



# INF1000: Forelesning 13

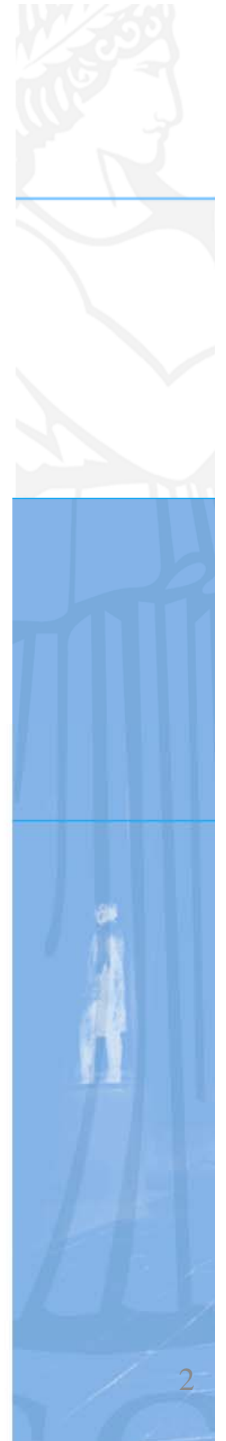
Tidligere eksamensoppgaver

- Høsten 2004: oppgave 11-14
- Våren 2005: oppgave 1 og 3



UNIVERSITETET  
I OSLO

# EKSAMEN HØSTEN 2004



## Oppgave 11

I programmet nedenfor skal du lage en konstruktør til klassen Sirkel som har sirkelens radius som parameter. Du skal også lage en objektmetode i klassen Sirkel som regner ut arealet og returnerer denne verdien (du kan bruke formelen:  $\text{areal} = 3.14 * \text{radius} * \text{radius}$ ). Du skal også skrive programkode i main som oppretter to sirkler, en med radius 10.4 og en med radius 244.99. Fra main skal du så kalle på arealberegningsmetoden i hvert av de to objektene og skrive ut en linje for hver sirkel med radius og areal.

1.  
2.  
3.

```
class Sirkel {
    double radius;

    // skriv konstruktør her

    // skriv objektmetode her som beregner og returnerer arealet

} // end class Sirkel
```





## Oppgave 12

På en barneskole finnes det 7 klasstrinn som hver igjen består av en eller flere grupper. Hver gruppe består så av fra 10 til 20 elever. Tegn et UML-klassediagram med de 4 (Java-)klassene som kan brukes til å representere dette problemet. Gi navn på disse klassene og plasser antall på forholdet mellom disse klassene.

Svar:

## Oppgave 13

Anta at følgende program utføres:

```
class Tre {
    Tre venstre, høyre;
    static int ant = 0;

    Tre (int x) {
        ant++;
        if (x > 10 ) venstre = new Tre(x-10);
        if (x > 20 ) høyre = new Tre(x-20);
    }

    public static void main(String[] args) {
        ant =0;
        Tre t1 = new Tre(11);
        System.out.println("1) " + ant);

        ant =0;
        Tre t2 = new Tre(21);
        System.out.println("2) " + ant);

        ant =0;
        Tre t3 = new Tre(31);
        System.out.println("3) " + ant);
    }
}
```

Tegn først opp for deg selv og finn ut hvor mange Tre-objekter vi får laget av de ulike setningene i main. Svar så på spørsmålet: hva skriver programmet ut på skjermen?



## Oppgave 14

Anta at du har en tekstfil **Bok.txt** som f.eks. inneholder teksten i en roman. Lag et komplett, kompilerbart program som leser filen ord for ord og som teller opp antall forekomster av hvert enkelt ord i filen. Programmet ditt skal deretter opprette en ny fil **Ordforekomster.txt** og skrive resultatet av analysen til denne filen. Følgende opplysninger om **Bok.txt** skal skrives til denne filen: (a) antall ord totalt; (b) antall ulike ord; og (c) en liste over alle ord som forekommer minst 6 ganger. Hver linje av denne listen skal inneholde ett ord og antall forekomster av dette ordet, atskilt av blanke tegn.

Du kan anta at filen **Bok.txt** kun inneholder bokstaver og blanke tegn, dvs spesialtegn som komma, punktum, kolon, osv, forekommer ikke i teksten. Du kan også anta at alle ord i teksten er skrevet med små bokstaver, slik at du slipper problemer med stor forbokstav i begynnelsen av en setning, osv.

Hint: bruk en HashMap til å holde rede på ordene du leser i **Bok.txt** og antall forekomster av dem. Tenk deg nøye om hva du skal legge inn i HashMap'en. Husk at både nøkkelen og verdien du legger inn i en HashMap må være (pekere til) objekter. Du vil kanskje ønske å definere en klasse Ord hvor hvert objekt lagrer et ord og antall forekomster av ordet.



# Klassen BokProgram – med main

```
import easyIO.*;
import java.util.*;

class BokProgram {
    public static void main(String[] args) {
        BokAnalyse ba = new BokAnalyse();
        ba.lesFraFil("Bok.txt");
        ba.skrivTilFil("Ordforekomster.txt");
    }
}
```





# Klassen Ord

```
class Ord {  
    private String tekst;  
    private int antall;  
  
    public Ord(String o) {  
        tekst = o;  
        antall = 1;  
    }  
  
    public void nyForekomst() {  
        antall++;  
    }  
  
    public int getAntall() {  
        return antall;  
    }  
}
```

# Klassen BokAnalyse – med LesFraFil



UNIVERSITETET  
I OSLO



```
class BokAnalyse {
    HashMap<String,Ord> alleOrd = new HashMap<String,Ord>();
    int antallOrdTotalt = 0;

    void lesFraFil(String filnavn) {
        In innfil = new In(filnavn);

        while (!innfil.lastItem()) {
            String ord = innfil.inWord();
            antallOrdTotalt++;

            if (alleOrd.containsKey(ord)) {
                Ord o = alleOrd.get(ord);
                o.nyForekomst();
            } else {
                Ord o = new Ord(ord);
                alleOrd.put(ord,o);
            }
        }
    } // slutt på lesFraFil
```

# Klassen BokAnalyse forts: SkrivTilFil



```
void skrivTilFil(String filnavn) {
    Out utfil = new Out(filnavn);

    utfil.outln("Antall ord totalt: " + antallOrdTotalt);
    utfil.outln("Antall ulike ord: " + alleOrd.size());

    Iterator it = alleOrd.keySet().iterator();

    while (it.hasNext()) {
        String ord = (String) it.next();
        Ord o = alleOrd.get(ord);

        int antall = o.getAntall();
        if (antall > 5) {
            utfil.outln(ord + "\t" + antall);
        }
    }
    utfil.close();
} // Slutt på skrivTilFil
} // Slutt på class BokAnalyse
```



UNIVERSITETET  
I OSLO

# EKSAMEN VÅREN 2005



# Oppgave 1 Kortsvarsoppgave (vekt 30%)

*Denne oppgaven består av flere små deloppgaver, som hver teller like mye i vurderingen.*

## 1a

Studer koden nedenfor:

---

```
class SkrivHei {
    public static void main (String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 2; j++) {
                i = i + 4;
                System.out.println("hei");
            }
        }
    }
}
```

---

Hvor mange ganger blir teksten hei skrevet ut når programmet SkrivHei kjøres?



## 1b

Hva skrives ut på skjermen når programmet nedenfor kjøres?

```
class EnkelRegning {  
    public static void main(String[] args) {  
        int a = 4;  
        int b = 9;  
        a = b;  
        b++;  
        System.out.println("a = " + a);  
        System.out.println("b = " + b);  
    }  
}
```

## 1c

Hva skrives ut på skjermen når programmet nedenfor kjøres?

```
class SkrivUt2{
    public static void main (String[] args){
        String s = "abcdefghijklm";
        String t = s.substring(0,3);
        System.out.println("t er nå:" + t);
        int lengde = s.length();
        int kvart = lengde/4;
        int halv = lengde/2;
        String u = s.substring(halv - kvart, halv + kvart);
        System.out.println("u er nå:" + u);
        if (s.indexOf("deF")>(-1)) {
            System.out.println("deF finnes");
        }
        else {
            System.out.println("deF finnes ikke");
        }
        if (s.endsWith("klm")){
            System.out.println("s ender med klm");
        }
    }
}
```

# 1d

Hva skrives ut på skjermen når programmet nedenfor kjøres?

```
class Beregninger{
    public static void main(String[] args){
        int x = 9;
        int y = 3;
        int z = x - y;
        System.out.println("verdien til z er " + z);
        if ((x-z) == y){
            System.out.println("like");
        }
        else {
            System.out.println("ulike");
        }
        if ((x > 8) || (y < 2)){
            System.out.println("sann");
        }
        else{
            System.out.println("usann");
        }
        x++;
        y += x;
        int d = --x + y--;
        System.out.println("d = " + d);
    }
}
```





**1e**

Hva skrives ut på skjermen når programmet nedenfor kjøres?

---

```
class WhileTest {  
    public static void main (String[] args) {  
        boolean fortsett = true;  
        int k = 3;  
        while (fortsett) {  
            k += 3;  
            fortsett = !fortsett;  
            System.out.println("k = " + k);  
        }  
    }  
}
```

---



## 1f

Studer kodelinjene nedenfor:

---

```
int[] a = new int[50];  
... // fyller inn tall i arrayen a  
int s = 1;  
for (int i = 0; i < a.length; ++i)  
    s *= a[i];  
}
```

---

Hva blir innholdet i variabelen `s` etter at denne koden er utført?



## 1g

Studer koden for klassen Person:

```
1 class Person [  
2     int navn;  
3     int alder;  
4     String adresse;  
5  
6     public Person (String n, int a, String adr) {  
7         navn = n;  
8         alder = a;  
9         adresse = adr;  
10    }  
11 ]
```

Finn og korriger alle feil i denne koden. Referer til linjenummer der du vil gjøre rettinger eller tillegg i koden.

# Oppgave 3a: DNA-sekvenser

- Gitt DNA-sekvensen AATGGATC.
- Denne sekvensen består av 8 symboler, hvorav 3 forekomster av A.
- Dermed er den relative frekvensen av A i sekvensen  $3/8 = 0,375$ .
- Skriv metoden

```
double[] symbolFrekvens(String sekvens) {  
    ...  
}
```

som tar en String som parameter og returnerer en array med 4 verdier (frekvensen til A, frekvensen til T, frekvensen til C, frekvensen til G).

```
double[] symbolFrekvens(String sekvens) {
    int antA = 0;
    int antT = 0;
    int antC = 0;
    int antG = 0;

    int lengde = sekvens.length();

    for (int i = 0; i < lengde; i++) {
        char b = sekvens.charAt(i);
        switch (b) {
            case 'A': antA++; break;
            case 'T': antT++; break;
            case 'C': antC++; break;
            case 'G': antG++; break;
        }
    }

    double[] frekvens = { (double) antA/lengde,
                          (double) antT/lengde,
                          (double) antC/lengde,
                          (double) antG/lengde };

    return frekvens;
}
```



## Oppgave 3b

- Lag en klasse DNAsekvens som inneholder følgende informasjon:
  - Navnet på sekvensen
  - Selve sekvensen
  - Lengden på sekvensen (antall symboler)
  - Frekvensen av A, T, C og G
- Lag en konstruktør i klassen slik at et nytt DNAsekvens objekt kan opprettes ved kodesetningen

```
DNAsekvens a = new DNAsekvens(navn, sekvens);
```

der argumentene navn og sekvens begge er av type String.



```
class DNasekvens {
    String navn;
    String sekvens;
    double[] frekvens;
    int lengde;

    DNasekvens(String navn, String sekvens) {
        this.navn = navn;
        this.sekvens = sekvens;
        lengde = sekvens.length();
        frekvens = symbolFrekvens(sekvens);
    }

    double[] symbolFrekvens(String sekvens) {
        // se forrige oppgave...
    }
}
```

# Oppgave 3c

- Eksempel på filen DNA.txt:

```
3  
AY1231 AAATCAGAAG  
AY5432 GGAATCCAGTAAAA  
AY3234 GGAGTCGATGA
```

Antall DNA-  
sekvenser i filen

Navnet på  
sekvensen

Selve  
sekvensen

- Lag en metode

```
DNasekvens[] lesSekvenserFraFil(String filnavn) {  
    ...  
}
```

som leser inn alle DNA-sekvensene i en array bestående av DNA-sekvens-objekter og returnerer denne.





```
DNasekvens[] lesSekvenserFraFil(String filnavn) {  
    In innfil = new In(filnavn);  
  
    int antall = innfil.InInt();  
    DNasekvens[] sekvensene = new DNasekvens[antall];  
  
    for (int i = 0; i < antall; i++) {  
        String navn = innfil.inWord();  
        String sekv = innfil.inWord();  
        sekvensene[i] = new DNasekvens(navn, sekv);  
    }  
    return sekvensene;  
}
```