Datalogi C
Henning Christiansen 24/11-2004

Opgave til stoffet gennemgået ved forelæsningen 23/11-2004

Denne opgave handler om at analysere HTML-tekst vha. Javas regulære udtryk og tilhørende facliciteter. Der følger en række af spørgsmål, som kan løses fra en ende af, så mange som vi nu kan nå.

Som testdata benyttes kildeteksten til kursets hjemmeside, som du finder her http://www.ruc.dk/~henning/DatCE2004/. Det er ikke nødvendigt for opgaven at finde en autoriseret definition af hvordan HTML skal skrives, du kan blot løse opgaven, så den passer med denne kildetekst.

De væsentligste strukturer i et HTML-dokument er angivet ved såkaldte tags som er af formen:

<EtEllerAndet>

eller

<EtEllerAndet attribut= "blabla" attribut= "blabla" >

Nogle strukturer sluttes med en såkaldt end-tag, </EtEllerAndet>, men andre kan stå alene, f.eks.
.

Det er ligegyldigt om man bruger store eller små bogstaver.

Spørgsmål 1
Skriv regulære udtryk vha. den notation som Java API stiller tilrådighed, som svarer til sådanne tags og end-tags. Skriv et Java-program, som er istand til at indlæse den omtale kildetekst som en "character sequence", og skriv en metode som læser sig igennem filen og udskriver navnene på de indgående tags.
NB: Måske du skulle starte med at eksperimentete med en mindre fil.

Spørgsmål 2
Nu skal vi skrive et program som checker om tags benyttes korrekt, dvs.

- Alle tags, som kræver en end-tag, har en sådan korrekt placeret efterfølgende
- Se bort fra om attributter inde i tags er "de rigtige".

Spørgsmål 2.1
Definér en Java-klasse for tags, med underklasser for start-tag og end-tag.
Skriv en metode baseret på dine regulære udtryk fra spørgsmål 1, som leverer tilbage et tag-objekt tilbage svarende til den læste tag.

Sæt toString-metoder på de nævnte klasser, og brug dem til en ny løsning på spørgsmål 1.

Spørgsmål 2.2

(1)

For at holde styr på at tagsene står rigtigt, kan vi ikke nøjes med regulære udtryk, men må bruge noget som minder om en kontekst-fri grammatik. Noget i stil med følgende.

struktur ::= *start-tag*(X) *strukturer* *end-tag*(X) | *tag-som-ikke-kræver-slut-tag*

strukturer ::= en sekvens af 0 eller flere eksemplarer af *struktur*

For at kontrollere at tagsene er placeret i overensstemmelse med denne grammatik kan vi benytte en algoritme nogenlunde i stil med følgende, hvor vi benytter en stak som datastruktur:

så længe der er *tag*s at læse
    t = næste *tag*;
    hvis t er en *start-tag*, læg den på stakken
    hvis t er en *tag-som-ikke-kræver-slut-tag,* spring den over
    hvis t er en *end-tag*, kontrollér at der ligger en matchende *start-tag*
        på toppen af stakken og fjern denne
      ellers udskriv en fejlmeddelelse
hvis stakken ikke er tom, skriv en fejlmeddelse

Spørgsmål 2.2.1
Foretag en håndsimulering af denne algoritme for at overbevise dig om at den fungerer korrekt.

Spørgsmål 2.2.2
Implementér i Java.