Afleveringsopgave 2

02102 Indledende Programmering

Carsten Nielsen, s123161
Søren Krogh Andersen, s123369

Arbejdsfordeling
Carsten Nielsen: Rapport: sektion 1.2 og 2, Programkode: Opgave 1 og 3
Søren Krog Andersen: Rapport sektion 1.1 og 3, Programkode: Opgave 1 og 2

3rd of March 2013

1 Opgave 1

2 Opgave 2

2.1 Beskrivelse og Design

Der ønskes et program der kan tjekke om en tekstreng indtastet af en bruger er et palindrom. Et palindrom er et stykke tekst der har samme sekvens af bogstaver ligemeget om den læses fra venstre til højre eller omvendt. Alle tegn der ikke er bogstaver, dvs. mellem, tal og specialtegn ignoreres. Proceduren programmet skal udføre er derfor:

1. Fjern alle specialtegn og mellemrum.
2. Evaluer om strengen har den samme sekvens af bogstaver i begge retninger

Programmet Palindrome implementerer disse 2 skridt direkte ved at splitte delskridtene op i enkelte metoder. Programmets brugerflade er konsollen hvor brugeren bliver bedt om at indtaste en tekststreng. Herefter bliver resultatet af evalueringen printet til konsollen, hvorefter brugeren bliver spurgt igen. Dette loop kører indtil brugeren skriver kommandoen "exit"i konsollen. Indtaster brugeren et tal vil dette evalueres som et palindrom da alle tegn der ikke er bogstaver ifølge definitionen ignoreres og inputtet er derfor den tomme streng der naturligvis er den samme læst fra begge sider.

2.2 Programtest

For at teste programmet's korrekthed skal følgene ting verificeres:

1. Programmet afbrydes når kommandoen "exit"modtages
2. Programmet kan ignorere alle tegn der ikke er bogstaver
3. Programmet kan skelne mellem strenge der er, og ikke er, palindromer

Der findes endnu et tilfælde, der dog ikke vil blive testet, dette er at programmet modtager en tekstreng der er for stor til at kunne håndteres. Dette er dog meget usandsynligt eftersom Java kan håndtere meget store tekstrenge og programmet ville tage meget lang tid om at evaluere den største tekstreng der kan være i input-bufferen.

For testpunkt nr. 1 indtastes "exit"kommandoen imens programmet venter på input og imens programmet er igang med at evaulere en lang streng. Dette giver følgene outputs

```
1 Input a string. Input "exit" to end the program.
  exit
3 Done!
```

Listing 1: Output fra testsekvens

Og imens programmet arbejder på en længere streng:

```
1 Input a string. Input "exit" to end the program.
  (lang streng vises ikke her)
3 exit
  ""
5 Is not a palindrome!
  Input a string. Input "exit" to end the program.
7 Done!
```

Listing 2: Output fra testsekvens imens programmet evaluerer

Her vises den indtastede streng ikke i outputtet da den overskrider konsol output grænsen i Eclipse. Det ses at programmet regner færdigt før det afsluttets, hvilket også fremgår af kildekoden.

Testpunkt nr 2. og 3. kan afprøves med det samme input. Anvendes palindromeksemplet angivet i opgavebeskrivelse "En af dem, der tit red med fane."kan det afprøves om programmet kan genkende palindromet selvom det indeholder specialtegn. Tilføjes et bogstav et sted i strengen, bortset fra i midten, kan det også verificeres at programmet kan afvise en streng som værende et palindrom. Dette ses fra programmets konsol output:

```
Input a string. Input "exit" to end the program.
En af dem, der tit red med fane.
"En af dem, der tit red med fane."
Is a palindrome.
```

Listing 3: Genkendelse af palindrom

```
Input a string. Input "exit" to end the program.
En af dem, der tit red med fane.E
"En af dem, der tit red med fane.E"
Is not a palindrome!
```

Listing 4: Afvisning af ikke-palindrom

Med disse tests er der gennemført 100% kodedækning. Dette er selvfølgelig ikke en garanti for at programmet altid vil virke, men det er realistisk set umuligt at teste alle mulige input da der er utælleligt mange. Dog kunne der skrives en lille test-suite der generede n antal palindromer, saltede dem med alle specialkarakterene fra unicode settet og tjekke at programmet outputtede det rigtige. Men dette bedømmes som unødvendigt.

3 Opgave 3