

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



1 Opgave 1

1.1 Beskrivelse og Design

1.2 Programtest

For at teste at dette program virker som forventet er det nødvendigt at teste følgende:

1. Programmet håndterer hver dekade korrekt og lægger dekader korrekt sammen
2. Programmet håndterer fejl-input uden at bryde sammen og at det genkender exit kommandoen

For at teste at programmet håndterer hver dekade op til 100.000.000 som er den højeste dekade programmet har et romersk numeral for, er der inkluderet en metode kaldet *testDecades* i klassen *RomanNumerals* som tilgås ved at indtaste den skjulte kommando "test" i konsollen. Resultatet af dette ses nedenfor som et screenshot da listings pakken til latex ikke kan håndtere unicode karakteren for et overstreget bogstav.

```
Input any positive integer. Type "exit" to end the program
test
STARTING TEST
1*10^1 = X 2*10^1 = XX 3*10^1 = XXX 4*10^1 = XL 5*10^1 = L 6*10^1 = LX 7*10^1 = LXX 8*10^1 = LXXX 9*10^1 = XC
1*10^2 = C 2*10^2 = CC 3*10^2 = CCC 4*10^2 = CD 5*10^2 = D 6*10^2 = DC 7*10^2 = DCC 8*10^2 = DCCC 9*10^2 = CM
1*10^3 = M 2*10^3 = MM 3*10^3 = MMM 4*10^3 = M̄V 5*10^3 = V̄ 6*10^3 = V̄M 7*10^3 = V̄MM 8*10^3 = V̄MMM 9*10^3 = M̄X
1*10^4 = X̄ 2*10^4 = XX̄ 3*10^4 = XXX̄ 4*10^4 = XL̄ 5*10^4 = L̄ 6*10^4 = LX̄ 7*10^4 = LXX̄ 8*10^4 = LXXX̄ 9*10^4 = XC̄
1*10^5 = C̄ 2*10^5 = CC̄ 3*10^5 = CCC̄ 4*10^5 = C̄|V̄| 5*10^5 = |V̄| 6*10^5 = |V̄|C̄ 7*10^5 = |V̄|CC̄ 8*10^5 = |V̄|CCC̄ 9*10^5 = C̄|X̄|
1*10^6 = |X̄| 2*10^6 = |X̄||X̄| 3*10^6 = |X̄||X̄||X̄| 4*10^6 = |X̄||C̄| 5*10^6 = |C̄| 6*10^6 = |C̄||X̄| 7*10^6 = |C̄||X̄||X̄|
8*10^6 = |C̄||X̄||X̄||X̄| 9*10^6 = |X̄||C̄|
1*10^7 = |C̄| 2*10^7 = |C̄||C̄| 3*10^7 = |C̄||C̄||C̄| 4*10^7 = |C̄||D̄| 5*10^7 = |D̄| 6*10^7 = |D̄||C̄| 7*10^7 = |D̄||C̄||C̄|
8*10^7 = |D̄||C̄||C̄||C̄| 9*10^7 = |C̄||M̄|
1*10^8 = |M̄|
Input any positive integer. Type "exit" to end the program
```

Figur 1: Output fra testDecade metoden

Herefter testes om programmet kan kombinere tal i flere dekader. Resultatet af testen ses nedenfor.

```
1 Input any positive integer. Type "exit" to end the program
1337
3 MCCCXXXVII
```

Listing 1: Test af dekadekomination

Hvilket er korrekt.

Til sidst testes punkt nr. 2, at programmet afbrydes når brugeren indtaster "exit" kommandoen.

```
1 Input any positive integer. Type "exit" to end the program
exit
3 Done!
```

Listing 2: Test af exit kommando

Og at det kan håndtere input af negative tal samt ikke-tal.

```
1 Input any positive integer. Type "exit" to end the program
-42
3 Please enter a positive integer, that's a number without a decimal point, larger than
zero
Input any positive integer. Type "exit" to end the program
NaN
5 Please enter a positive integer, that's a number without a decimal point, larger than
zero
7 Input any positive integer. Type "exit" to end the program
```

Listing 3: Test af dårligt input

Disse tests opnår 100% kode dækning. Selvom dette selvfølgelig ikke er en garanti for at programmet altid vil virke vil vi ikke gå videre med tests da det ville kræve at teste alle tænkelige input, hvilket er umuligt med den tid vi har til rådighed. Yderligere er programmet ikke begrænset af integer overflow, så dette er ikke testet.

2 Opgave 2

2.1 Beskrivelse og Design

Der ønskes et program der kan tjekke om en tekststreng 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 følgende egenskaber 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 tekststreng der er for stor til at kunne håndteres. Dette er dog meget usandsynligt eftersom Java kan håndtere meget store tekststreng og programmet ville tage meget lang tid om at evaluere den største tekststreng 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 evaluere en lang streng. Dette giver følgende outputs

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

Listing 4: 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 5: 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:

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

Listing 6: Genkendelse af palindrom

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

Listing 7: 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 tjekkede at programmet outputtede det rigtige. Men dette bedømmes som unødvendigt.

3 Opgave 3

3.1 Beskrivelse og Design

3.2 Programtest