INF1000 : Forelesning 1 (del 2)

Java
Variable og tilordninger
Heltall, desimaltall og sannhetsverdier
Utskrift på skjerm

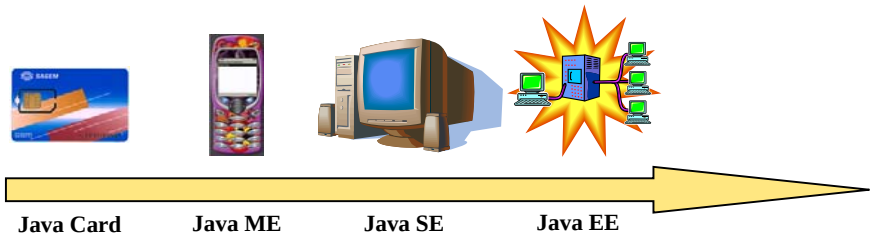
Ole Christian Lingjærde
Gruppen for bioinformatikk
Institutt for informatikk
Universitetet i Oslo

Java

- **Java**:
  - programmeringsspråk med grammatiske regler, semantikk, osv
  - programsystemet som må være installert for å kompilere og kjøre Java-programmer.
- Het opprinnelige Oak og skulle brukes for å styre digitale "dingser" som spillkonsoll, digital kabel-TV (*video on demand*), osv. Dette ble ingen suksess.
- På denne tiden kom Internett for fullt, og Java-teamet brukte Oak/Java teknologien til å lage en nettleser i 1994 som de kalte WebRunner (etter filmen Blade Runner), senere omdøpt til HotJava. Første nettleser med dynamisk innhold og animasjon!
- Java er gratis og kan lastes ned fra http://java.sun.com (og fra Ifi-CD'en).
- Versjonen som brukes nå kalles *Java 2 plattformen.*


James Gosling

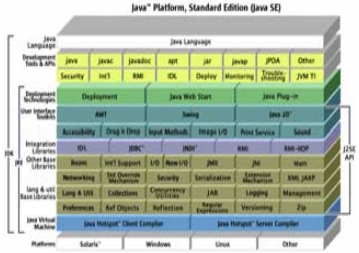
Ulike varianter for ulike behov



Java Standard Edition (Java SE)

To sentrale begreper:

- **Java SE Runtime Environment (JRE)**
  System for å kjøre kompilerte Java-programmer.
- **Java SE Development Kit (JDK)**
  JRE + system for å utvikle Java- programmer (kompilere, feilsøke, dokumentere, m.m.)

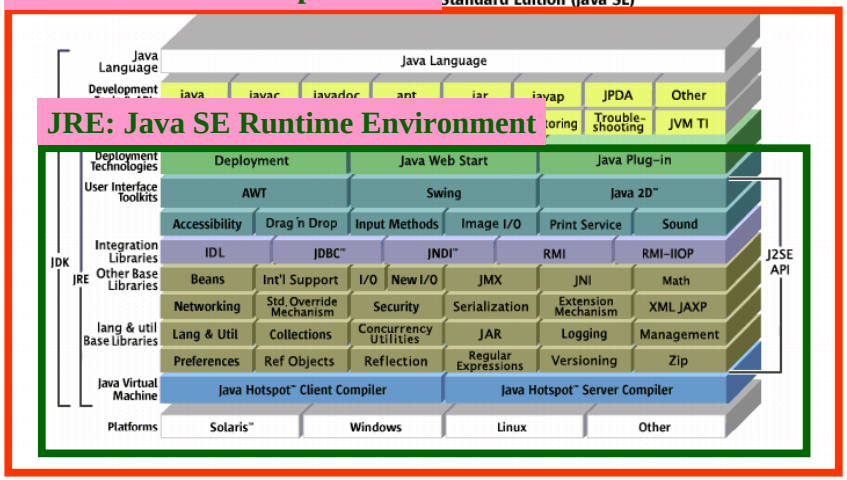


I tillegg til dette kan man legge til egne "pakker" som gir økt funksjonalitet.
I INF1000 brukes det en slik pakke, med navn easyIO.

Java Standard Edition (Java SE)

JDK: Java SE Development Kit

JRE: Java SE Runtime Environment



Installasjon av Java på egen maskin

For å installere Java på egen maskin:

TRINN 1: Installere Java SE Development Kit

Kan hentes fra Ifi-CD'en eller fra Sun Microsystems:

http://java.sun.com/javase/downloads/index.jsp

Velg da nedlasting av JDK 6 Update N (N = versjonsnr).

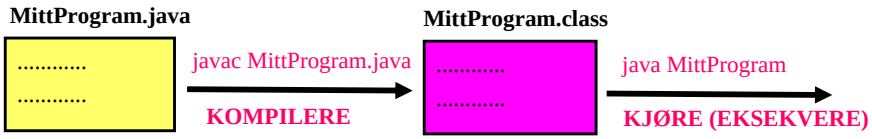
Merk: På Mac OS X er Java allerede ferdig installert.

TRINN 2: Installere INF1000-pakken easyIO

Tilleggsfunksjonalitet som benyttes i undervisningen. Kan hentes fra Ifi-CD'en eller fra lærebokas nettsider:

http://www.universitetsforlaget.no/java

Kompilere, kjøre



Programfil (lages av deg i en teksteditor)

Kompilert program ("class-fil")

Kompilere og kjøre i Linux, Windows, Mac OS X

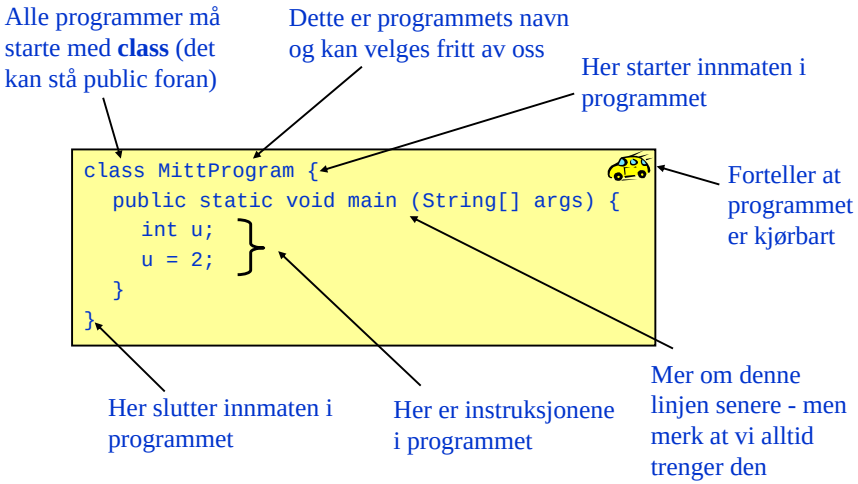
Generell oppskrift:

1. Åpne et terminalvindu/kommandovindu
2. Endre filområde (directory) til der programfilen ligger
3. Kompiler med **javac MittProgram.java**
4. Kjør med **java MittProgram**

For utfyllende forklaringer, se:

http://www.ifi.uio.no/ifidvd/Programmer/Linux/Java/index.html
http://www.ifi.uio.no/ifidvd/Programmer/Win/Java/index.html
http://www.ifi.uio.no/ifidvd/Programmer/Mac/Java/index.html

Bestanddelene i et Java-program



Eksempel: Utskrift på skjerm

Her kommer utskriften i kommandovinduet:

```
class MittProgram {
   public static void main (String[] args) {
      System.out.println("Velkommen til UiO");
      System.out.print("og velkommen til ");
      System.out.println("INF 1000");
   }
}
```

Test programmet

Eksempel 2: Utskrift på skjerm

Her kommer utskriften i et eget vindu:

```
import javax.swing.*;

class MittProgram2 {
   public static void main (String[] args) {
      JOptionPane.showMessageDialog(null, "Velkommen til UiO");
   }
}
```

Test programmet

Viktig

- Instruksjoner i Java følger en presis syntaks. Spesielt interesserte kan se på den her (ikke pensum): http://java.sun.com/docs/books/jls/third_edition/html/j3TOC.html
- Nær sagt *enhver trykkfeil* (f.eks. at det står **clas** istedet for **class**) vil gjøre at instruksjonen ikke blir forstått, eller at den blir forstått feil.

Variabler (minneplasser)

- Vi kan reservere plass i datamaskinens hukommelse til et heltall:

```
int lengde;
```

slutt på instruksjonen

sett av plass til et heltall (engelsk: **int**eger)

gi plassen navnet "lengde"

- Vi kan fylle plassen (= variabelen) med en verdi:

```
lengde = 14;
```

leses "settes lik" eller "gis verdien"

- Vi kan senere avlese/bruke verdien:

```
int svar;
svar = lengde * 2;
```


Variabler

- Etter at vi har reservert plass i minnet til en variabel, f.eks. slik:

```
int lengde;
```

så kan vi endre verdien til variabelen så mange ganger vi vil, f.eks.:

```
lengde = 14;
lengde = 434;
lengde = lengde + 2;
lengde = lengde;
```

- Hva skjer egentlig når vi skriver `lengde = lengde + 2;` ? I detalj:
  1. Verdien som ligger i variabelen `lengde` hentes fram (f.eks. 434)
  2. En ny verdi regnes ut ved å legge til 2 (434 + 2 = 436)
  3. Variabelen `lengde` gis den nye verdien (436)

Variabel-deklarasjoner

- Instruksjoner av typen

```
int alder;
int vekt;
int personnummer;
```

kalles variabel-deklarasjoner.

- Vi kunne erstattet de tre instruksjonene ovenfor med:

```
int alder, vekt, personnummer;
```

(NB: komma mellom variablene)

Normalt er det ryddigere å bruke den første varianten, men du møter begge i kurset.

Variabeldeklarasjoner

- Variable kan deklareres hvor som helst i et program, og de kan endres hvor som helst etter at de er deklarert.
- Variable har ingen verdi rett etter en deklarasjon:

```
int lengde;
lengde = lengde + 1;        // Ulovlig!
```

- Vi kan gi variable en verdi når vi deklarerer dem:

```
int lengde = 4;
lengde = lengde + 1;        // Lovlig
```

- Vi kan også vente med å gi en variabel verdi:

```
int lengde;
.....
lengde = 4;
lengde = lengde + 1;        // Lovlig
```

Hvis du glemmer å initialisere en variabel

Forsøk på å hente/benytte verdien til en variabel som ikke har blitt initialisert gir feilmelding:

```
C:\Eksempel.java:4: variable lengde might not have been
   initialized
     lengde = lengde + 1;
              ^
1 error

Tool completed with exit code 1
```

Vanlig feil, så lær deg å kjenne igjen denne feilmeldingen.

Ting å passe på

- Vi kan ikke deklarere flere variable med samme navn. Dette er ulovlig:

```
int alder;
int alder;      (Ulovlig - variabelen alder er allerede deklarert!)
```

- En variabel kan hete hva som helst, men bruk bare bokstaver og tall, og begynn alltid navnet med en bokstav. Eksempler:

```
int etVeldigLangtVariabelNavn;     (Denne en lovlig)
int år2001;                        (Denne er også lovlig)
int 2001år;                        (Denne er ikke lovlig)
```


Vi kan ha mange variable

- I et program kan vi deklarere så mange variable vi vil, f.eks.

```
int alderKari;
int alderPer;
int alderOla;
int sumAlder;
alderKari = 20;
alderPer = 10 + alderKari;
alderOla = 10 + alderKari + alderPer;
sumAlder = alderKari + alderPer + alderOla;
```

- Hvilken verdi har de fire variablene når alle instruksjonene ovenfor er utført?

alderKari:	20
alderPer:	30
alderOla:	60
sumAlder:	110

Variabel-tilordninger

- Instruksjoner av typen

```
alder = 3;
```

kalles variabel-tilordninger (eller bare tilordninger).

- Generell form:

```
variabel = uttrykk;
```

her må det stå navnet på en variabel som er deklarert

her må det stå en verdi eller et regneuttrykk. To eksempler:
345
(56+36) * 14 - 3 + 53

- Først utføres regnestykket på høyresiden av = , og deretter settes variabelen på venstre side av = lik den utregnede verdien

Avsluttende om variable

- Unngå i størst mulig utstrekning å samle mange variabeldeklarasjoner på en linje:

```
int år, måned, dag, alder;
```

Uoversiktlig

```
int år;    // Fødselsår
int måned; // Fødselsmåned (1-12)
int dag;   // Fødselsdato (1-31)
int alder; // Alder i antall år
```

Oversiktlig

- Deklarer variable først når du trenger dem – ingen grunn til å samle alle variabeldeklarasjoner ett sted med mindre de naturlig hører sammen.

Datatyper vi kommer til å benytte

Datatype	Beskrivelse	Eksempel
int	heltall	`int k = 3;`
double	desimaltall	`double x = 3.14;`
boolean	sannhetsverdi	`boolean b = true;`
char	tegn	`char c = '@';`
String	tekst	`String s = "Hei på deg";`

Det finnes noen flere (short, long, byte, ...) som du gjerne må se på, men de er ikke nødvendig å kjenne til i dette kurset.

De numeriske datatypene

- int og double er eksempler på numeriske datatyper
- Java har ialt seks numeriske datatyper:

Datatype	Lovlige verdier
byte	{-128, -127, ..., 127}
short	{-32768, ..., 32767}
int	$\{-2^{31}, ..., 2^{31}-1\}$
long	$\{-2^{63}, ..., 2^{63}-1\}$
float	(-3.4e38, 3.4e38)
double	(-1.7e308, 1.7e308)

I praksis er det disse to du trenger i INF 1000

- Antall signifikante siffer er 6-7 med float og 14-15 med double.

Desimaltall

- Variable av typen int kan bare holde heltallsverdier (...-2, -1, 0, 1, 2, ...)
- Hvis vi ønsker å lagre desimaltall (også kalt flyttall) kan vi bruke double:

```
double pi = 3.14;
double radius = 0.332;
double omkrets = 2 * pi * radius;
```

- Vi kan godt gi et heltall som verdi til en double-variabel:

```
double radius = 2;
```

...men inne i datamaskinen vil det bli lagret med desimaler: 2.0000.....

- Eksempel:

```
int radius = 2;                (Tallet 2 som heltall)
double nyradius = radius;      (Tallet 2 som desimaltall)
```

- Desimaltall kan angis på flere måter:

```
-10.5    .435    15.      1.23e5   1.23e+4   1.15e-3
```

Sannhetsverdier

- I programmer har vi ofte behov for å ta avgjørelser som avhenger av om noe er tilfelle eller ikke, f.eks. om det er sant eller usant at x > 0.

- Derfor finnes det en egen variabeltype som bare kan holde de to verdiene true og false. Denne typen heter boolean:

```
boolean b;
b = true;
b = false;

int x = 3;
b = (x < 3);       // Nå får b verdien false
```