.0 Lav en funktion,

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

som starter en dialog med brugeren. Programmet skal spørge brugeren om navnet på en fil, og derefter skrive filens indhold ud på skærmen.

9ø.1 Der skal laves et program

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

som erstatter alle forekomster af needle argumentet med replace argumentet i filen med navn filename. Løsningen skal som minimum bruge funktionerne System.IO.File.OpenText, ReadLine og WriteLine til at tilgå filerne. Der skal 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.

### 9ø.2 Lav en funktion,

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

som indlæser indholdet af internetsiden på url og returnerer resultatet som en streng.

### 9ø.3 Lav en lommeregner,

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

som starter en uendelig dialog med en bruger. Brugeren skal kunne indtaste simple regnestykker på positive 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.

## Afleveringsopgaver

### 9i.0 Der skal laves følgende implementationer af samme funktion

```
safeIndexIf : arr:'a [] -> i:int -> 'a  
safeIndexTry : arr:'a [] -> i:int -> 'a  
safeIndexOption : arr:'a [] -> i:int -> 'a option
```

De skal alle returnere værdien i arr på plads i, hvis i er et gyldigt index, og ellers håndtere fejlsituationen. Fejlsituationerne skal håndteres forskelligt:

- safeIndexIf må ikke gøre brug af [try-with](#) og må ikke kaste en undtagelse.
- safeIndexTry skal benytte [try-with](#), og ved fejltilstand skal failwith kaldes.
- safeIndexOption skal returnere None i en fejlsituation.

Der skal laves en kort afprøvning af alle 3 funktioner, der indebefatter at den indicerede værdi udskrives til skærmen som et heltal og ikke som en option type. Afprøvningen skal også afprøve korrekt håndtering af funktionernes evt. kastede undtagelser. Lav en kort beskrivende sammenligning af metodernes evne til at håndtere fejltilstande.

### 9i.1 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 delstreng.

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 andet sted. 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.

- 9i.2 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.

Afleveringen skal bestå af

- en zip-fil
- en pdf-fil

Zip-filen skal indeholde en `src` mappe og filen `README.txt`. Mappen skal indeholde fsharp koden, der skal være en fsharp tekstfil per fsharp-opgave, og de skal navngives `9i0.fsx` osv. De skal kunne oversættes med `fsharpc` og den oversatte fil skal kunne køres med `mono`. Funktioner skal dokumenteres ifølge dokumentationsstandarden. Filen `README.txt` skal ganske kort beskrive, hvordan koden oversættes og køres. Pdf-filen skal indeholde jeres rapporten oversat fra  $\text{\LaTeX}$ .

God fornøjelse.