

Implementering: Svoemmer

```
public class Svoemmer
{
    private String navn;
    private boolean koen;
    private int aargang;

    public Svoemmer(String nv, boolean k, int a)
    {
        navn    = nv;
        koen     = k;
        aargang  = a;
    }

    public String getNavn() { return navn; }

    public boolean getKoen() { return koen; }

    public int getAargang() { return aargang; }

    public String toString(){ return navn + "    Aargang: " + aargang; }
}
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 9

Implementering: Tid

```
public class Tid
{
    private int minut;    private int sekund;    private int hundrede;

    public Tid(int m, int s, int h) throws SvoemException
    {
        if (0 <= m && m <= 99 && 0 <= s && s <= 59 && 0 <= h && h <= 99)
        {
            minut = m;    sekund = s;    hundrede = h;
        }
        else throw new SvoemException( "Ulovlig tid: " + m + "." + s + "." + h);
    }

    public boolean less(Tid t)
    {
        return  minut < t.getMinut()
            || (minut == t.getMinut() && sekund < t.getSekund())
            || (minut == t.getMinut() && sekund == t.getSekund()
                && hundrede < t.getHundrede());
    }

    public String toString()
    {
        return cvt(minut, " ") + "." + cvt(sekund, "0") + ":" + cvt(hundrede, "0");
    }

    private static String cvt(int i, String s)
    {
        return (i>9 ? "" : s) + i;
    }
}
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 10

Implementering: Naale

```
public class Naale
{
    private String disciplin;    private boolean koen;

    private Tid[] tider;
    private static final String[] NAALE = {"Elite", "Guld", ...};
    private static final int ANTAL_NAALE = NAALE.length;

    public Naale(String disc, boolean k, Tid[] tr) throws SvoemException
    {
        if (tr.length == ANTAL_NAALE) ...
    }

    public boolean udmaerkelseOK(Tid t){ return t.lessEq(tider[ANTAL_NAALE-1]);}

    public String findNaal(Tid t) throws SvoemException
    {
        if (udmaerkelseOK(t))
        {
            int i = 0;
            while (tider[i].less(t))
                i++;
            return NAALE[i];
        }
        else throw new SvoemException("tiden " + t + " giver ikke en naal");
    }

    ...
}
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 11

Implementering: Loeb

```
public class Loeb
{
    private String disciplin;
    private boolean koen;
    private Svoemmer[] startliste;

    public Loeb(String disc, boolean k, Svoemmer[] stl) throws SvoemException
    {
        for (int i=0; i<stl.length; i++)
            if (stl[i].getKoen() != k) throw new SvoemException("svoemmer ...");

        disciplin = disc;    koen = k;    startliste = stl;
    }

    public String toString()
    {
        String res = "Startliste i disciplinen: " + disciplin
            + " for " + (koen ? "damer" : "herrer") + "\n";

        for (int i=0; i<startliste.length; i++)
            res += startliste[i] + "\n";

        return res;
    }

    ...
}
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 12

Implementering: ResultatListe

```
public class ResultatListe
{ private Loeb loeb;
  private Tid[] tidsliste;
  ...

  public String tildelNaale(Naale nt) throws SvoemException
  { if (loeb.getDisciplin() != nt.getDisciplin()
      || loeb.getKoen() != nt.getKoen())
      throw new SvoemException("disciplin eller koen passer ikke");

  String res = "";

  for (int i=0; i<tidsliste.length; i++)
    if (nt.udmaerkelseOK(tidsliste[i]))
      res += loeb.getSvoemmer(i).getNavn() + " tildeles "
        + nt.findNaal(tidsliste[i]) + "naalen\n";

  return res;
}
...
}
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 13

Hovedprogram (I)

Indfør konstanter for køn og discipliner:

```
static final boolean DAMER = true;
static final boolean HERRER = false;
```

```
static final String BRYST100 = "100 bryst";
static final String BRYST50 = "50 bryst";
```

Et nåletidsobjekt oprettes på følgende måde:

```
Naale naaleBryst100Herrer =
  new Naale(BRYST100, HERRER,
    new Tid[]
      { new Tid(1,3,90), new Tid(1,7,59),
        new Tid(1,12,59), new Tid(1,24,0),
        new Tid(1,30,0), new Tid(1,39,0),
        new Tid(2,3,0), new Tid(2,21,0)
      }
  );
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 14

Hovedprogram (II)

Et løb oprettes på følgende måde:

```
Loeb bryst100Herrer =
  new Loeb(BRYST100, HERRER,
    new Svoemmer[]
      { new Svoemmer("Mads", HERRER, 87),
        new Svoemmer("Finn", HERRER, 90),
        new Svoemmer("Aage", HERRER, 79),
        new Svoemmer("Hans", HERRER, 82),
        new Svoemmer("Sten", HERRER, 89)
      }
  );
```

En resultatliste oprettes på følgende måde:

```
ResultatListe bryst100HerrerResultater =
  new ResultatListe(bryst100Herrer,
    new Tid[]
      { new Tid(1,44,25), new Tid(1,30,0),
        new Tid(1,24,42), new Tid(1,2,59),
        new Tid(2,22,25)
      }
  );
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 15

Udskrift

Ved udførelse af sætningen

```
System.out.println( bryst100HerrerResultater.tildelNaale(naaleBryst100Herrer) );
```

udskrives

```
Mads tildeles Flippernaalen
Finn tildeles Broncedelfinnaalen
Aage tildeles Broncedelfinnaalen
Hans tildeles Elitenaalen
```

02161 Software Engineering 1 ©Michael R. Hansen, Spring 2009 – p. 16