

# Variabler, input og beslutninger

Hvordan få et program til å gjøre noe

INF1000, uke1  
Geir Kjetil Sandve

# Programmert fra før?

- Faget forutsetter ingen programmeringserfaring, og opplegget er tilpasset deretter
  - Vi vet samtidig at noen har programmert før studiet
- Dersom de første to-tre ukene føles kjent:
  - Reflekter! Er måten det blir presentert på i tråd med din oppfatning? Ville du forklart det annerledes?
  - Refleksjon er utrolig lærerikt i seg selv
  - Vi er også takknemlige for alle konstruktive innspill: Dropp gjerne innom "forum for fagutvikling" - torsdager kl 10-11, rom 1251 Scheme (t.o.m. 18.sep)

# Hva er en “Computer”?

- "The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations which could be done by a human computer" (Alan Turing, 1950)
- Tradisjonell definisjon av “computer”:  
En ansatt som gjør utregninger ved å blindt følge en liste instruksjoner med penn og papir, så lenge som det trengs,  
uten at det krever noen form for innsikt.  
(fra 1600-tallet og frem til andre verdenskrig)

# Nå skal dere få være computere!

- Har noen lagt merke til blyantene på bakerste rad? Det er også post-it lapper under pultene..
- Regn ut summen av to nederste tall, skriv denne summen under de andre tallene, og send lappen (og blyant) til personen foran deg
- Alle som mottar en lapp (fra personen bak) gjør det samme som beskrevet over
- Jeg samler inn lappene fra fremste rad

# Dere har nå regnet ut Fibonaccis tallrekke!

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
- Merk at resultatet av denne prosedyren er mer sofistikert enn operasjonen i hvert steg (plusse to tall)

# Vi har programmert!

- Eller rettere sagt:  
Jeg har programmert dere
- Oppfordring:  
Forsøk alltid å først løse problemet for hånd,  
ut fra presis sekvens av enkle operasjoner

# Hva er programming?

- *"Software development happens in your head, not in an editor" (Andy Hunt)*
- *"Programming is all about problem solving. It requires creativity, ingenuity, and invention"*
- *".. combining rich, flexible human thought with the rigid constraints of a digital computer exposes the power and the deepest flaws of both"*

# Syntaks og fokus

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

- Er alt dette egentlig kritisk informasjon for vårt formål?
- Kunne det ha vært en kortere måte å si hallo på, eller må man være så omstendelig når man har å gjøre med en datamaskin?



# Syntaks og fokus

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

- Dette er egentlig den kritiske informasjonen!  
(på dette stadiet)

# Syntaks og fokus (forts)

- Komplet program i Java:

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

- Tilsvarende (komplett) program i Python  
(et annet programmeringspråk):

```
print "Hallo Verden"
```

# Syntaks og fokus: konklusjon

- Unødvendig kompleks i Java
  - .. men likevel uproblematisk så lenge du ikke selv gjør det til et problem!
- Anbefalt strategi (nå foreløpig):
  - Legg inn rammeteksten blindt, og forsøk å ha 100% av fokuset på det meningsbærende i midten.
- Generelt tips:
  - Bruk copy-paste når du ønsker å IKKE lære noe
  - (og dermed også motsatt: unngå for all del copy-paste på alt du skal lære i faget!)

# Mål for dagen

- Hvordan få noe til å skje!  
(med programmering)
  - Utregninger og variabler
  - Innlesning fra tastatur
  - Beslutninger i programmer

Variabler

# Formålet med variabler

- {Kodefiler `u1var1.java` .. `u1var5.java`}:  
kode som introduserer uttrykk og variable
- I tillegg til å skrive “Hallo verden” kan du nå:
  - Regne ut uttrykk
  - Assosiere en verdi med et navn, og så referere til dette navnet videre

# Hva er nå egentlig en variabel

- Ikke det samme som x og y på ungdomsskolen: dette er mye lettere!
  - IKKE:  $x^2 - 3x = 28$
  - MEN:  $x=7$
  - ELLER:  $x=7*7 - 3*7$  (også helt rett frem)
- Navn som representerer verdi, eller boks som man putter en verdi inn i..

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = lengde * bredde;
```



# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = 7 * bredde;
```

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = 7 * 4 ;
```

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = 28 ;
```

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = 28 ;  
System.out.println("Rektangelet har  
storrelse: " + areal) ;
```

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
area1 = 28 ;  
System.out.println("Rektangelet har  
storrelse: " + 28 ) ;
```

# Et lite og intuitivt program

```
lengde = 7;  
bredde = 4;  
areal = lengde * bredde;  
System.out.println("Rektangelet har  
storrelse: " + areal) ;
```

# Programmet med variabel-initialisering

```
int lengde;  
int bredde;  
int areal;
```

```
lengde = 7;  
bredde = 4;  
areal = lengde * bredde;  
System.out.println("Rektangelet har  
storrelse: " + areal) ;
```

# Programmet i komplett versjon

```
public class u1Area1 {  
    public static void main(String[] args) {  
        int lengde;  
        int bredde;  
        int area1;  
  
        lengde = 7;  
        bredde = 4;  
        area1 = lengde * bredde;  
        System.out.println("Rektangelet har storrelse:  
" + area1) ;  
    }  
}
```



# Programmet, slik man bør fokusere

```
public class u1Area1 {  
    public static void main(String[] args) {  
        int lengde;  
        int bredde;  
        int area1;  
  
        lengde = 7;  
        bredde = 4;  
        area1 = lengde * bredde;  
        System.out.println("Rektangelet har storrelse:  
" + area1) ;  
    }  
}
```

# Variabel-typer

- “**int** lengde” definerte ikke bare at det var en variabel, den fortalte også maskinen hvilken type verdi denne kunne ha (int=heltall)
- Siden det er et heltall kunne lengden være 7, men kunne ikke vært 7,1
- Vi kunne alternativt definert som “**double** lengde” - da kunne den vært et tall med desimaler (flyttall)

# Strenger

- Int og double er tross alt ganske likt
- Per nå nøyer vi oss med én ekstra: **String**
- Om vi hadde definert lengden som  
String lengde;  
kunne den vært en vilkårlig tekst
- Vi kunne da etterpå satt den til f.eks.  
lengde="veldig lang"

# Operasjoner avhenger av datatyper

- Med strenger blir også operasjoner annerledes (pluss betyr noe annet).
  - {u1Strings.java}
- Numeraler kan være enten tall eller tekst
  - {u1IntVsString.java}

# Oppsummering av variabler

- Variabler kan brukes for mange formål:
  - Representere en bestemt verdi som skal brukes flere plasser, og tillate enkel endring alle steder  
*(samme tall som skal ganges med 2, plusses med 3 osv.)*
  - Holde på en mellomregning  
*(holde på areal som lengde ganget med bredde)*
  - Holde på verdier hentet utenfra  
*(holde på tastatur-input fra bruker)*

# Oppsummering av variabler

- Variabler er av en bestemt type:
  - Man må fra starten av binde seg på om en variabel skal representere tekst eller tall (*int, double, String*)
  - Datatypen avgjør også hva en bestemt operasjon innebærer (*å plusse tallene 20 og 14 gir 34, å plusse tekstene "20" og "14" gir "2014"*)

# En første test på problemløsning

# Skriv (med blyant og papir) en kode som bytter om verdiene mellom to variable

Prøv/tenk selv i 3 minutt.

Skriv bare linjene som trengs der det nå er rød kommentar.

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
        //skriv koden som trengs her for å bytte om verdiene i  
        //variablene tall1 og tall2, uansett hva verdiene er  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```



# Hvorfor blir følgende kode feil?

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        tall1 = tall2;  
        tall2 = tall1;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Hvorfor blir følgende kode feil?

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        tall1 = 9 ;  
        tall2 = tall1;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Hvorfor blir følgende kode feil?

```
public class u1var1 {
    public static void main(String[] args) {
        int tall1;
        tall1 = X;
        int tall2;
        tall2 = 9;

        tall1 = 9 ;
        tall2 = tall1;

        System.out.println("Nå har tall1 blitt ni:" + tall1 );
        System.out.println("mens tall2 er fire:" + tall2 );
    }
}
```

# Hvorfor blir følgende kode feil?

```
public class u1var1 {
    public static void main(String[] args) {
        int tall1;
        tall1 = X;
        int tall2;
        tall2 = 9;

        tall1 = 9 ;
        tall2 = 9 ;

        System.out.println("Nå har tall1 blitt ni:" + tall1 );
        System.out.println("mens tall2 er fire:" + tall2 );
    }
}
```

# Hvorfor dette er litt triksent

- Hver variabel kan bare holde på én verdi
  - Når man flytter verdien fra **tall1** over til **tall2**, mister man verdien tall2 tidligere holdt på
  - Det er faktisk ikke mulig å bytte om verdiene i begge retninger samtidig
- Hvordan går vi da frem for å løse problemet?
  - Vi glemmer datamaskiner og tenker litt praktisk!
  - Om du har en gul kopp med melk, og en hvit kopp med jus, hvordan bytter du mellom?

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        int tall3;  
        tall3 = tall2;  
        tall2 = tall1;  
        tall1 = tall3;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        int tall3;  
        tall3 = 9 ;  
        tall2 = tall1;  
        tall1 = tall3;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        int tall3;  
        tall3 = 9 ;  
        tall2 = 4 ;  
        tall1 = tall3;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```



# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 8;  
  
        int tall3;  
        tall3 = 9 ;  
        tall2 = 4 ;  
        tall1 = tall3;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 8;  
  
        int tall3;  
        tall3 = 9 ;  
        tall2 = 4 ;  
        tall1 = 9 ;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        int tall3;  
        tall3 = 9 ;  
        tall2 = 4 ;  
        tall1 = 9 ;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# Man må introdusere en tredje variabel!

```
public class u1var1 {  
    public static void main(String[] args) {  
        int tall1;  
        tall1 = 4;  
        int tall2;  
        tall2 = 9;  
  
        int tall3;  
        tall3 = tall2;  
        tall2 = tall1;  
        tall1 = tall3;  
  
        System.out.println("Nå har tall1 blitt ni:" + tall1 );  
        System.out.println("mens tall2 er fire:" + tall2 );  
    }  
}
```

# En lærdom å ta med seg

- Løsningen kommer ofte lettere til en dersom man tenker på fysiske paralleller til problemet
  - Man kan lett låse seg fast i søken etter en teknikalitet, mens det som egentlig trengs er en alternativ innfallsvinkel
  - Dette gjelder ikke bare de enkle problemene vi begynner med de første ukene, men vel så mye de store utfordringene man møter senere i studiet og som ferdig utdannet

**Innlesing fra tastatur**

# Slipp brukeren til!

- De fleste program tar en eller annen jevnlig input fra brukeren
- Dette gir også mye mer dynamikk i programmene enn det vi har sett på til nå

# Enkel input fra brukeren

- `{u1Input.java}`:  
motta input fra tastatur



# Inputt av tall fra brukeren

- `{u1InputSum.java}`:  
Motta input i form av heltall fra tastaturet

# Beslutninger

# Ikke bare følge strømmen

- For det vi har sett på til nå, har programmet alltid fulgt én bestemt sekvens av instruksjoner
  - Vi kan lese ulike startverdier fra tastatur og dermed få ulike resultat, men alltid basert på de samme operasjonene
- Vi trenger mer variasjon!
  - Det blir mye artigere å programmere dersom hvilke operasjoner som utføres også kan avhenge av verdier man regner ut

# Et konkret problem som krever beslutninger

- Problemstilling:
  - Vi ønsker å lage et program som spør brukeren om alder. Dersom brukeren er under 6 år, skal den skrive "du er liten". Dersom brukeren er 6 år eller mer skal den skrive "du er gammel".
- Vi trenger altså to ulike `println`
- Vi må imidlertid sørge for at kun én `println` blir kjørt, hvor hvilken som blir kjørt avhenger av innlest verdi for `alder`

# En uferdig løsning (mangler beslutning)

```
import java.util.Scanner;

public class u1BeslutningUferdig{
    public static void main(String[] args) {
        Scanner in;
        String innlest;
        int alder;
        in = new Scanner(System.in);
        innlest = in.nextLine();
        alder = Integer.parseInt(innlest);

        System.out.println("du er liten");
        System.out.println("du er gammel");
    }
}
```

# Beslutninger i et program: if

- Selve syntaksen (skrivemåten) er veldig enkel:
  - `if (something) {do1; do2; ..}`
  - `if (alder<6)`
    - `{`
    - `System.out.println("du er liten");`
    - `}`
- `{u1Beslutning1.java}`

# En praktisk kortform: **if-else**

- Syntaksen er fremdeles veldig enkel:
  - `if (something) {do1; do2; ..} else {do3; do4; ..}`
- `{u1Beslutning2.java}`

# Prøv en variant selv!

- Modifisert problemstilling:
  - Mindre enn 6: skriv "Du er liten"
  - Mindre enn 3: skriv "Du er veldig liten"
  - Skal alltid skrive ut maksimalt én setning
- Hvordan vil du nå skrive koden?
  - Prøv selv med blyant og papir! (3 minutt)
  - Etterpå diskuter med nabo (3 minutt)
  - (du trenger ikke skrive "main" o.l. - fokuser på det sentrale..)



# Hvorfor blir følgende løsning feil?

```
if (alder<3)
{
System.out.println("du er veldig liten");
}
```

```
if (alder<6)
{
System.out.println("du er liten");
}
```

# Løsning med kombinert uttrykk

```
if (alder<3)
{
System.out.println("du er veldig liten");
}
```

```
if (alder>=3 && alder<6)
{
System.out.println("du er liten");
}
```

# Løsning med **else if**

```
if (alder<3)
{
System.out.println("du er veldig liten");
}
else if (alder<6)
{
System.out.println("du er liten");
}
```

# Og hvorfor går ikke den motsatte **else if**?

```
if (alder<6)
{
System.out.println("du er liten");
}
else if (alder<3)
{
System.out.println("du er veldig liten");
}
```

# Løsning med nøsting

```
if (alder<6)
{
    if (alder<3)
    {
        System.out.println("du er veldig liten");
    }
    else
    {
        System.out.println("du er liten");
    }
}
```

# Konklusjon rundt beslutninger

- Å lære syntaksen (skrivemåten) er enkelt:
  - `if (something) {do1; do2; ..} else {do3; do4; ..}`
- Å bruke det for å løse et konkret problem er utfordringen!
  - Programmering handler om å kreativt løse problemer, ikke å lire av seg kryptiske tegn
  - Problemene vil bli gradvis mer utfordrende utover semesteret

# En siste liten test på problemløsning

Skriv (med blyant og papir) en kode som finner  
den minste av to verdier:

```
import java.util.Scanner;
public class U1Minst {
    public static void main(String[] args) {
        Scanner in;
        int tall1, tall2, minst;
        tall1 = Integer.parseInt( in.nextLine() );
        tall2 = Integer.parseInt( in.nextLine() );
        //skriv kode her som gir variabelen minst
        //den minste av verdiene tall1 og tall2
        System.out.println(minst);
    }
}
```

Mulig løsning: {U1Minst.java}

# Oppsummering

- En variabel er et navn som representerer en verdi (en boks hvor man kan putte en verdi)
- Programmene blir ofte mer dynamiske når man leser inputt fra brukeren
- Beslutninger (**if**) lar programmene utføre ulike operasjoner avhengig av verdien til en variabel
- Programming handler om problemløsning, og krever kreativitet og oppfinnsomhet
- Programmering lærer man gjennom trening og atter trening