

MAT1030 - Eksamensnotater

Sjur Hernes & Eivind Lindbråten

07 June 2010

Innhold

1	Tall	1
1.1	Integere	1
1.2	desimaltall	2
2	Sannhetsverdier - boolean	2
2.1	presidensregler	3
3	Tekst - strenger og char	3
3.1	char	4
3.2	String	4
4	Arrayer	4
5	løkker	5
5.1	while	5
5.2	for	5
5.3	while .. do	5
6	hasjkart	5
7	Lenkede lister	6

1 Tall

Type	Lovlige verdier
byte	± 127
short	$\pm 32\,767$
int	$\pm 2\,147\,483\,647$
long	$\pm 9\,223\,372\,036\,854\,775\,807L$
float	$\pm 3.402\,823\,47\,E+38F$
double	$\pm 1.797\,693\,134\,862\,315\,70\,E+308$

negative tall er har alltid en større range

1.1 Integere

Integere er heltall, og lite annet.

Hvis vi vil konvertere en streng til en int sier vi:

```
int x = Integer.parseInt("123");
```

Hvis vi vil konvertere et annet tall til int skriver vi:

```
int x = (int) 3.14
```

Operasjon	Beskrivelse	Eksempel
+, -, *, /	De fire regneartene	1+1
++	legge til 1	i++
-	trekke fra 1	i--
%	modulo(rest)	5%2 -> 1
Math.sqrt(...)	Kvadratrot	Math.sqrt(4) -> 2
Math.pow(x, y)	Potens (x^y)	Math.pow(2, 3) -> 8

- Metodekall

- ++ og --
- * og /
- + og -
- -(som negativt fortegn)

husk at:

```
int x, y=1;
```

```
x = y++ + y + ++y;
```

nå er $x = 6$, fordi stykket blir $1 + 2 + 3$. først er $y = 1$, og så øker den slik at neste gang y brukes er den 2. når vi skriver $++y$ så vil y være 3 før vi legger det til.

1.2 desimaltall

Jeg velger å bare leke med floats, men reglene er ca det samme som for int.

konvertering fra heltall til desimaltall:

```
float x = (3 + 0.0) / 2;
```

altså vi bare legger til 0.0 som det som skjer først, slik at ett av tallene er et desimaltall, ellers vil vi få heltallsdivisjon. Enkelt og greit, utenom det gjelder de samme operatorene og de samme presidentsreglene.

2 Sannhetsverdier - boolean

- &&
 - Og
 - true && true
 - ||
 - * Eller
 - * true || false
- ikke b = !false
- < og >
 - mindre enn, større enn
 - b = x < y
- <= og >=
 - mindre eller lik
 - b = x <= y
- ==
 - er lik
 - b = x == y
- !=
 - er ikke lik
 - b = x != y

2.1 presidentsregler

- Metodekall
 - !
 - <, <=, >=, >
 - =, !
 - &&
 - ||

3 Tekst - strenger og char

vi har strengen s = "kake"

Navn	Forklaring	Eksempel
charAt(...)	tegnet i gitt posisjon(fra 0)	s.charAt(2)=='k'
length()	gir lengden på teksten	s.length()==4
substring(...)	delteksten fra- og tilposisjon gir indeksen og ut	s.substring(1,3)=="ak" s.substring(1)=="ake"
equals(...)	tester likhet mellom strenger	s.equals("kake")
indexOf(...)	posisjonen til tegnet/tekst	s.indexOf('a')==1
startsWith(...)	starter teksten med ...	s.startsWith("ka")
endsWith(...)	ender teksten med ...	s.endsWith("ke")
compareTo(...)	sammenligning av tekster	s.compareTo("bake")<0

3.1 char

en char-verdi er rett og slett en bokstav, den kan sammenlignes ('a' < 'b') og vil da sammenlignes ut i fra ascii-verdier (alle store bokstaver er mindre enn de små bokstavene).

3.2 String

En string er en rekke med char-verdier, altså ord. Man kan legge ord sammen med pluss-operatoren ("heisann" + " " + navn), man kan konvertere tall til strenger på denne måten

```
String s = "" + 42;
```

og verdien til s vil være "42".

Strenger kan også deles opp i arrayer ved hjelp av en split-funksjon. eks:

```
String[] t = s.split(" ");
```

4 Arrayer

Arrayer er en indeksert(fra 0) gruppe av objekter. Man må definere størrelsen når man lager objektet.

```
String[ ] a = new string[3];
```

man kan nå finne lengden på arrayet og bruke det som en `int a.length`;

5 løkker

navn	beskrivelse	eksempel
for	bestemt antall ganger alle objekter i array alle objekter i hash	<pre>for(int i=0;i<3;i++){ for(String s : a){ for(String s : hm.values())</pre>
while	i mens test er sann	<pre>while(b){}</pre>
do-while	utfører løkka før testen	<pre>do {} while(b);</pre>

5.1 while

while-løkker er kanskje den enkleste formen for løkker, den gjør en blokk kode så lenge predikatet i parameteret er sant.

Man trenger ikke nødvendigvis å kjøre en kodeblokk, for eksempel hvis man har en boolsk funksjon ("`kan()`") som parameteret kan man kjøre koden "`while(kan());`" og dermed si at man skal kjøre helt til "`kan()`" returnerer false.

5.2 for

for-løkker er en naturlig utvidelse av while da man ofte trenger tellere eller ting som skjer for hver gang kodeblokken skal kjøres, for eksempel en teller når man går igjennom et array.

- ```
for(int i=0;i<array.length;i++){
 – System.out.println(array[i])
}
```

Men for-løkker kan brukes på andre måter da den har en ganske enkel måte å oppføre seg på "`for(initialiseringskode;predikat;postkode)`", eneste som det er strengt hva man må ha i en for-løkke er predikatet. initialiseringen, kan være å sette en teller til noe, eller når man går igjennom noder i en graf, så kan man f.eks sette `for(Node n = root;n.next!=null;n=n.next)`

### 5.3 while .. do

Mindre brukt løkketype, egentlig en while-løkke hvor koden kjøres minst en gang før predikatet blir testet.

## 6 hasjkart

Hashmaps er en enkel måte å ordne mange objekter med et objekt som indeks.  
import java.util.\*

```
HashMap<string,Person> personregister = new HashMap<String,Person>();
```

| Metode                | beskrivelse                                                         |
|-----------------------|---------------------------------------------------------------------|
| put(nøkkel, peker)    | legge til objekt i HM                                               |
| get(nøkkel)           | hente peker til objekt                                              |
| remove(nøkkel)        | fjerne nøkkel fra HM                                                |
| containsKey(nøkkel)   | bool om nøkkelen er der                                             |
| containsValue(objekt) | bool om objektet er der                                             |
| values()              | lager en mengde av alle verdiene i HM,<br>brukes til itterering     |
| keySet()              | brukes til å lage en mengde av alle nøklene<br>brukes til iterering |
| isEmpty()             | returnerer true hvis tabellen er tom.                               |
| size()                | Metoden returnerer antall nøkler i tabellen                         |

## 7 Lenkede lister