## 2.0 Variabler(husk se javabog.dk af Jacob Nordfalk)

Variabler bruges til at opbevare og ændre data. En variabel kan opfattes som en navngiven papirlap, hvor der til enhver tid kan stå netop én ting.

Variabler skal altid erklæres, dvs. at man skal fortælle computeren, at der skal oprettes en variabel, hvad slags data den skal indeholde, og hvad den skal hedde.

***En variabel er et navn på et sted i computerens hukommelse, beregnet på at indeholde data af en bestemt type***

***En variabel erklæres ved at skrive:*** *variabel****type*** *variabel****navn****;*

***f.x int antal = 3; //int står for integer (hele tal)  
double fraktion = 7.5; //double står for kommatal (reelle tal)  
String mitnavn=”Oscar”;//String står for tekststreng(”tekst”)***

Det er en god vane at give variablerne sigende navne. Navnene bør starte med et lille bogstav.

I det følgende gennemgår vi to af Javas variabeltyper: int (heltal) og double (kommatal).

### 2.1 Heltal

En variabel af typen int (et heltal, eng.: integer) erklæres med

**int** tal;

*Efter 1. tildeling*Nu er der reserveret plads i hukommelsen til et heltal. Pladsen der blev reserveret er på 4 bytes = 32 bits. Man får fat i pladsen ved at bruge variabelnavnet 'tal'. Efter at variablen er erklæret kan den tildeles en værdi, dvs. man kan skrive data ind i den:

tal = 22;

Eller erklæring og tildeling af værdi på en gang

int tal = 22;

Nu er værdien af tal 22. (Du kender dette fra Wordmat’s tildeling af værdi til en variabel tal := 22)  
Vi kan bruge tal-variablen i stedet for at skrive 22, f.eks. til at skrive ud til skærmen:

System.out.println(”svaret på livet, universet og alt det der: ” + tal);

Her slår computeren op i hukommelsen, læser indholdet af tal-variablen og skriver det ud til skærmen (+'et vil blive forklaret i næste afsnit).  
Variabler kan, som navnet siger, ændre værdi. Det gør vi ved at tildele variablen en ny værdi:  
  
*Efter 2. tildeling*

tal = 42;

Herefter er den gamle værdi fuldstændigt glemt og erstattet med den nye. Når programudførelsen når et punkt, hvor variablen læses, vil det være den nye værdi, 42, der gælder.

***I en tildeling læses eller beregnes værdien på højre side og gemmes i variablen på venstre side. Dette svarer helt til Wordmat.***

Herunder er eksemplet i sin helhed (den væsentlige del af koden er fremhævet med fed):

// Eksempel på brug af en variabel

// koden skal være i filen Variabler.java

**public class** Variabler

{

**public static void** main (String[] args)

{

**int tal;**

**tal = 22;**

System.out.println(”Svaret på livet, universet og alt det der:” **+ tal**);

**tal = 42;**

System.out.println(”Undskyld, svaret er: ” **+ tal**);

}

}

Udskriften vil herefter se således ud

Svaret på livet, universet og alt det der: 22

Undskyld, svaret er: 42

### 2.2 Sammensætte strenge med +

Som det er vist i ovenstående eksempel, kan vi med tegnet **+** sætte strenge sammen med noget andet:

System.out.println(**”Svaret på livet, universet og alt det der: ” + tal**);

Herunder sætter vi to strenge sammen:

// Sammensæt to strenge med +

// koden skal være i filen HejVerden2.java

**public class** HejVerden2

{

**public static void** main (String[] args)

{

System.out.println(”Hej ” + ”Verden!”);

}

}

Herefter ser udskriften således ud

Hej Verden!

Det, der egentlig sker, er, at det hele bliver sat sammen til én streng, og den sendes til System.out.println().

***En streng + noget andet, sættes sammen til en samlet streng***

### 2.3 Beregningsudtryk

Man kan erklære flere variabler på samme linie:

**int** antalHunde, antalKatte, antalDyr;

antalHunde = 5;

antalKatte = 8;

Tildelinger kan indeholde regneudtryk på højre side af lighedstegnet. Udtrykket antalHunde + antalKatte udregnes, og resultatet lægges i variablen på venstre side (det er ikke tilladt at have beregningsudtryk på venstre side):

antalDyr = antalHunde + antalKatte;

Beregningsudtrykkene undersøges af Java ved at indsætte værdien af variablerne(hentes fra hukommelsen/RAM’en i PC’en). Her indsætter Java 5 + 8 og får 13, som lægges i antalDyr.

**public class** Dyreinternat

{

**public static void** main(String[] args)

{

**int** antalHunde, antalKatte, antalDyr;

antalHunde = 5;

antalKatte = 8;

//udregn summen

antalDyr = antalHunde + antalKatte;

// udskriv resultatet

System.out.println(”Antal dyr: ” + antalDyr);

antalHunde = 10;

// antalDyr er uændret

System.out.println(”Antal dyr nu: ” + antalDyr);

}

}

Antal dyr: 13

Antal dyr nu: 13

Beregningen sker én gang *på det tidspunkt, hvor kommandoen udføres*. Derfor er antalDyr's værdi ikke påvirket af, at vi sætter antalHunde til noget andet efter udregningen.

### Ligesom i almindelig matematik har \* (multiplikation) og / (division) højere prioritet end + og -. I Java skrives ”9 divideret med 3” som 9/3 og ”3 gange 3” som 3\*3. - Præcist som i MathCad.

### Resultatet af en heltalsudregning er også et heltal. Det skal man være opmærksom på ved division, hvor eventuelle decimaler efter kommaet smides væk. Heltalsudregningen 13 / 5 giver altså 2, fordi 5 går op i 13 to gange. For at gennemtvinge en normal division, kan man i stedet skrive 13./5. Resten ved heltalsdivision kan også beregnes, det kommer vi til i en senere lektionsplan.

### 2.4 Kommatal

Der findes mange andre variabeltyper end heltalstypen int. Hvis man vil regne med kommatal, bruger man typen double. En variabel af typen double erklæres med:

**double** højde;

Herved reserveres plads - 8 bytes = 64 bits – i hukommelsen.

Her er et eksempel på beregning af en cylinders rumfang:

// Beregning af rumfang for en cylinder

//

**public class** Cylinderberegning

{

**public static void** main(String[] args)

{

**double** radius;

radius = 5.0;

**double** højde = 12.5;

//beregn rumfang

**double** volumen = radius \* radius \* højde \* 3.14159;

System.out.println(”Cylinderens højde: ” + højde);

System.out.println(”Cylinderens radius: ” + radius);

System.out.println(”Cylinderens volumen: ” + volumen);

}

}

Cylinderens højde: 12.5

Cylinderens radius: 5.0

Cylinderens volumen: 981.7468749999999

Læg mærke til, at man godt kan erklære en variabel og tildele den værdi i samme linje:

**double** højde = 12.5;

er altså det samme som:

**double** højde;

højde = 12.5;

Udregninger sker normalt fra venstre mod højre, men ligesom i den almindelige matematik kan man påvirke udregningsrækkefølgen ved at sætte parenteser.

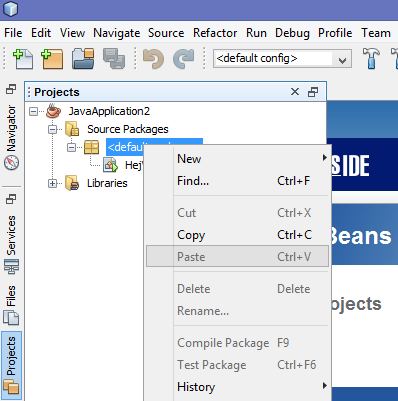
I stedet for at skrive pi med 6 betydende cifre, kan man hente værdien fra Math biblioteket ved at skrive:

Math.PI

### 2.5 Interaktivt program:

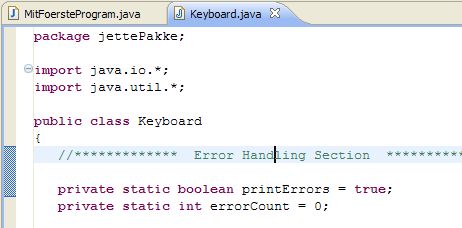
Det bliver meget sjovere at køre programmer hvis man kan indtaste inddata direkte fra tastaturet. Herunder beskrives to måder.

I Fronter-arkivet ligger en fil: Keyboard.java. Gem den på din harddisk.  
Kopier filen og sæt ind(paste eller crtl v) i Netbeans under pakkenavnet.



Vigtigt:

* Åben Keyboard.java i Netbeans og sørg for at pakke-navnet er helt rigtigt. (Eller kopier filen fra Stifinder og ”sæt ind”/”paste” direkte i Netbeans Editoren i/på SRC mappen, så ændrer Netbeans automatisk pakkenavnet til det rigtige.(Hvis ikke, må du åbne filen og rette pakkenavnet til korrekte, eller udkommentere ved at sætte // foran package, hvis default pakke benyttes.)
* Husk java skelner mellem store og små bogstaver.
* Ret navnet og gem fil, før du fortsætter!



Keyboard-klassen er nødvendig, for at vi kan modtage besked fra Keyboardet om, hvilken tast, der er blevet aktiveret.

Nu kan du åbne en ny klasse og skrive/kopiere dette ind:

// Beregning af rumfang for en cylinder

// med interaktiv indtastning af radius og højde

**package** jettePakke;

**public class** Cylinderberegning2

{ **public static void** main(String[] args)

{

**double** radius,hoejde;

System.out.print(”Indtast radius: ”);

**radius = Keyboard.readDouble();**

System.out.print(”\nIndtast hoejde: ”);

hoejde = Keyboard.readDouble();

//beregn rumfang

**double** volumen = radius \* radius \* hoejde \* 3.14159;

System.out.println(”Cylinderens hoejde: ” + hoejde);

System.out.println(”Cylinderens radius: ” + radius);

System.out.println(”Cylinderens volumen: ” + volumen);

}

}

Hvis du vil indlæse heltal bruges Keyboard.readInt()

***Eller du kan benytte (fra javabog.dk)***

### 2.3.1 Indlæsning fra tastaturet

Et program bliver selvfølgelig først rigtigt sjovt, hvis brugeren kan påvirke dets udførelse, f.eks. ved at programmet kan bede brugeren om at indtaste sin alder. Det kan gøres med:

public class AlderMedTastaturindlaesning

{

public static void main(String[] arg)

{

**java.util.Scanner** tastatur = new java.util.Scanner(System.in); *// forbered*

System.out.println("Skriv din alder herunder og tryk retur:");

int alder;

**alder = tastatur.nextInt();**  *// læs et tal fra tastaturet*

if (alder >= 18) System.out.println("Du er myndig.");

System.out.println("Du er " + alder + " år.");

}

}

Programmet vil nu stoppe op og vente på, at der bliver indtastet noget efterfulgt af tryk på retur-tasten. Den øverste linje (med kommentaren *forbered*) skal kun forekomme én gang øverst i programmet (i [kapitel 3](http://javabog.dk/OOP/kapitel3.jsp) vil vi se nærmere på, hvad der sker i de to specielle linjer).

Du kan også få et grafisk indtastningsvindue, som brugeren kan udfylde til at dukke op (se [afsnit 2.12.1](http://javabog.dk/OOP/kapitel2.jsp#afsn2.12.1)).

### 2.6 Datatyper

Du er nu blevet præsenteret for to simple datatyper i Java: integer og double.

Der findes 8 simple datatyper i Java.

Heltal: *byte, short, int, long* f.eks. –11, 0, 345, 12367

Decimaltal: *float, double* f.eks. 5.75, -1389.225

Skrifttegn: *char*  f.eks. A, N

Udsagn: *boolean* f.eks. (sum>100) , (tal == 3)   
 der enten er sandt eller falsk.

Der findes flere typer af heltal og decimaltal alt efter hvor meget plads de optager i hukommelsen og dermed hvor store tal de kan rumme. Herunder er vist en oversigt over hvilke værdier data­typerne kan repræsentere og hvor megen hukommelsesplads de optager:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Hukommelses-**  **­­­plads** | **Min. Værdi** | **Max. værdi** |
| Byte | 8 bits = 1 byte | -128 | 127 |
| short | 16 bits | -32.768 | 32.767 |
| Int | 32 bits | -2.147.486.648 | 2.147.486.647 |
| Long | 64 bits | < -9 \*1018 | > 9\*1018 |
|  |  |  | |
| Float | 32 bits | +/- 3.4\*1038  med 7 betydende cifre | |
| double | 64 bits | +/- 1.7\*10308 med 15 betydende cifre | |
|  |  |  |  |
| Char | 16 bits | 0 | 65.536 |
| boolean | 1 bit | 0 | 1 |

Følgende opgaver afleveres i afleveringsmappen på Fronter.

**Opgave:2.1.**Skriv et program der beregner volumenet af en kugle.

Du skal kun aflevere filen: *filnavn.java*

**Opgave 2.2.**Skriv et program der beregner arealet af et cirkeludsnit.  
Du skal kun aflevere filen: *filnavn.java*

God arbejdslyst

Just