

Leksjon 1

Forstå program

Kom i gang med Java

- Lese programkode
 - Realistisk program: [FinnOrdIFil.java](#)
 - Treningsprogram: [Mal.java](#)
- Skrive programkode
 - Tekstbehandler eller IDE
- Kompilering av program
 - Kontroll av programkode og oversettelse til kjørbart versjon, f.eks: FinnOrdIFil.class

Utføre program

- Dialog med brukeren



Input

?

Filnavn: avslutt med 0

handleliste.txt

OK Cancel



Input

?

Søkeord: slutt med 0

juice

OK Cancel

- Inndata fra fil

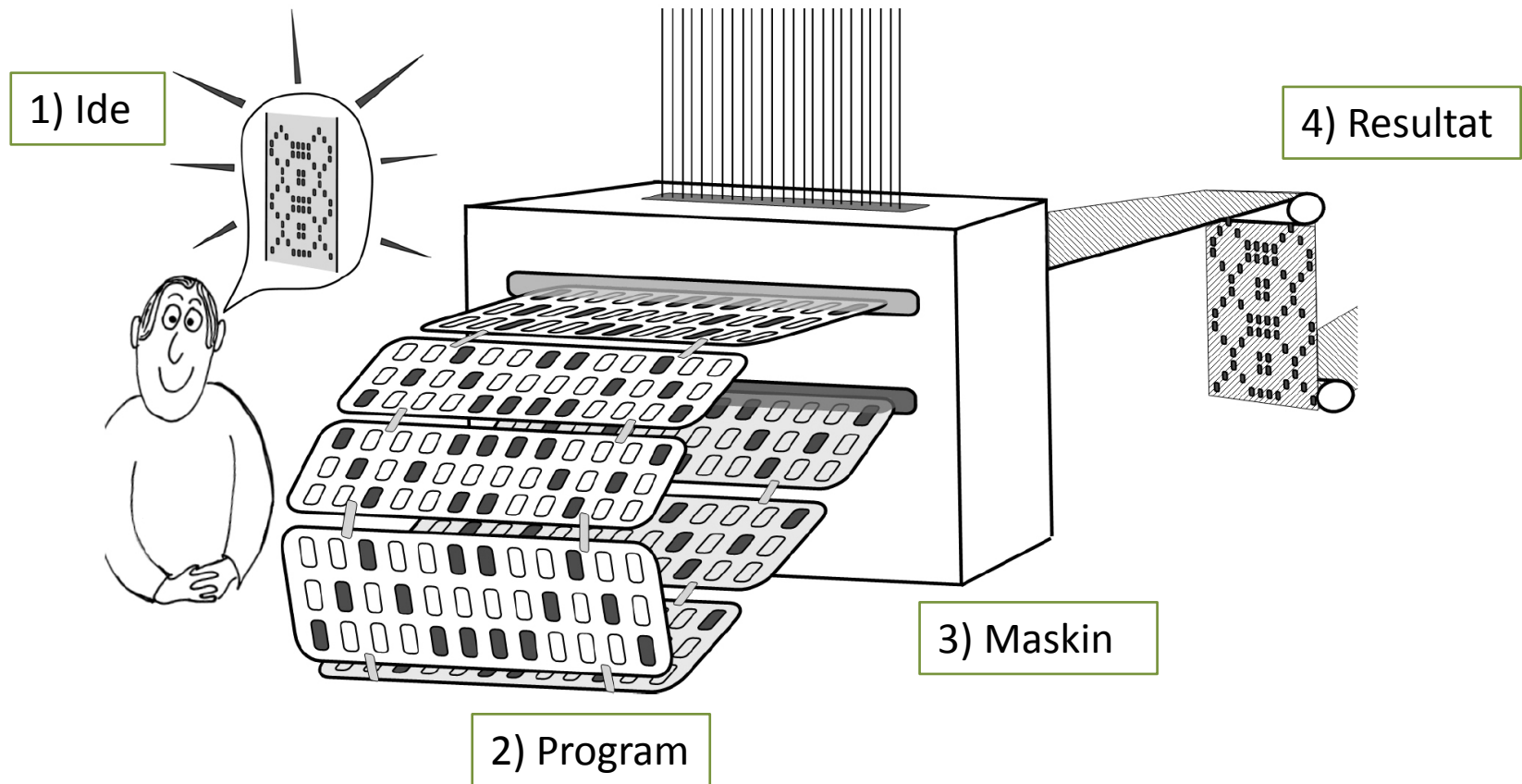
```
handleliste.txt  
  
Handleliste:  
  
Brød  
Leverpostei  
Appelsinjuice  
Eplejuice  
Pizza
```

- Utskrift til konsollet

```
5:Appelsinjuice  
6:Eplejuice
```

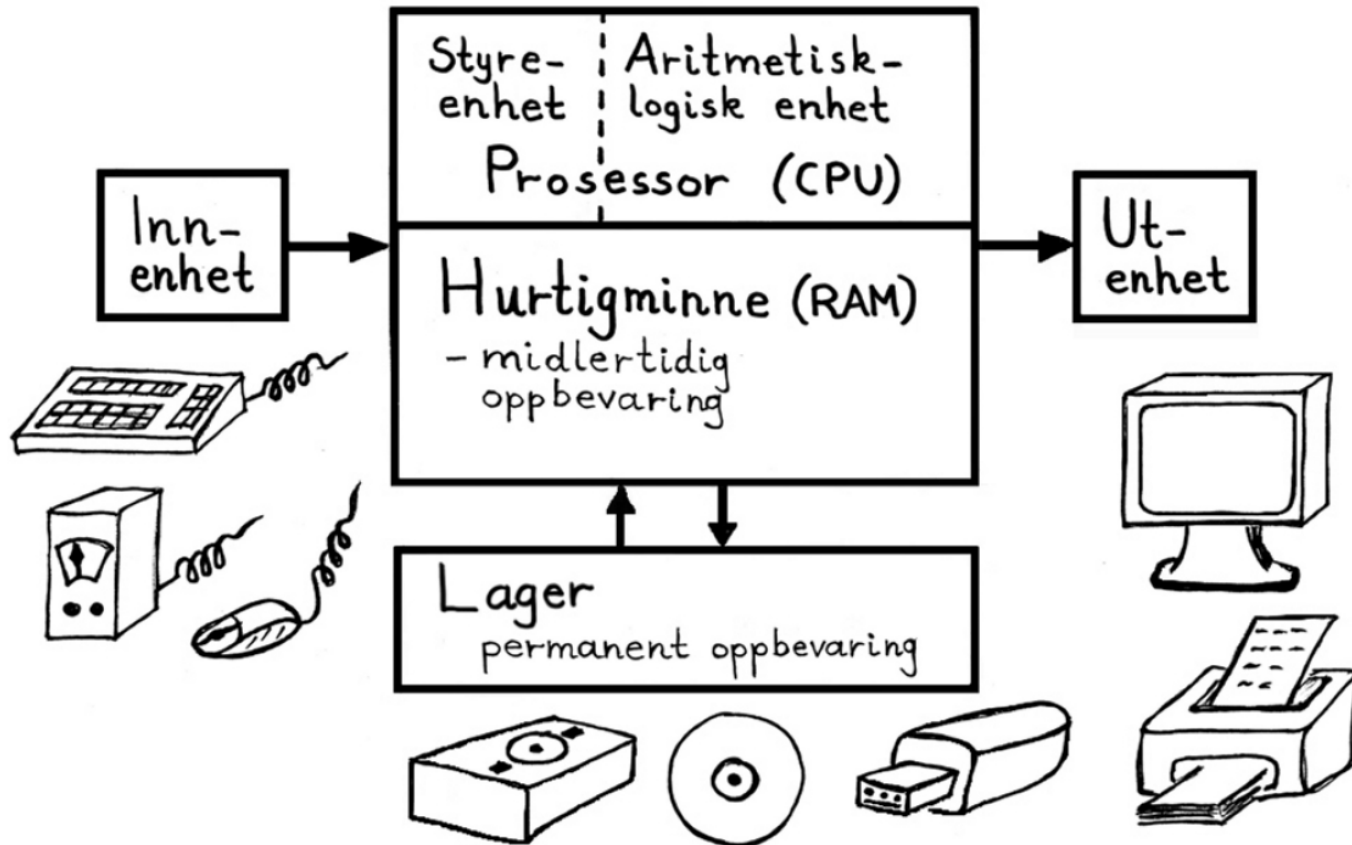
Maskin og program

Fra ide til resultat, via program og maskin



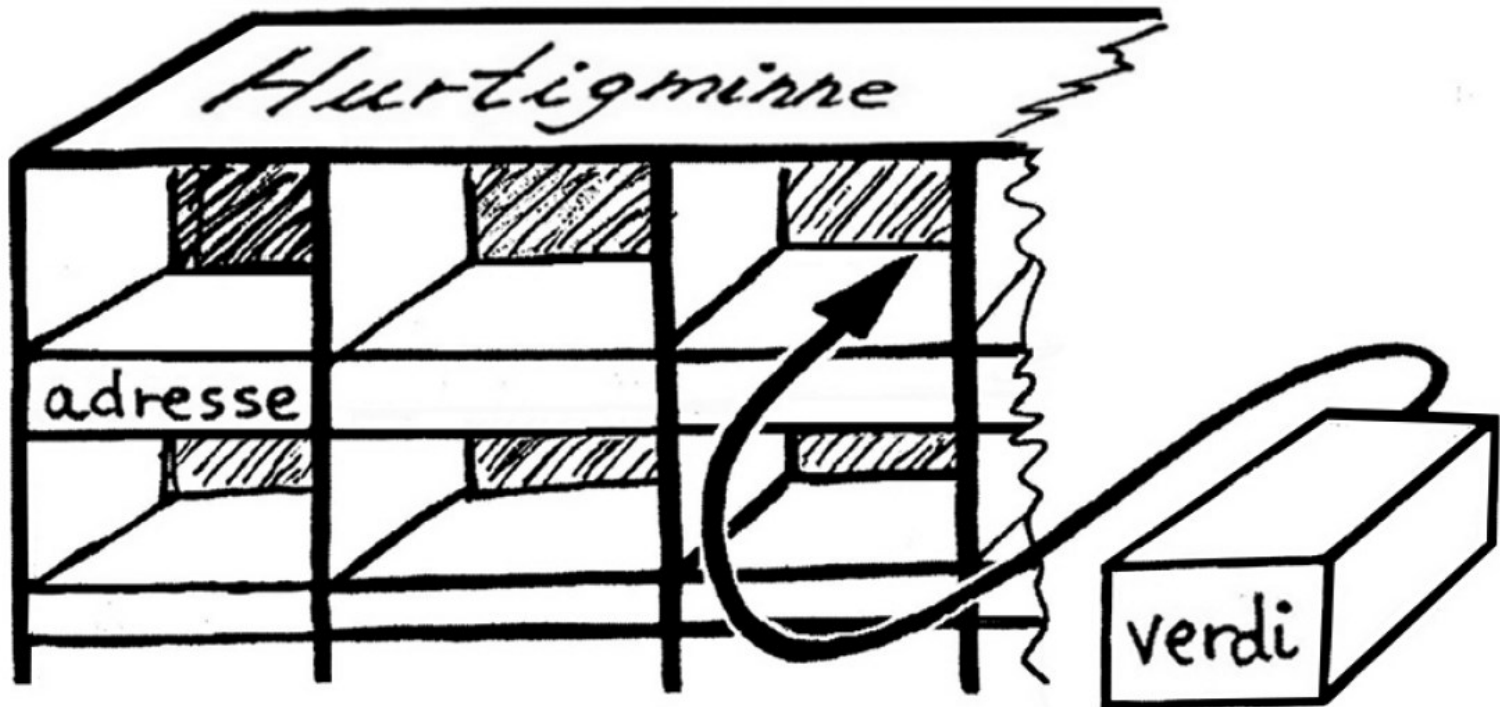
Maskin og program

Program: Sekvens av instruksjoner for å få utføre en oppgave på en maskin.



Hurtigminne

- **Variabel:** Navnsatt lagringsplass i hurtigminnet (oppbevaringsboks).
- **Verdi:** Innholdet i en variabel.
- **Datatype:** Hvor mye plass skal avsettes til en verdi, og hvordan skal den lagres og avleses?



Tallsystem og lagring

- Vi benytter 10-tallsystemet:

$$\begin{aligned} 65042 &= 6 \times 10^4 + 5 \times 10^3 + 0 \times 10^2 + 4 \times 10^1 + 2 \times 10^0 \\ &= 60\,000 + 5000 + 0 + 40 + 2 \end{aligned}$$

- Maskin benytter 2-tallsystemet:

$$\begin{aligned} 1101 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 8 + 4 + 0 + 1 = 13 \end{aligned}$$

Tallsystem og lagring

- Minste lagringsenhet: BiT = **B**inary digIT (0 , 1)
- Alle data og all programaktivitet håndteres som binærsekvenser

Hurtigminne

```
0110101100111011001110101  
1011000001101101 ...
```

Program

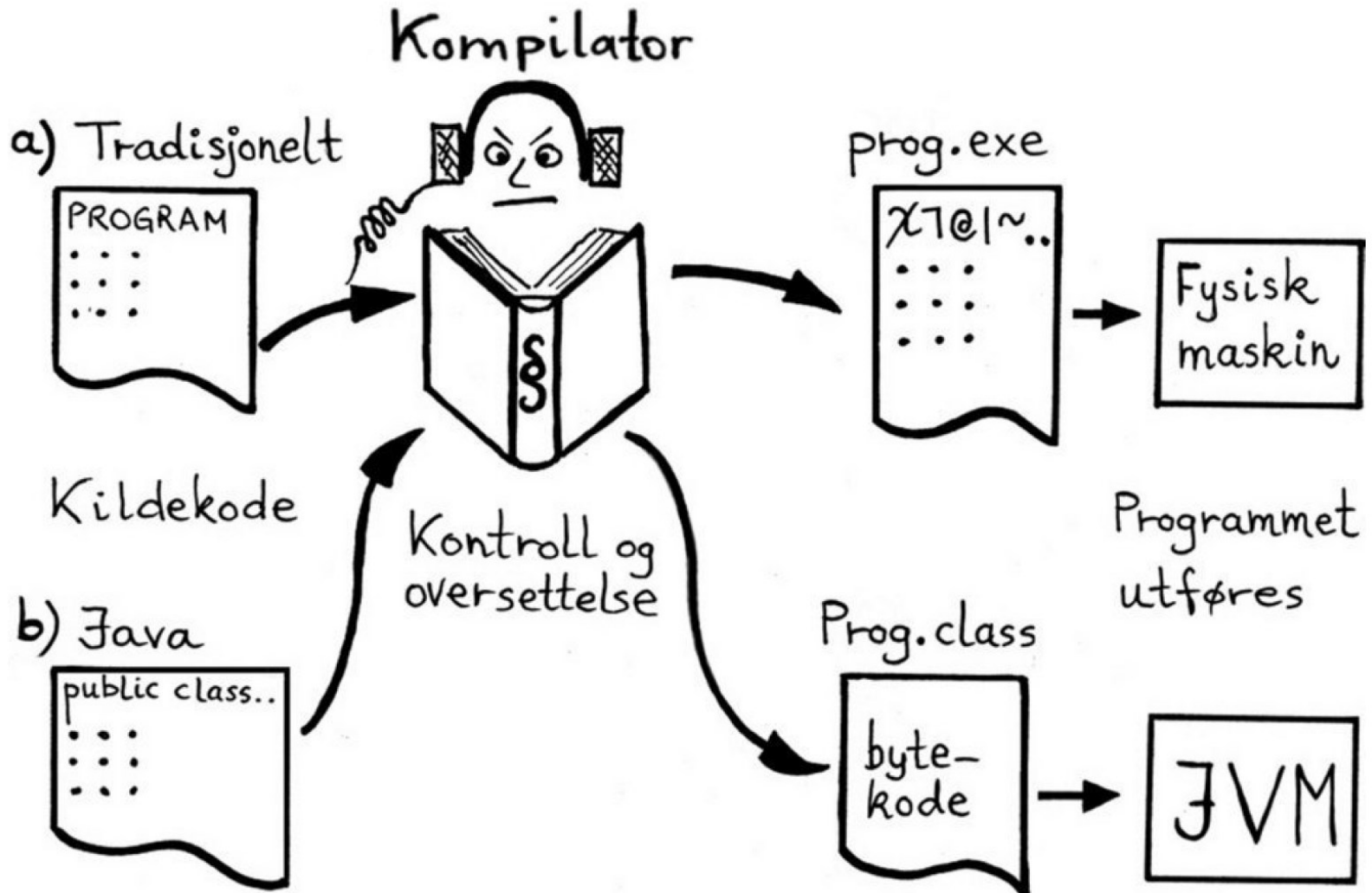
:

```
0101100111011001110101101  
1000001101100010101011 ...
```

Data

:

Kompilering av program



Programmeringsspråk

Språkopplæringen inneholder tre trinn

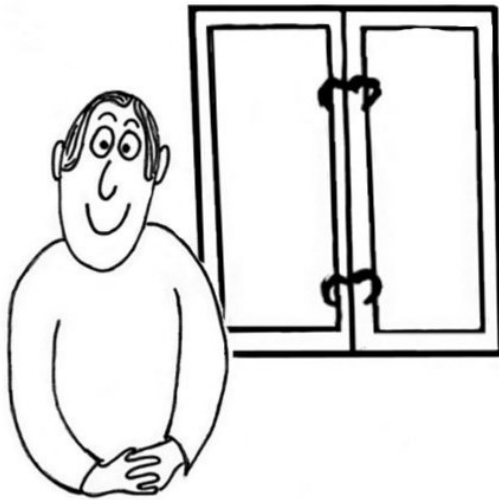
- Syntaks: Setningers oppbygging og grammatikk
- Semantikk: Setningers meningsinnhold
- Pragmatikk: Setningers bruk og effektivitet

Syntaks og semantikk vil være hovedfokuset i den grunnleggende språkopplæringen.

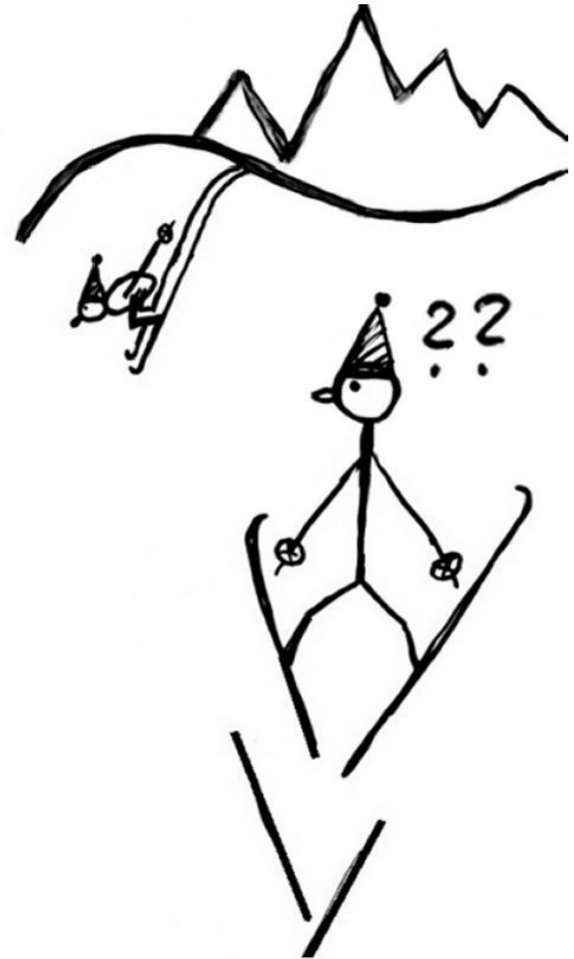
I pragmatikk studeres mer avanserte temaer som algoritme (oppskrift) og datastruktur (lagringsmåte), og man ser på programmernes effektivitet.

Syntaks og semantikk

①



②



Lukk opp vinduet!

Det første programmet

- Standard program for å skrive en hilsen i konsollet.
- Alle kjørbare program har `main`-metode.
- Bruker `import`-setning for å hente bibliotekressurser.
- Bruker besvergelsen `static` for å slippe å skrive `System` i koden for hver utskrift til konsollet.

```
1 import static java.lang.System.*;
2
3 public class SkrivMelding {
4     public static void main(String[] args) {
5         out.println("Hei verden!");
6     }
7 }
```

Kommunikasjon via dialogvindu

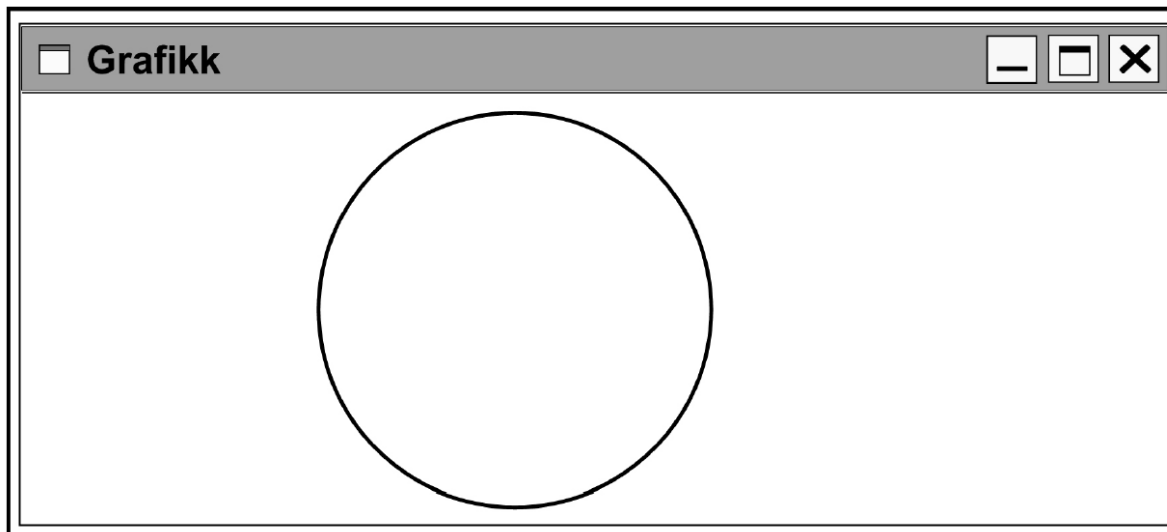
- Standard programkode for å lese inn tekst fra brukeren.
- Innlesing og utskrift foregår i hvert sitt lille grafiske vindu, kalt *dialogvindu*.
- Bruker besvergelsen `static` for å slippe å skrive `JOptionPane` i koden for hvert dialogvindu.

```
import static javax.swing.JOptionPane.*;

public class Dialog {
    public static void main(String[] args) {
        String navn = showInputDialog("Navn: ");
        showMessageDialog(null, "Hei " + navn);
    }
}
```

Enkel grafikk

```
1 public class Grafikk extends EasyGraphics {
2
3     public static void main(String[] args) {
4         launch(args);
5     }
6
7     public void run() {
8         makeWindow("Grafikk", 350, 150);
9         drawCircle(150, 70, 60);
10    }
11 }
```



EasyGraphics er en ressurs for å kunne lage enkel grafikk tilpasset de grunnleggende programmerings-temaene.

Programmet har `main`-metode, men tegningen bygges opp i `run`-metoden.

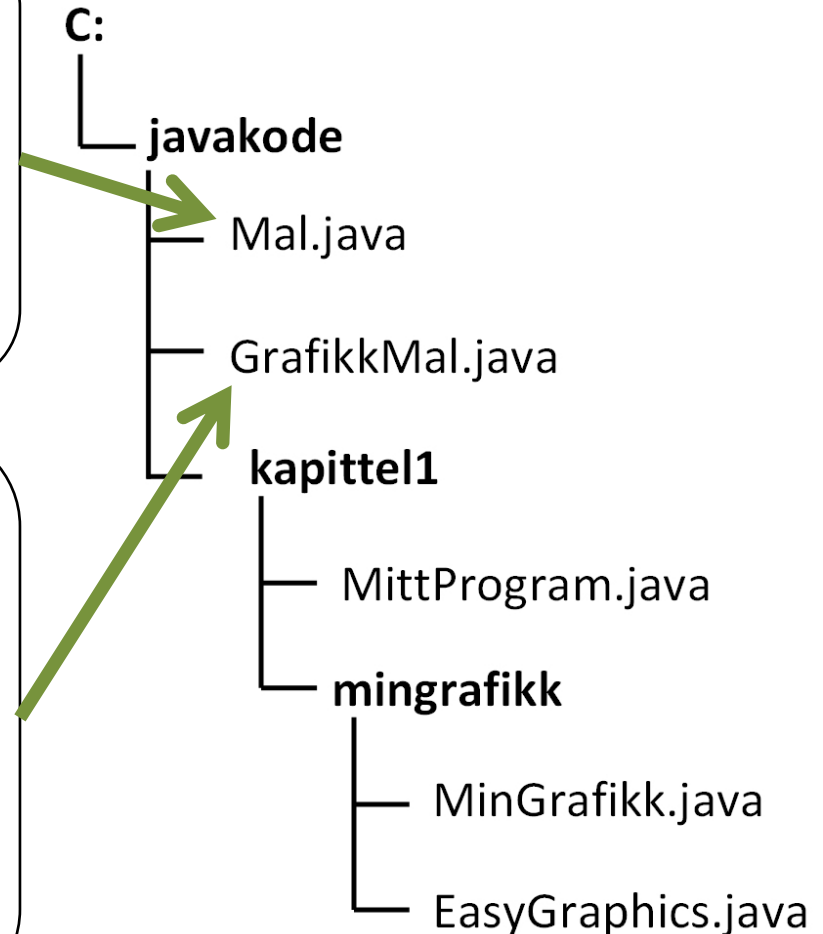
Malfiler og mappestruktur

```
/* Mal for vanlig Java-program
*/
import static java.lang.System.*;
import static javax.swing.JOptionPane.*;

public class Mal {
    public static void main(String[] args) {
        // Skriv Java-setninger her
    }
}
```

```
/*
 * Mal for grafikk-program
 */
public class GrafikkMal extends EasyGraphics {
    public static void main(String[] args) {
        launch(args);
    }

    public void run() {
        makeWindow("Tittel", 600, 400);
        // Skriv tegnekommandoer her
    }
}
```



Noen språkelement i Java

- Reserverte ord (se s.403):

`public class static import void`

- Standardnavn: `main String`

- Egendefinerte ord: `SkrivMelding args out`

- Spesialtegn: `; . , {} [] () // /* */`

- Kommentarsætning: Korte forklaring

- Metode: Selvstendig, navngitt, delprogram

Programproduksjon

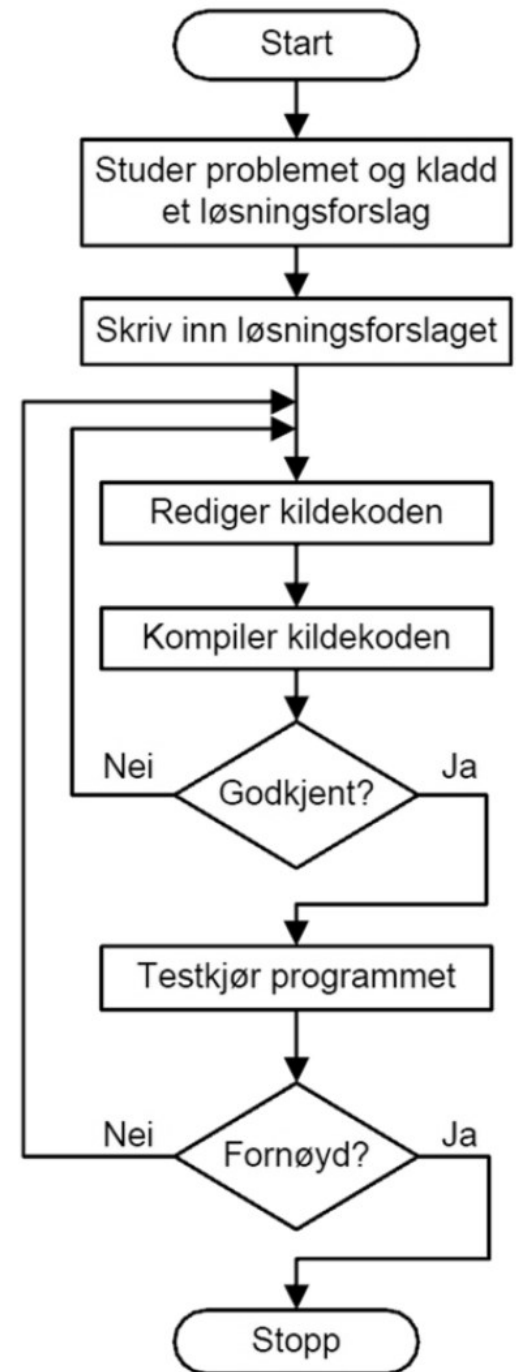
- Kladdeark og blyant
- Editor, tekstbehandler
- Rediger koden, gjør evt. endringer
- Kompilator: `javac EtProgram.java`
- Enten: Rett opp feil som er påpekt, eller: programkoden er godkjent

- JVM: `java EtProgram`
- Fornøyd med resultat?

Enten: Nei

eller: Ja

Ferdig!



Programproduksjon: Løsningskladd

- Hvordan skal en problemstilling/oppgavetekst forstås?
- Hva skal utføres, hva er ønsket resultat?

Finn linjer i en fil som inneholder et søkeord.

Inndata

Filnavn: handleliste.txt

Søkeord: juice

Utdata

5. Appelsinjuice

6. Eplejuice

1. Handleliste:
- 2.
3. Brød
4. Leverpostei
5. Appelsinjuice
6. Eplejuice
7. Pizza