

DA-ALG1000

Obligatorisk oppgave 3: Prosjektrapport



Av: Martin Kowalik Gran

Forord

Just in time.

Innholdsfortegnelse

1.0 Innledning.....	3
1.1 Oppgaveteksten	3
2.0 Beskrivelse av programmet.....	3
2.1 Beskrivelse av fremgangsmåten.....	3
3.0 Brukerveiledning	4
3.1 UML	4
4.0 Feiltesting	4
5.0 Konklusjon	9

1.0 Innledning

Denne rapporten omhandler den tredje obligatoriske programmeringsoppgaven i faget DA-ALG1000 våren 2015. Oppgaven var å skrive et program som lagrer data ved bruk av hashing og lenkede lister.

1.1 Oppgaveteksten

Skriv et program som benytter teknikken for «Chaining» når det gjelder hashing. Programmet skal lese inn lengden på tabellen som skal inneholde referanse til hver enkelt lenket liste som skal genereres. Videre skal alle verdier, som skal lagres, leses inn. En kjøring bør typisk lese inn ca. 15-20 verdier. Programmet skal ha en egen funksjon/metode for hashing der indeksen finnes. Når programmet har generert de lenkede listene, skal teoretisk 'load factor' beregnes og skrives ut. Hele tabellen skal skrives ut på enkel måte. Programmet skal benytte klassene for SingelLinkedList og Node som du brukte i 1. obligatoriske oppgave. Rapport skal leveres med og denne skal inneholde eksempel på kjøring av programmet. Det er ikke nødvendig med GUI-løsning.

2.0 Beskrivelse av programmet

2.1 Beskrivelse av fremgangsmåten

Løsningen på programmet var å skrive et program som leste inn, hashet og lagret funksjoner i lenkede lister.

Programmet inneholder en «main» der det opprettes en tabell av typen «SingelLinkedList». Tabellens lengde defineres av bruker gjennom skanner og lagres som «int antall». Det opprettes objekter av typen SingeLinkedListLister ved navn hashing_Table til hver indeks i tabellen.

Hashingfunksjonen, hashing(int a, int b), finner modulo(rest i heltallsdivisjon) på verdien som leses inn ved å dele verdien på tilsvarende verdi av SingelLinkedLists lengde.

Main leser inn verdier som hashes og lagres i SingelLinkedList med indeks lik keyen hashingfunksjonen gir det innleste dataet.

Når antallet ønskede verdier er lest inn skrives teoretisk loadfaktor ut gjennom kall på metoden loadFactor(int a, int b)

Tilslutt skrives den innleste tabellen ut ved å skrive ut alle verdier lagret under de ulike indeksene i tabellen hashing_Table.

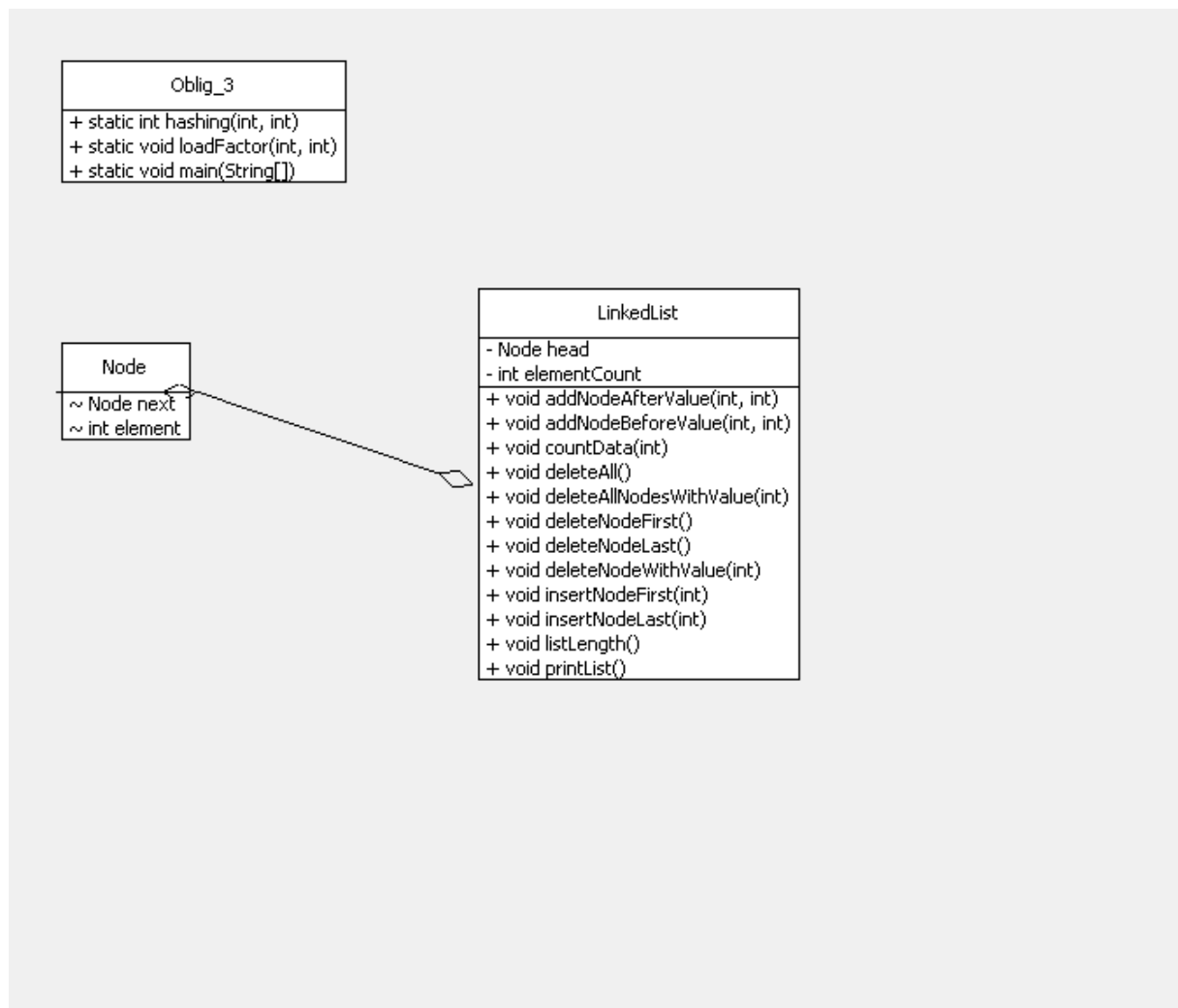
3.0 Brukerveiledning

Programmet ber bruker først om å velge antall indekser, dvs lister, man ønsker å opprette. Dette gjøres ved å skrive inn et heltall. Antallet lister avgjør også med hvilken verdi «hashingmetoden» bruker til å gi keys til de innskrevne verdiene. Avhengig av datasett bør derfor antall indekser vurderes.

Deretter ber programmet om en spesifisering av antall verdier man ønsker å skrive inn. Dette gjøres ved å skrive inn et heltall.

Programmet lagrer deretter dataene og skriver ut de lagrede dataene fra indeks 0 til indeks n.

3.1 UML



Figur 2: Figuren viser UML for programmet med variabler og metoder.

4.0 Feiltesting

Det ble gjennomført en gjennomkjøring av programmet der 20 verdier ble lest inn. Det ble valgt å ha en 5 «indekser». Da hashes verdiene gjennom en modulodeling modulo 5. De innleste verdiene var 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20.

Programmet deler deretter verdiene i 5 grupper. I testtifellet ble dette:

Indeks 0: 20,15,10,5

Indeks 1: 16,11,6,1

Indeks 2: 17,12,7,2

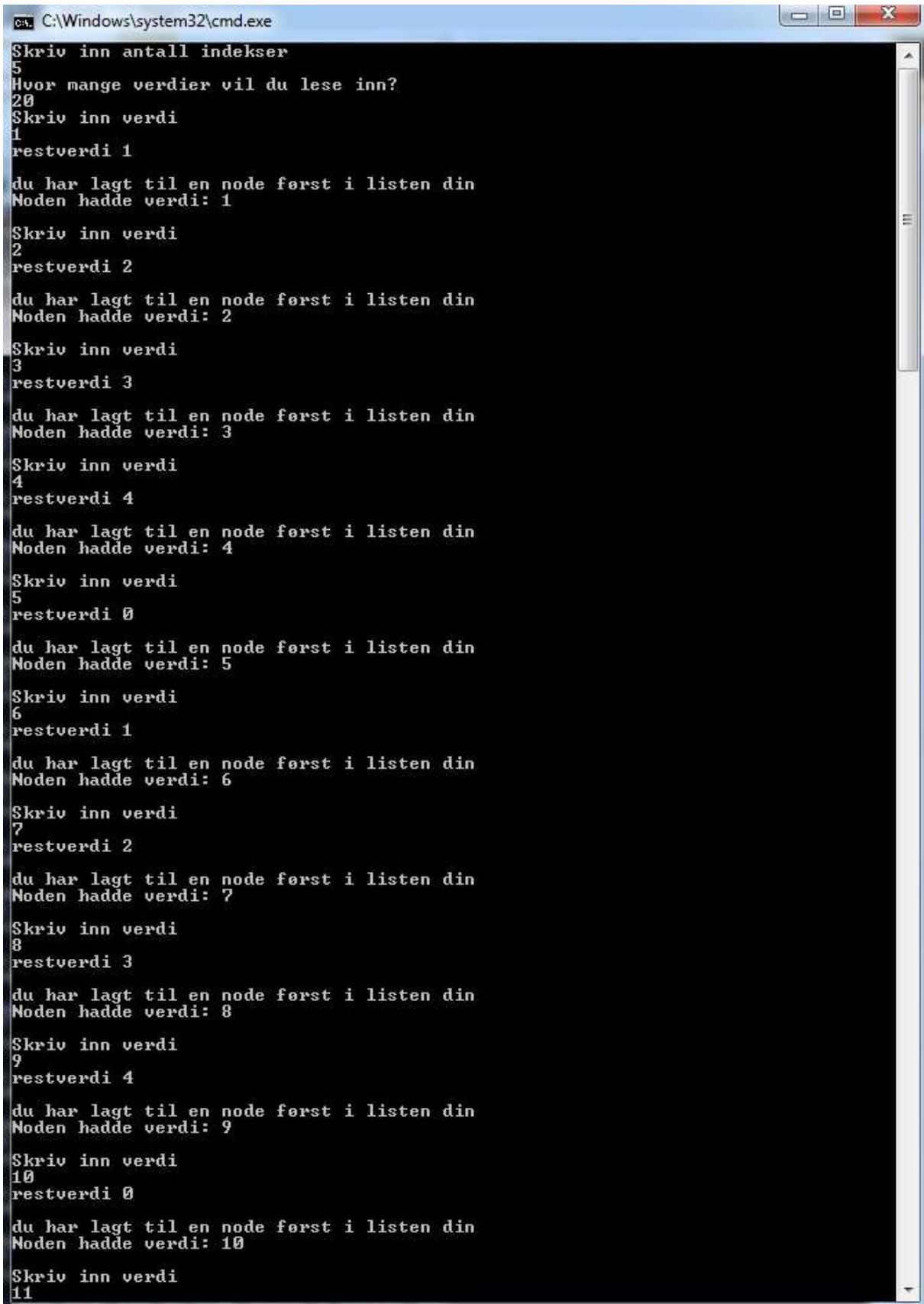
Indeks 3: 18,13,8,3

Indeks 4: 19,14,9,4

Teoretisk loadfaktor med denne konfigurasjonen ble beregnet til 0,2.

4.1 Kjøring av programmet.

Under vises skjermbilder av en gjennomkjøring av programmet.



```
C:\Windows\system32\cmd.exe
Skriv inn antall indekser
5
Hvor mange verdier vil du lese inn?
20
Skriv inn verdi
1
restverdi 1
du har lagt til en node først i listen din
Noden hadde verdi: 1
Skriv inn verdi
2
restverdi 2
du har lagt til en node først i listen din
Noden hadde verdi: 2
Skriv inn verdi
3
restverdi 3
du har lagt til en node først i listen din
Noden hadde verdi: 3
Skriv inn verdi
4
restverdi 4
du har lagt til en node først i listen din
Noden hadde verdi: 4
Skriv inn verdi
5
restverdi 0
du har lagt til en node først i listen din
Noden hadde verdi: 5
Skriv inn verdi
6
restverdi 1
du har lagt til en node først i listen din
Noden hadde verdi: 6
Skriv inn verdi
7
restverdi 2
du har lagt til en node først i listen din
Noden hadde verdi: 7
Skriv inn verdi
8
restverdi 3
du har lagt til en node først i listen din
Noden hadde verdi: 8
Skriv inn verdi
9
restverdi 4
du har lagt til en node først i listen din
Noden hadde verdi: 9
Skriv inn verdi
10
restverdi 0
du har lagt til en node først i listen din
Noden hadde verdi: 10
Skriv inn verdi
11
```

```
C:\Windows\system32\cmd.exe

10
restverdi 0

du har lagt til en node først i listen din
Noden hadde verdi: 10

Skriv inn verdi
11
restverdi 1

du har lagt til en node først i listen din
Noden hadde verdi: 11

Skriv inn verdi
12
restverdi 2

du har lagt til en node først i listen din
Noden hadde verdi: 12

Skriv inn verdi
13
restverdi 3

du har lagt til en node først i listen din
Noden hadde verdi: 13

Skriv inn verdi
14
restverdi 4

du har lagt til en node først i listen din
Noden hadde verdi: 14

Skriv inn verdi
15
restverdi 0

du har lagt til en node først i listen din
Noden hadde verdi: 15

Skriv inn verdi
16
restverdi 1

du har lagt til en node først i listen din
Noden hadde verdi: 16

Skriv inn verdi
17
restverdi 2

du har lagt til en node først i listen din
Noden hadde verdi: 17

Skriv inn verdi
18
restverdi 3

du har lagt til en node først i listen din
Noden hadde verdi: 18

Skriv inn verdi
19
restverdi 4

du har lagt til en node først i listen din
Noden hadde verdi: 19

Skriv inn verdi
20
restverdi 0

du har lagt til en node først i listen din
Noden hadde verdi: 20
```



```
C:\Windows\system32\cmd.exe
20
restverdi 0

du har lagt til en node først i listen din
Noden hadde verdi: 20

teoretisk loadfaktor er 0.2
index 0
20
15
10
5

index 1
16
11
6
1

index 2
17
12
7
2

index 3
18
13
8
3

index 4
19
14
9
4

Press any key to continue . . .
```

5.0 Konklusjon

Programmet virker ut fra bestillingens parametere. Den hasher verdier av innskreven type og gir tilbakemelding på loadfaktor basert på brukers ønske om antall indekser å lagre verdiene under. Programmets brukerveiledning bør leses for å unngå bruk av feil verdier ved input. Programmet kan tilpasses små datamengder eller store datamengder avhengig av hva bruker ønsker.