

# Introduktion til Objekt-Orienteret Programmering

## Øvelsesbog – Modul 3

Aslak Johansen <asjo@mmmi.sdu.dk>  
Anders Clausen <ancla@mmmi.sdu.dk>

September 10, 2021

## Contents

<b>I</b>	<b>Øvelser</b>	<b>4</b>
<b>1</b>	<b>Typer</b>	<b>4</b>
1.1	Hello, World . . . . .	4
1.2	Temperatur . . . . .	4
1.3	Måned . . . . .	4
1.4	Heltallige Grænser . . . . .	4
1.5	Casting . . . . .	4
<b>2</b>	<b>Expressions</b>	<b>5</b>
2.1	Definition . . . . .	5
2.2	Tildeling . . . . .	5
2.3	Expression vs Statement . . . . .	5
<b>3</b>	<b>Variable</b>	<b>5</b>
3.1	Areal af Cirkler . . . . .	5
3.2	Sum af Areal af Cirkler . . . . .	5
3.3	Celcius til Fahrenheit . . . . .	5
3.4	Epoch . . . . .	6
3.5	Inkrementering af Måned . . . . .	6
3.6	Værdi vs Variabel . . . . .	6

3.7	Daglige Differencer . . . . .	6
3.8	Gennemsnitlig Alder . . . . .	7
3.9	Printf . . . . .	8
<b>4</b>	<b>Booleans</b>	<b>8</b>
4.1	Definition . . . . .	8
4.2	Oprettelse . . . . .	8
4.3	Købsbeslutning . . . . .	8
4.4	Terninger . . . . .	8
<b>5</b>	<b>Branches</b>	<b>9</b>
5.1	Epoch . . . . .	9
5.2	Epoch Diff . . . . .	9
5.3	Juleudsalg . . . . .	9
5.4	Ferie . . . . .	10
<b>6</b>	<b>Loops</b>	<b>10</b>
6.1	Celcius til Fahrenheit . . . . .	10
6.2	Celcius til Fahrenheit i Omvendt Rækkefølge . . . . .	10
6.3	Celcius til Fahrenheit Alternativer . . . . .	10
6.4	Areal af Cirkler . . . . .	10
6.5	Længden af en Måned . . . . .	11
6.6	Printal . . . . .	11
<b>7</b>	<b>Arrays</b>	<b>12</b>
7.1	Definition . . . . .	12
7.2	Anvendelse . . . . .	12
7.3	Type af Indhold vs Type af Array . . . . .	12
7.4	Størst i Array . . . . .	12
7.5	Deklaration af Størrelse . . . . .	12
7.6	Multiplikationstabel . . . . .	12
7.7	Sudoku Plade . . . . .	12
7.8	Areal af Cirkler . . . . .	13
7.9	Daglige Differencer . . . . .	13
7.10	Længden af en Måned . . . . .	13

7.11	Primtal . . . . .	13
7.12	Kalender . . . . .	13
7.13	Kalender Prettyprinting . . . . .	14
7.14	Sudoku Checker . . . . .	14
<b>II</b>	<b>Vejledende Løsninger</b>	<b>14</b>
<b>8</b>	<b>Typer</b>	<b>15</b>
8.1	Hello, World . . . . .	15
8.2	Måned . . . . .	15
8.3	Heltallige Grænser . . . . .	15
<b>9</b>	<b>Expressions</b>	<b>16</b>
9.1	Tildeling . . . . .	16
9.2	Expression vs Statement . . . . .	16
<b>10</b>	<b>Variable</b>	<b>16</b>
10.1	Areal af Cirkler . . . . .	16
10.2	Inkrementering af Måned . . . . .	17
10.3	Værdi vs Variabel . . . . .	17
10.4	Daglige Differencer . . . . .	17
10.5	Gennemsnitlig Alder . . . . .	18
10.6	Printf . . . . .	18
<b>11</b>	<b>Booleans</b>	<b>19</b>
11.1	Definition . . . . .	19
11.2	Oprettelse . . . . .	19
11.3	Købsbeslutning . . . . .	19
<b>12</b>	<b>Branches</b>	<b>20</b>
12.1	Juleudsalg . . . . .	20
12.2	Ferie . . . . .	20
<b>13</b>	<b>Loops</b>	<b>22</b>
13.1	Areal af Cirkler . . . . .	22
13.2	Længden af en Måned . . . . .	22

13.3 Primaltal . . . . .	23
--------------------------	----

## Part I

# Øvelser

## 1 Typer

### 1.1 Hello, World

Skriv et program, der udskriver teksten “Hello, World”.

### 1.2 Temperatur

Hvilke datatyper er velegnede til at representere en temperatur?

### 1.3 Måned

Hvilke typer er velegnede til at repræsentere en måned?

### 1.4 Heltallige Grænser

1. Hvilke heltallige datatyper kan man benytte sig af i Java?
2. Vælg én af dem.
3. Skriv et program der eksperimentelt afslører hvad der sker når man overskrider den størst mulige værdi.
4. Beskriv hvad det er I observerer?

### 1.5 Casting

Når vi godt vil konvertere mellem `int` værdier og `long` værdier skal der udføres et cast. Men, hvornår er det at dette cast skal gøres eksplicit, og hvornår må det være implicit?

- Hvorved adskilker `int` og `long` sig?
- Prøv at skrive et program der (i) erklærer `i` som en `int` variabel, (ii) tildeler den en værdi, (iii) erklærer `l` som en `long` variabel, (iv) tildeler værdien af `i` til `l`, og (v) slutteligt tildeler værdien af `l` til `i`.
- Eksperimentér med hvor det er nødvendigt at have eksplicite casts.

- Gør nu det samme med en `float` variabel `f` og en `double` variabel `d`.
- Eksperimentér igen med hvor det er nødvendigt at have eksplicitte casts.
- Afhænger resultatet af disse eksperimenter af den værdi som I initielt tildeler den første variabel?

## 2 Expressions

### 2.1 Definition

Hvad er et expression?

### 2.2 Tildeling

Er en tildeling (af en værdi til en variabel) et expression?

### 2.3 Expression vs Statement

Hvad er forskellen på et expression og et statement?

## 3 Variable

### 3.1 Areal af Cirkler

Skriv et program der udregner og udskriver arealet ( $\pi \cdot r^2$ ) af tre cirkler med radius på hhv. 1, 3 og 5.

### 3.2 Sum af Areal af Cirkler

Skriv et program der udregner og udskriver omkredsen ( $2 \cdot \pi \cdot r$ ) af tre cirkler med radius på hhv. 1, 3 og 5, og afslutter med at udskrive summen af disse.

### 3.3 Celcius til Fahrenheit

Skriv et program, hvori

1. En temperatur angives i Celcius via en variabel.
2. Denne temperatur konverteres til Fahrenheit og gemmes i en anden variabel.
  - Formel:  $T_F = 32 + \frac{9}{5}T_C$
3. Konverteringen udskrives på en passende måde.

### 3.4 Epoch

Skriv et program hvori

1. Et antal sekunder siden et bestemt tidspunkt (fx 1. Januar 1970) gemmes i en variabel.
2. Konvertér dette tal til et helt antal år (lad os antage at der er 365 dage på et år) og et antal hele dage indenfor det sidste år. Lægges disse to tal sammen skal resultatet altså være indenfor 24 timer af udgangspunktet.
3. Udskriv disse to tal.

Verificér at programmet virker.

### 3.5 Inkrementering af Måned

Skriv et program, hvori

1. En heltallig variabel bruges til at repræsentere en måned.
2. Denne variabel tildeles en værdi (du vælger selv).
3. Udskriv variabelens værdi.
4. Forøg værdien af denne variabel med en halv.
5. Udskriv variabelens værdi.
6. Forøg værdien af denne variabel med en halv.
7. Udskriv variabelens værdi.

Få dette program til at oversætte, kød det og beskriv hvad du observerer.

Forklar hvorfor programmet opfører sig sådan.

### 3.6 Værdi vs Variabel

Hvad er forholdet mellem en værdi og en variabel?

### 3.7 Daglige Differencer

Skriv et program, der givet 7 dagstemperaturer udregner og udskriver temperaturdifferencen mellem alle to på hinanden følgende dage (dvs. Tirsdag-Mandag, Onsdag-Tirsdag ... Søndag-Lørdag).

Dagstemperaturerne kunne være:

- Mandag: 21.5

- Tirsdag: 23.7
- Onsdag: 19.6
- Torsdag: 22.5
- Fredag: 25.3
- Lørdag: 21.7
- Søndag: 18.9

### 3.8 Gennemsnitlig Alder

Betragt følgende program:

```
public class AvgAge
{
    public static void main (String[] args) {
        int ada_lovelace      = 36; // https://en.wikipedia.org/wiki/Ada\_Lovelace
        int dennis_ritchie    = 70; // https://en.wikipedia.org/wiki/Dennis\_Ritchie
        int grace_hopper      = 85; // https://en.wikipedia.org/wiki/Grace\_Hopper
        int hedy_lamarr        = 85; // https://en.wikipedia.org/wiki/Hedy\_Lamarr
        int edsger_dijkstra    = 72; // https://en.wikipedia.org/wiki/Edsger\_W.\_Dijkstra
        int douglas_engelbart = 88; // https://en.wikipedia.org/wiki/Douglas\_Engelbart

        float male_avg  = (float)(dennis_ritchie+edsger_dijkstra+douglas_engelbart)/3;
        float female_avg = (float)(ada_lovelace+grace_hopper+hedy_lamarr)/3;
        float avg = (male_avg+female_avg)/2;
        float diff = male_avg-female_avg;

        System.out.print("Average lifespan of a male computer scientist: ");
        System.out.println(male_avg);
        System.out.print("Average lifespan of a female computer scientist: ");
        System.out.println(female_avg);
        System.out.print("Average lifespan of a computer scientist: ");
        System.out.println(avg);
        System.out.print("Males lives this much longer than female: ");
        System.out.println(diff);
    }
}
```

Udfør programmet. Hvad sker der?

Skriv nu en tekst på dansk hvor I ved hjælp af fagtermer forklarer hvad der sker. Sørg for at denne tekst er grundig nok til at en programmør kan genkonstruere ovenstående kode.

### 3.9 Printf

Undersøg hvordan følgende stykke kode fungerer ved at modificere indholdet af strengen i den sidste linje:

```
int i = 42;
long l = 56;
float f = 3.14159;
double d = 3.14159*10;
System.out.printf("i=%d l=%,4d f=%f d=%6.2f", i, l, f, d);
```

## 4 Booleans

### 4.1 Definition

Hvad er en boolean?

### 4.2 Oprettelse

Hvilke operatorer kan man bruge til at få et *expression* til at evaluere til en boolsk værdi?

### 4.3 Købsbeslutning

Betragt følgende kodeudtræk:

```
double price = 599.95;
double budget = 1000.0;
boolean requiredReading = true;
boolean shouldBuy = price < budget && requiredReading;
```

Forklar sidste linje og fokuser på:

- I hvilken rækkefølge bliver hvad udregnet?
- Hvilke værdier (navngivne eller ej) udføres de enkelte operatorer på?
- Hvilke typer har disse værdier?
- Hvad repræsenterer variablen `shouldBuy`?

### 4.4 Terninger

Skriv et program, hvori

1. Værdien af et terningslag er gemt i en variabel ved navn `dice`.



- Hvilken type giver det mening at erklære variabelen som?
  - Vælg selv en specifik værdi.
2. Opret en boolsk variabel og tildel den en værdi der repræsenterer hvorvidt værdien fra variabelen `dice` er lige og større end 3.
  3. Udskriv den værdien af denne boolske variabel.

## 5 Branches

### 5.1 Epoch

Skriv et program, hvori

1. I tager udgangspunkt i opgaven fra afsnit 3.4.
2. I erklærer en variabel hvis værdi repræsenterer et antal sekunder siden nytår.
3. På baggrund af værdien af denne variabel udregner I hvilken måned og hvilken dag der er tale om (I kan gå ud fra at alle måneder er 30 dage lange).
4. Udskriv "Det er jul!" hvis det er tilfældet (det er jul den 24. December).

### 5.2 Epoch Diff

Udvid opgaven fra afsnit 5.1 til – hvis man skal vente – at udskrive hvor lang tid man skal vente på at det er jul.

### 5.3 Juleudsalg

Skriv et program, hvori

1. En variabel oprettes (deklarerer) og initialiseres til værdien 21816000. Dette tal repræsenterer et antal sekunder siden nytår (alle måneder antages at være 30 dage lange).
2. En anden variabel indeholder en pris på 599,95 dkr.
3. Der skal gives et 30% tilskud hvis det er Jul. Find selv på en fornuftig definition af hvornår det er Jul.
4. Udregn den gældende pris (eventuelt tilskud medregnet) og gem denne i en variabel.
5. Udskriv denne variabel.
6. Sørg for at teste den logik I har skrevet ved at prøve at tildele den første variable forskellige andre værdier. Hvilke værdier vil være fornuftige at teste?

## 5.4 Ferie

Undervisningskalenderen fortæller os at der (blandt andet) er følgende ferier:

- *Efterårsferie* Oktober
- *Juleferie* December
- *Påskeferie* April
- *Sommerferie* Juli + August

Skriv et program, hvori

1. Et månedsnummer gives via en variabel.
2. Afhængigt af indholdet af denne variabel udskrives en feries navn (hvis der er ferie i måneden) eller "Hårdt arbejde" (hvis der ikke er)

## 6 Loops

### 6.1 Celcius til Fahrenheit

Skriv et program, hvori

- Der udskrives en tabel af matchende Celcius og Fahrenheit værdier.
  - Formel:  $T_F = 32 + \frac{9}{5}T_C$
- Der skal være ét sæt matchende værdier per linje.
- Listen skal starte med -5°C og slutte ved 40°C.
- Listen skal have én linje for hver 0,5°C.

### 6.2 Celcius til Fahrenheit i Omvendt Rækkefølge

Omskriv programmet fra opgave 6.1 til at vende rækkefølgen om sådan at første linje udskriver 40°C og sidste -5°C.

### 6.3 Celcius til Fahrenheit Alternativer

Lav to andre udgaver af programmet fra opgave 6.1, hvori loopet omskrives til hver af de to resterende typer af loops.

### 6.4 Areal af Cirkler

Skriv et program der udregner og udskriver arealet ( $\pi \cdot r^2$ ) af tre cirkler med radius på hhv. 1, 3 og 5.

## 6.5 Længden af en Måned

Skriv et program, hvori

1. En måneds nummer gemmes i en variabel. Denne variabel skal fungere som input til jeres program.
  - Hvilket navn ville være passende for denne variabel?
  - Hvilken type ville være passende for denne variabel?
2. Skriv noget kode der på baggrund af denne variabel bestemmer hvor mange dage der er i denne måned (vi antager at det ikke er skudår).
3. Skriv dette tal ud på skærmen.
4. Overbevis jer selv om at jeres kode er korrekt.

## 6.6 Primaltal

Skriv et program, der udregner alle primaltal under 1.000.000, og udskriver det største.

Hints (se bort fra dette hvis I er friske på en udfordring):

- Benyt en del-og-hersk strategi, hvor I ser opgaven som tre delopgaver:
  - Gennemløb alle positive heltal under 1.000.000. Har vi ikke en konstruktion der kan det?
  - Bestem om et givent positivt heltal er et primaltal.
  - Udprint et heltal (hvis det altså er et primaltal).
- Et positivt heltal er et primaltal hvis og kun hvis der ikke er andre heltal end 1 der går op i det.
- For at bestemme om et givent positivt heltal er et primaltal kan I endnu engang benytte del-og-hersk:
  1. Opret en variable af typen `boolean` ved navn `is_prime` som er `True`.
  2. Gennemløb alle heltal fra og med 2 til (men ikke med) 1.000.000.
  3. For hvert af disse tal skal I checke om dette tal går op i det potentielle primaltal. Gør de det, så sættes `is_prime` til `False`. Men hvordan undersøger man om ét tal går op i et andet?
    - Man prøver det da!
    - Hvis en heltalsdivision går op er det ikke nogen rest og en modulo operation (via operatoren `%`) giver nul.
    - Alternativt kan man udnytte at en heltalsdivision kommer til at foretage en afrunding hvis divisionen ikke går op sådan at  $(a/b) \cdot b \neq a$ .
  4. Herefter repræsenterer `is_prime` en sandhedsværdi for hvovidt et tal er et primaltal.

## 7 Arrays

### 7.1 Definition

Hvad er et array?

### 7.2 Anvendelse

Hvornår giver det mening at bruge arrays?

### 7.3 Type af Indhold vs Type af Array

Hvad er sammenhængen mellem typen af et array og de data som det kan indeholde?

### 7.4 Størst i Array

Skriv et program, der finder det største tal i et array af typen `int[]`, og udskriver indeks for dette tal. Find selv på noget passende indhold til dette array.

### 7.5 Deklaration af Størrelse

Hvilket syntaktiske element bruges – ved oprettelse af et array – til at erklære hvor mange elementer der skal være plads til?

### 7.6 Multiplikationstabel

Skriv et program, hvori

1. En heltallig variabel ved navn `size` initialiseres til en værdi under 30. I vælger selv den konkrete værdi.
2. Der oprettes et array med en længde som modsvarer værdien af `size`.
3. Arrayet fyldes op med 3-tabellen. Altså, elementet med indeks  $n$  skal have værdien  $3 \cdot n$ .
4. Et (eller flere) velvalgt(e) element(er) udskrives for at verificere korrektheden.

### 7.7 Sudoku Plade

Hvordan ville du repræsentere en sudoku<sup>1</sup> plade i Java?

Hvordan hænger denne datastruktur sammen i hukommelsen?

---

<sup>1</sup><https://en.wikipedia.org/wiki/Sudoku>

## 7.8 Areal af Cirkler

Skriv et program der udregner og udskriver arealet ( $\pi \cdot r^2$ ) af tre forskellige cirkler med radius 1, 3, og 5.

## 7.9 Daglige Differencer

Skriv et program, der givet 7 dagstemperaturer udregner og udskriver temperaturdifferencen mellem alle to på hinanden følgende dage (dvs. Tirsdag-Mandag, Onsdag-Tirsdag ... Søndag-Lørdag).

Dagstemperaturerne kunne være:

- Mandag: 21.5
- Tirsdag: 23.7
- Onsdag: 19.6
- Torsdag: 22.5
- Fredag: 25.3
- Lørdag: 21.7
- Søndag: 18.9

## 7.10 Længden af en Måned

Skriv et program, der givet en måneds nummer udskriver antallet af dage i denne måned (se bort fra skudår).

## 7.11 Primtal

Skriv et program der udregner alle primtal under 1.000.000 og udskriver det største.

Gør dette ved at implementere Eratosthenes Si<sup>2</sup>.

Kvadratroden af *i* udregnes som `java.lang.Math.sqrt(i)`.

## 7.12 Kalender

Skriv et program, hvori

1. En variabel initialiseres til at være et array der indeholder antallet af dage i hver af de 12 måneder i et normalt år. Det første element vil da indeholde antallet af dage i Januar.

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Sieve\\_of\\_Eratosthenes](https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes)

2. En anden variabel initialiseres til at være et array der indeholder antallet af dage i hver af de 12 måneder i et skudår.
3. Gennemløb årene 2000 til 2020.
4. Brug en ny variabel af samme type som de to forregående til at holde styr på hvilket array der er korrekt for det aktuelle år.
  - **Hint:** Vi kan i denne opgave tillade os at forsimple skudårsreglerne til at det er skudår hvis 4 går op i årstallet.
5. For hvert år udskrives indholdet af det array som den sidste variabel peger på.

### 7.13 Kalender Prettyprinting

Lav et program, hvori

1. Der oprettes en datastruktur til at holde en kalender der spænder ét år. Man skal kunne indeksere ind med en dato (måned + dag) og få en dag (fx Mandag).
  - Hvad er typen af denne datastruktur?
  - Hvordan initialiseres en variabel af denne type?
  - Hvordan kan man sørge for at indholdet er korrekt?
2. Udskriv denne datastruktur på en “pæn” måde.

### 7.14 Sudoku Checker

Skriv et program, hvori

- En variabel initialiseres til at indeholde en udfyldt sudokuplade.
- Skriv kode der vurderer om pladen repræsenterer en korrekt løsning.
  - **Hint:** Dette er tilfældet når (i) alle felter er udfyldte, og (ii) ingen af de tre regler er blevet brudt.
- Udskriv resultatet af denne vurdering.

## Part II

# Vejledende Løsninger

## 8 Typer

### 8.1 Hello, World

```
public class Hello
{
    public static void main (String[] args) {
        System.out.println("Hello, World");
    }
}
```

### 8.2 Måned

Der er 12 forskellige måneder, hvor man kunne sige at 1 repræsenterer Januar, 2 Februar ... Det er ikke klart hvad 1.1 bør betyde, så det giver mening at begrænse sig til heltal. Derfor vil vi naturligt bruge en heltallig type, og da vi vi kun har 12 måneder er `int` mere pladsøkonomisk end `long`.

### 8.3 Heltallige Grænser

I Java er der 4 (primitive) heltalstyper: `byte`, `short`, `int` og `long`.

En `byte` kan repræsenterer alle heltal fra og med -128 til og med 127. For at finde ud af hvad der sker når vi overskrider det største heltal der kan repræsenteres, skriver vi følgende program:

```
public class IntegerBoundary
{
    public static void main (String[] args) {
        byte b = 127;
        System.out.println(b);
        b += 1;
        System.out.println(b);
    }
}
```

Kører vi dette program får vi:

```
127
-128
```

Vi har altså, at  $127 + 1$  giver os  $-128$ . Eller med andre ord; når vi lægger én til det størst mulige tal så får vi det mindst mulige tal. Dette er ikke et tilfælde, og gør sig generelt gældende for alle de 4 ovenstående typer.

## 9 Expressions

### 9.1 Tildeling

Du kan jo prøve:

```
public class ExprTest
{
    public static void main (String[] args) {
        int var1;
        int var2 = var1 = 42;
        System.out.println(var1);
        System.out.println(var2);
    }
}
```

### 9.2 Expression vs Statement

Expressions og statements er forskellige syntaktiske konstruktioner. Det betyder at Javas oversætter (nydansk: compiler) forventer expressions nogen steder og statements andre steder. Overholder ens program ikke disse *syntaktiske* regler vil oversætteren give en såkaldt syntaksfejl og opgive.

Den vigtigste forskel er at et expression – i modsætning til statements – kan evalueres til en værdi. Hovedstrukturen af et stykke logik er udgøres derfor at statements og expressions bruges til at udregne mellemresultater.

## 10 Variable

### 10.1 Areal af Cirkler

```
public class CircleArea
{
    public static void main (String[] args) {
        final double pi = 3.14;
        double r;

        r = 1;
        System.out.println("r="+r+" -> area="+ (pi*r*r));

        r = 3;
        System.out.println("r="+r+" -> area="+ (pi*r*r));
    }
}
```



```

        r = 5;
        System.out.println("r="+r+" -> area="+ (pi*r*r));
    }
}

```

## 10.2 Inkrementering af Måned

**Observation:** Da vi er i den niende måned gav jeg 9 som initiel værdi. Denne værdi ændrede sig ikke da jeg lagde en halv til, og heller ikke da jeg gjorde dette igen.

**Forklaring:** Ved addition af forskellige typer foretager Java en implicit konvertering til den første type. Denne type er heltallige og værdien 0.5 bliver derfor rundet ned til 0. Derfor lægges der i praksis nul til to gange.

## 10.3 Værdi vs Variabel

En variabel er en symbolsk reference (eller indirektion) til en værdi. Vi siger at variabelen *peger* på værdien. Vi kan ændre på hvilken værdi en variabel peger på ved at tildele den en ny værdi. I praksis ligger værdien et sted i hukommelsen og variabelen indeholder adressen på dette sted.

Dette er en *abstraktion* der gør det muligt at skrive kode der er uafhængig af specifikke værdier.

## 10.4 Daglige Differencer

```

public class DailyDiffs
{
    public static void main (String[] args) {
        final double day1 = 21.5;
        final double day2 = 23.7;
        final double day3 = 19.6;
        final double day4 = 22.5;
        final double day5 = 25.3;
        final double day6 = 21.7;
        final double day7 = 18.9;

        System.out.println(day2-day1);
        System.out.println(day3-day2);
        System.out.println(day4-day3);
        System.out.println(day5-day4);
        System.out.println(day6-day5);
        System.out.println(day7-day6);
    }
}

```

## 10.5 Gennemsnitlig Alder

Når programmet udføres printes følgende ud:

```
Average lifespan of a male computer scientist: 76.666664
Average lifespan of a female computer scientist: 68.666664
Average lifespan of a computer scientist: 72.666664
Males lives this much longer than female: 8.0
```

Programmet har en main metode der består af (i) 6 variabel erklæringer med initiale værdier, (ii) en række variabel erklæringer der initialiseres til at holde resultatet af nogle udregninger, og til sidst udskrives resultaterne.

I den første del erklæres 6 `int` variabler. Hver af disse er navngivet efter en kendt datalog og initialiseres til hvor gamle de blev. For hver af disse er der et udkommenteret link til deres side på Wikipedia. Dernæst kommer der fire udregninger. Hver udregning gemmes i en `float` variabel.

I den første udregning lægges samtlige mandlige variable sammen. Resultatet af denne opsummering castes til en `float` hvorefter det divideres med 3 (antallet) og gemmes i `male_avg` variablen. Tilsvarende udregnes og gemmes gennemsnittet for kvinder i `female_avg`. I den tredje udregning lægges disse to gennemsnit sammen og resultatet af dette divideres med 2 og gemmes i variablen `avg`. Dette er korrekt eftersom der er lige mange mænd og kvinder, og da `male_avg` er af typen `float` er det ikke nødvendigt at caste for at undgå afrundingsfejl i divisionen. I den sidste udregning trækkes `female_avg` fra `male_avg` og resultatet gemmes i variablen `diff`.

Til sidst printes hver af disse 4 udregnede variable ud på skærmen. For hver af dem bruges to statements. I det første statement udskrives en streng der beskriver hvad værdien betyder uden linjebrud. I det andet og sidste statement udskrives værdien med linjebrud.

## 10.6 Printf

Den fulde test kode:

```
public class Formatting
{
    public static void main (String[] args) {
        int i = 42;
        long l = 56;
        float f = 3.14159F;
        double d = 3.14159*10;
        System.out.printf("i=%d l=%,4d f=%f d=%6.2f", i, l, f, d);
        System.out.println("");
    }
}
```

Vi kan bruge `System.out.printf` til at *padde* værdier med mellemrum og

bestemme hvor mange decimaler vi ønsker efter kommaet. Herved kan vi formatere udskrifter.

## 11 Booleans

### 11.1 Definition

En `boolean` er både en type og en variabel af denne type. Det betyder at variabelen have (pege på) én af to værdier; `true` (sandt) og `false` (falsk). Dette er en enkelt bit information, men tildeles typisk mere plads i hukommelsen. En række særlige (boolske) operationer er defineret på denne type. Disse kan bruges til at implementere Boolsk logik.

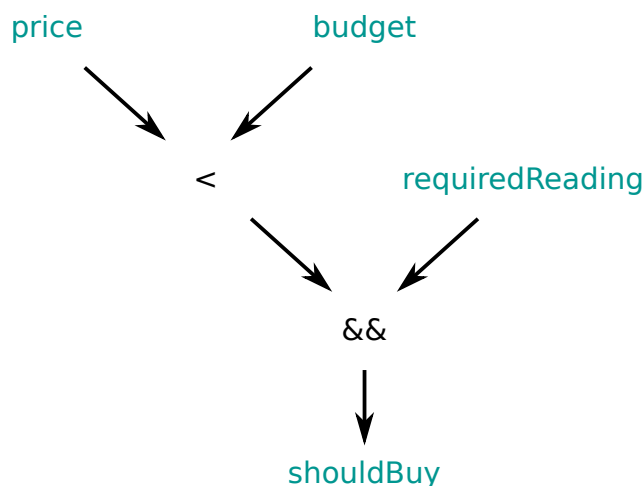
### 11.2 Oprettelse

Boolske værdier bliver produceret når:

1. Man i koden skriver `true` eller `false`.
2. Som resultatet af en sammenligning.
3. Som resultatet af en boolsk operation.

### 11.3 Købsbeslutning

Udregningen foregår således:



Variabelen `shouldBuy` repræsenterer en beslutning for hvorvidt en bog skal købes. Dette sker når prisen er indenfor budgettet og bogen er en del af læsepensum.

## 12 Branches

### 12.1 Juleudsalg

```
public class ChristmasSale
{
    public static void main (String[] args) {
        final int secsPerDay = 24*60*60;
        final int secsPerYear = 360*secsPerDay;
        final int xmas = (11*30+23)*24*60*60;

        //      long epoch = xmas-secsPerDay;
        //      long epoch = xmas+secsPerDay;
        long epoch = xmas;

        long years = epoch/secsPerYear;
        long days = (epoch-years*secsPerYear)/secsPerDay;
        long month = days/30;
        long day = days%30;

        double price = 599.95 * (month==11 && day==23 ? 0.7 : 1.0);
        System.out.println(price);
    }
}
```

### 12.2 Ferie

En typisk implementering:

```
public class Holiday
{
    public static void main (String[] args) {
        final int JANURAY = 0;
        final int FEBRUARY = 1;
        final int March = 2;
        final int APRIL = 3;
        final int MAY = 4;
        final int JUNE = 5;
        final int JULY = 6;
        final int AUGUST = 7;
        final int SEPTEMER = 8;
        final int OCTOBER = 9;
        final int NOVEMBER = 10;
        final int DECEMBER = 11;

        // choose a test case
        long month = OCTOBER;
        //      long month = NOVEMBER;
    }
}
```

```

switch ((int)month) {
case OCTOBER:
    System.out.println("Efterårsferie");
    break;
case DECEMBER:
    System.out.println("Juleferie");
    break;
case APRIL:
    System.out.println("Påskeferie");
    break;
case JULY:
case AUGUST:
    System.out.println("Sommerferie");
    break;
default:
    System.out.println("Hårdt arbejde");
    break;
}
}
}

```

Karl Koders version:

```

public class KarlsHoliday
{
    public static void main (String[] args) {
        final int JANURAY = 0;
        final int FEBRUARY = 1;
        final int March = 2;
        final int APRIL = 3;
        final int MAY = 4;
        final int JUNE = 5;
        final int JULY = 6;
        final int AUGUST = 7;
        final int SEPTEMER = 8;
        final int OCTOBER = 9;
        final int NOVEMBER = 10;
        final int DECEMBER = 11;

        // choose a test case
        long month = OCTOBER;
        //      long month = NOVEMBER;

        String message = (month==OCTOBER ? "Efterårsferie"
                           : (month==DECEMBER ? "Juleferie"
                              : (month==APRIL ? "Påskeferie"
                                 : (month==JULY || month==AUGUST ? "Sommerferie"
                                    : "Hårdt arbejde"))));
        System.out.println(message);
    }
}

```

```
    }
}
```

## 13 Loops

### 13.1 Areal af Cirkler

```
public class CircleArea
{
    public static void main (String[] args) {
        final double pi = 3.14;

        for (double r=1 ; r<=5 ; r+=2) {
            System.out.print("r=");
            System.out.print(r);
            System.out.print(" -> area=");
            System.out.println(pi*r*r);
        }
    }
}
```

### 13.2 Længden af en Måned

```
public class Months
{
    public static void main (String[] args) {
        final int month = 6;
        int length = -1;

        if (month==1 || month==3 || month==5 || month==7 || month==8 ||
            month==10 || month==12) {
            length = 31;
        } else if (month==2) {
            length = 28;
        } else if (month==4 || month==6 || month==9 || month==11) {
            length = 30;
        }

        if (length==-1) {
            System.out.println("Fejl: Månedens \"+month+\"er ikke indenfor [1,12]");
        } else {
            System.out.println(length);
        }
    }
}
```

Karl Koder anbefaler dog:

```

public class KarlsMonths
{
    public static void main (String[] args) {
        final int month = 6;
        int length = -1;

        length = (month==1 || month==3 || month==5 || month==7 || month==8 ||
            month==10 || month==12 ? 31 : 0)
            + (month==2 ? 28 : 0)
            + (month==4 || month==6 || month==9 || month==11 ? 30 : 0);

        if (length==-1) {
            System.out.println("Fejl: Månedens \""+month+"\"er ikke indenfor [1,12]");
        } else {
            System.out.println(length);
        }
    }
}

```

### 13.3 Primal

```

public class Primes
{
    public static void main (String[] args) {
        final int last = 1000000;
        int max = -1;

        for (int i=0 ; i<last ; i++) {
            boolean isPrime = true;

            // even numbers over 2 are not prime
            if (i>2 && (i&1)==0) isPrime = false;

            // primes have a remainder when divided by a lower odd number
            for (int j=3 ; j<i ; j++) {
                if (i%j==0) {
                    isPrime = false;
                    break;
                }
            }

            if (isPrime) {
                // System.out.println(i);
                max = i;
            }
        }

        System.out.println("Highest prime is "+max);
    }
}

```

}