



Programmering og Problemløsning

Aflevering 7i

Adam Ingwersen,

Datalogisk Institut
Københavns Universitet

November 4, 2016



1 Delopgave 7i.0

1.1 `safeIndexIf`

Funktionen er bygget således, at denne tager et brugerinput. Ved korrekt indicering - returneres værdien af det *x*'te index i arrayet *A*.

Givet den betragtede array af længde 6, vil enhver værdi, der ikke er mellem 0 og 6 karakteriseres som invalid indeksering - hertil returneres default-værdien for typen array. I det tilfælde, hvor input er af en anden type end *integer*, bliver brugeren præsenteret med følgende fejlmeddelelse; `"System.FormatException: Input string was not in a correct format."`.

Fejlmeddelelsen kunne præsenteres pænere ved at kalde funktionen med en `try/with` struktur fremfor bare at bruge `println` - dog bliver der i opgaveformuleringen specifikt spurgt efter en funktion, der ikke anvender `try/with`.

1.2 `safeIndexTry`

Denne funktion anvender `try/with` strukturen til at kaste en fejl ved brug af *failwith*. Hvis input til *x* er af korrekt type, men ugyldig værdi, vil fejlen `"Index is of invalid size"` returneres. Hvis ikke input *x* er af korrekt type vil fsharp returnere samme fejlmeddelelse som *safeIndexIf*. For *safeIndexTry* har jeg valgt ikke at specificere typen for inputtet *x* - men grundet fsharp's typegenkendelse, vil denne ikke acceptere, at andre typer end *integer* anvendes til at indicere et array.

1.3 `safeIndexOption`

I *safeIndexOption* anvendes igen `try/with` strukturen, men nu med *option*-typerne *Some* og *None* samt *Option.get*-argumentet. Dette resulterer i, at et *integer* returneres ved gyldigt input til *x* - mens *null* returneres hvis *x* er out-of-bounds. *Option.get* anvendes udelukkende idet opgaveformuleringen dikterer, at ingen af funktionerne må returnere andet end heltal.

Hvis ikke input *x* er af korrekt type vil fsharp returnere samme fejlmeddelelse som *safeIndexIf*.

2 Delopgave 7i.1

Problemstillingen i denne opgave, løses ved at konstruere to hjælpefunktioner som det første. En funktion, der læser (`readFile`) og en funktion, der skriver (`writeFile`). Disse anvendes efterfølgende i `fileReplace`. Disse to funktioner anvender `streamReader`- hhv. `streamWriter`-typerne. Der kan nu læses og skrives - for at erstatte en streng med en anden streng i teksten, anvendes `Regex.Replace((text : string), (pattern : string), (replacement : string))` fra *System.Text.RegularExpressions* namespace.

Konstruktionen anvendt i programmet bevirker, at uanset indholdet af `filename`, `needle` og `replace`, køres programmet. I det tilfælde, hvor indholdet af tekstfilen er tom, returneres bare endnu en tom streng - og denne skrives til den i forvejen tomme tekstfil. Såfremt `needle` ikke findes i tekstfilen, erstattes intet. `replace` vil kun blive skrevet til tekstfilen, hvis `needle` findes i tekstfilen.

Der er ikke anvendt eksplicit fejlhåndtering - men en mulig udbygning kunne være, at prompte brugeren, hvis f.eks. ingen matches findes - eller hvis den indgående tekstfil er tom.

3 Delopgave 7i.2

Til dette program, inviteres brugeren til at vælge et URL samt et søgekriterium. Programmet forløber som følger:

1. Interager med brugeren via konsol : læs brugerinput
2. Hent url-stream på baggrund af brugers input
 - (a) Sikr at URL'et er validt
 - (b) Skriv *url* til *string*
 - (c) Skriv *string* til tekst-fil lokalt
3. Verificér at filen er blevet lavet og eksisterer
4. Læs fil som *string*
5. Interager med brugeren via konsol : læs brugerinput
6. Udfør pattern match i indlæst fil på baggrund af brugers input

Programmet kan potentielt set fejle efter 1. brugerinteraktion, såfremt brugeren indtaster et ugyldigt URL. I dette tilfælde kastes en fejl: *"Sitet kan ikke findes. Indtast venligst gyldig URL"*.

Programmet har ligeledes mulighed for at fejle efter 2. brugerinteraktion, hvis brugeren indtaster en streng der returnerer 0 matches. Denne fejl håndteres ved at definere en exception og bruge try/with med printfn, som anvist i *fsharpNotes11-12.pdf* under *Listing 11.5*.

Funktionen *countHtmlTag* anvender namespace: *System.Text.RegularExpressions* for bekvemhed. Herunder funktionen *Regex.Matches*, som både tager regular-expressions eller en simpel-tekststreng som søgekriterium.