

INF1000 – Uke 13

Gjennomgang av prøveeksamen fra høsten 2004

Oppgave 1

Er dette lovlige deklarasjoner (når de foretas inni en metode)?

- | JA | NEI | |
|-------------------------------------|-------------------------------------|--|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>int i;</code> |
| | | <code>int k = i;</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>int i, j = 4;</code> |
| | | <code>int k = j;</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>double[] x = new double[1];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>String[] s = new char[10];</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>String s = {"Arne", "Ole"};</code> |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | <code>int i = new int;</code> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | <code>char[] c = null;</code> |

Oppgave 2

Hvor mange ganger blir "INF1000" skrevet ut?

a)

```
for (int i=0; i<4; ++i) {
    System.out.println("INF1000");
}
```

Svar: **4 ganger**

b)

```
for (int i=1; i<4; i++) {
    for (int j=1; j<5; j++) {
        System.out.println("INF1000");
    }
}
```

Svar: **12 ganger**

c)

```
for (int i=0; i<4; i+=2) {
    for (int j=0; i+j<4; j+=i+1) {
        System.out.println("INF1000");
    }
}
```

Svar: **5 ganger**

i	j
0	0
0	1
0	2
0	3
2	0

Oppgave 3

Anta at følgende kodelinjer utføres:

	x	y	z	test
<code>int x = 22;</code>	22	-	-	-
<code>int y = x / 27;</code>	22	0	-	-
<code>int z = y * 27;</code>	22	0	0	-
<code>boolean test = (x == z y > 0);</code>	22	0	0	false

Hva er verdien til variabelen `test` rett etterpå?

Svar: **false**

Oppgave 4

Anta at følgende setninger utføres:

	i	j	k	verdi
<code>int k = 33;</code>	-	-	33	-
<code>int i = 2*k;</code>	66	-	33	-
<code>int j = k++;</code>	66	33	34	-
<code>if (k>j) j += 67;</code>	66	100	34	-
<code>k = k + j + i++;</code>	67	100	200	-
<code>int verdi = ++k;</code>	67	100	201	201

Hva er verdien til variabelen `verdi` like etter at setningene over er utført?

Svar:²⁰¹.....

Oppgave 5

Anta at følgende er deklart i en metode i et program:

```
int[] vekt = new int[100];
int sum = 0;
```

Anta at arrayen `vekt` fylles opp med tilfeldige verdier. Vil alternativene under alltid gi som resultat at `sum` inneholder summen av alle elementene i arrayen `vekt`?

- JA** **NEI**
- ```
int k = 0;
while (k < vekt.length) {
 sum = sum + vekt[k];
}
```
- JA**  **NEI**
- ```
for (int k = vekt.length; k > 0; k--) {
    sum += vekt[k-1];
}
```
- k øker ikke i løkka*

JA **NEI**

```
int k = 0;
do {
    k++;
    sum = sum + vekt[k];
} while (k < vekt.length);
```

k løper fra 1 til vekt.length når vi summerer

JA **NEI**

```
int k = 0;
do {
    sum = sum + vekt[k];
    k++;
} while (k < vekt.length);
```

JA **NEI**

```
for (k = 0; k < vekt.length; k++) {
    if (vekt[k] > 0) {
        sum += vekt[k];
    } else {
        sum -= vekt[k];
    }
}
```

k ikke deklart + feil oppdatering når vekt[k] < 0

JA **NEI**

```
int k = 0;
for (int i = 0; i < vekt.length; i++) {
    sum = vekt[k++];
}
```

Summerer ikke i for-løkka

JA **NEI**

```
for (int i = 0; i < vekt.length - 1; i += 2) {
    sum += vekt[i] + vekt[i+1];
}
```

JA **NEI**

```
boolean ferdig = false;
int k = 0;
while (!ferdig) {
    if (k > 99) {
        ferdig = true;
    } else {
        sum = sum + vekt[k++];
    }
}
```

Oppgave 6

Vi kan lage en sekvens med heltall på følgende måte: la de to første elementene i sekvensen begge være 1 og la deretter hvert element være summen av de to foregående. Den sekvensen vi da får starter slik: 1, 1, 2, 3, 5, 8, 13, og kalles Fibonacci-sekvensen.

Du skal nå lage en metode som beregner de n første elementene i Fibonacci-sekvensen og returnerer disse i en heltallsarray av lengde $n+1$ (det første elementet i arrayen, svarende til posisjon 0, skal ikke benyttes).

Posisjon k i arrayen som returneres skal inneholde det k 'te elementet i Fibonacci-sekvensen for $k=1, \dots, n$. Du kan anta at $n \geq 2$.

```
int[] fibonacci (int n) {
    int[] a = new int[n+1];
    a[1] = 1;
    a[2] = 1;
    for (int i=3; i<n+1; i++) {
        a[i] = a[i-1] + a[i-2];
    }
    return a;
}
```

Oppgave 7

Skriv ferdig metoden under, som teller opp hvor mange ganger tegnet i char-variabelen **tegn** forekommer i tekststrengen **tekst**, og returnerer dette antallet.

For eksempel skal kallet `antallForekomster ("Engelske", 'e')` returnere verdien 2. Hvis `tekst == null` så skal metoden returnere verdien 0.

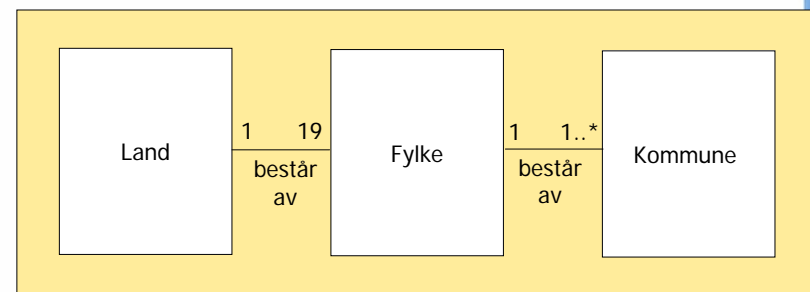
```
int antallForekomster (String tekst, char tegn) {
    if (tekst == null) return 0;

    int antall = 0;
    for (int i=0; i<tekst.length(); i++) {
        if (tekst.charAt(i) == tegn) antall++;
    }

    return antall;
}
```

Oppgave 8

Norge består av 19 fylker som hver består av et antall kommuner. Anta at vi modellerer dette ved hjelp av tre klasser **Land**, **Fylke** og **Kommune** slik at Norge representeres ved et objekt av klassen **Land**, hvert fylke er representert ved et objekt av klassen **Fylke** og hver kommune er representert ved et objekt av klassen **Kommune**. Lag et UML klassediagram som viser forholdet mellom de tre klassene **Land**, **Fylke** og **Kommune**, og få på riktige antall i hver ende av forholdene.



Oppgave 9

```
class StudentSystem {
    public static void main (String[] args) {
        Studieprogram inf =
            new Studieprogram("Informatikk");
        Studieprogram bio =
            new Studieprogram("Biologi");

        new Student("Kari Olsen", inf);
        new Student("Eva Larsen", inf);
        new Student("Per Jensen", bio);

        bio.liste();
        inf.liste();
    }
}

class Student {
    String navn;

    Student (String navn, Studieprogram sp) {
        this.navn = navn;
        sp.meldInnStudent(this);
    }

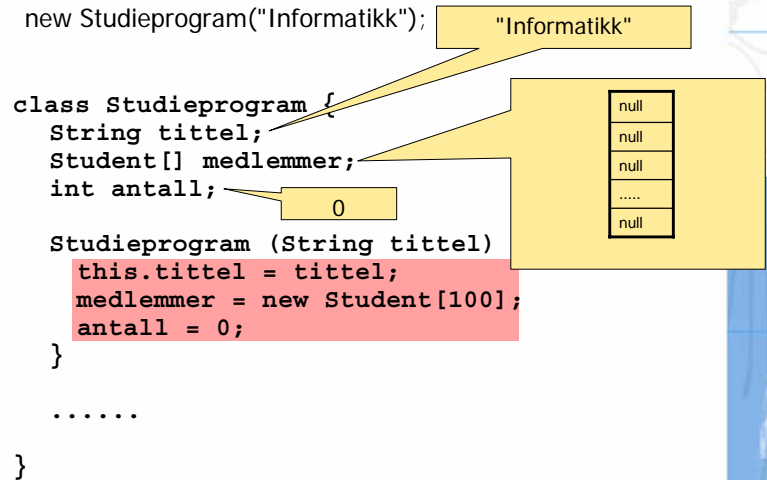
    String somTekst() {
        return navn;
    }
}
```

```
class Studieprogram {
    String tittel;
    Student[] medlemmer;
    int antall;

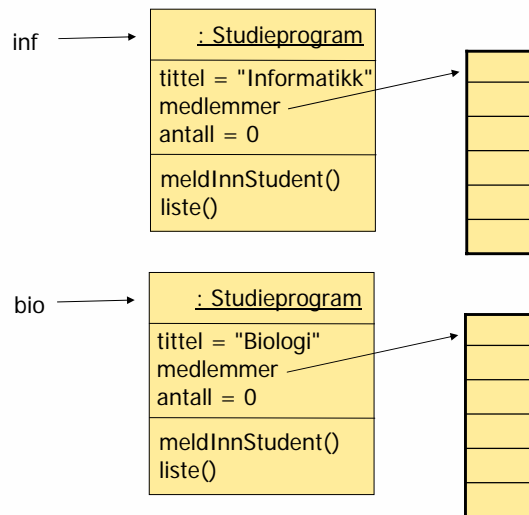
    Studieprogram (String tittel) {
        this.tittel = tittel;
        medlemmer = new Student[100];
        antall = 0;
    }

    void meldInnStudent(Student stud) {
        medlemmer[antall] = stud;
        antall++;
    }

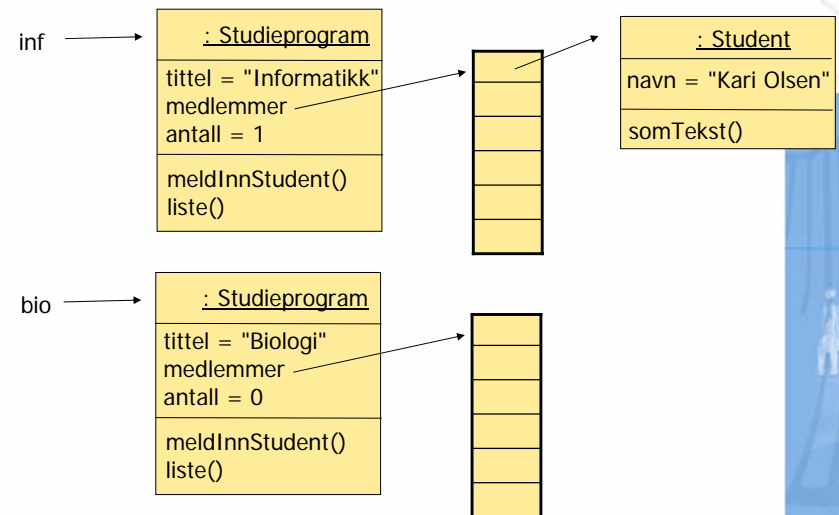
    void liste() {
        for (int i=0; i<antall; i++) {
            System.out.println(memlemmer[i].
                somTekst());
        }
    }
}
```



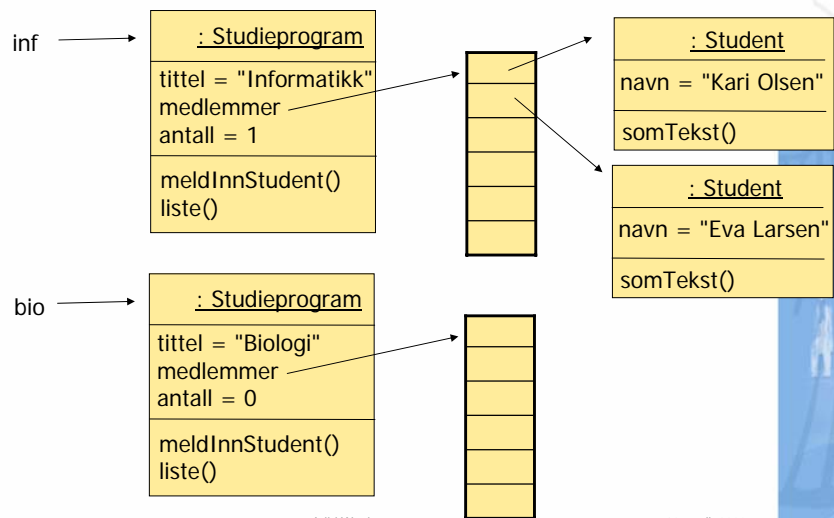
```
Studieprogram inf = new Studieprogram("Informatikk");
Studieprogram bio = new Studieprogram("Biologi");
```



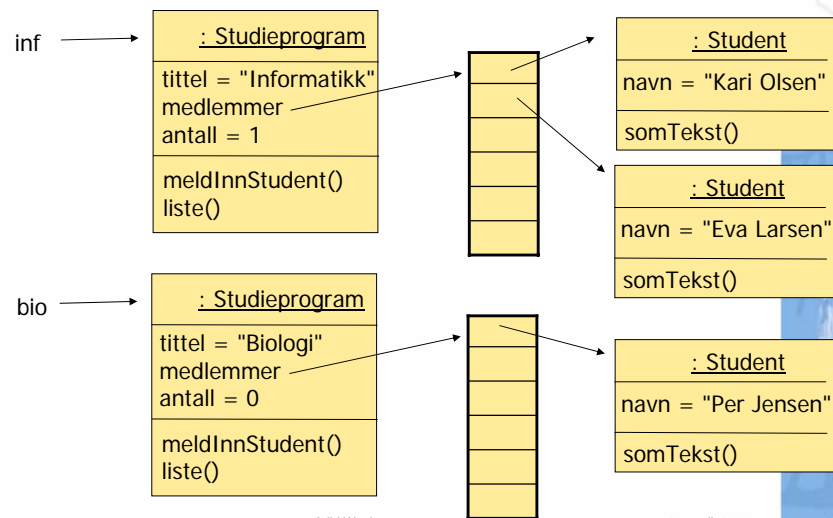
```
new Student ("Kari Olsen", inf);
```



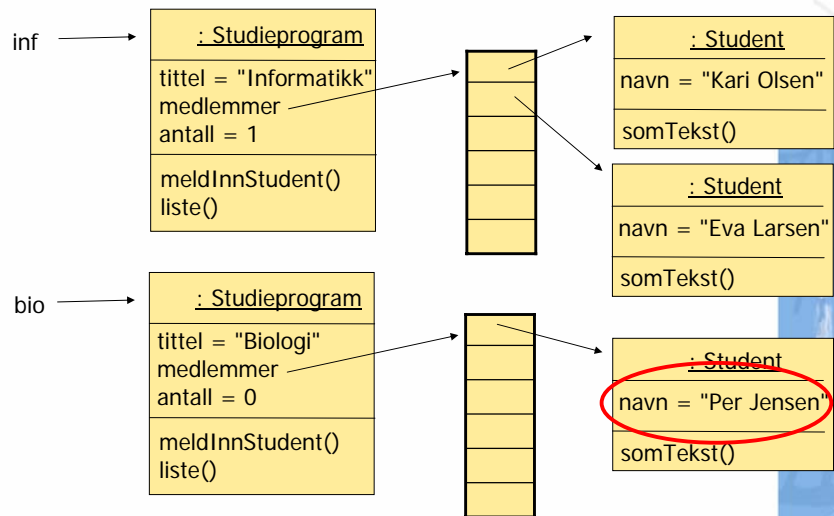
new Student ("Eva Larsen", inf);



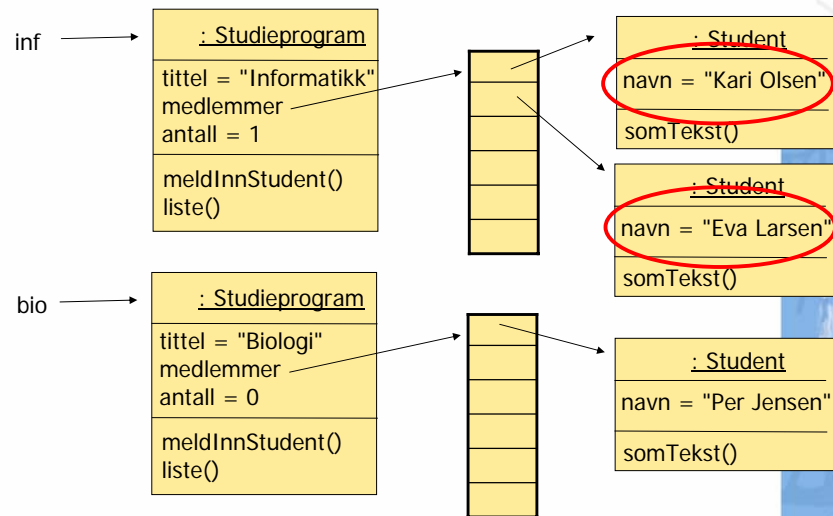
new Student ("Per Jensen", bio);



bio.liste();



inf.liste();



Svar:

Per Jensen
Kari Olsen
Eva Larsen

Oppgave 10

Anta at du har deklartert en HashMap:

```
HashMap cdSamling = new HashMap();
```

og at du legger inn informasjon om CD'ene dine i HashMapen, med platens tittel som nøkkel og artistnavnet som verdi (du antar at alle platene dine alle har ulik tittel). Et eksempel er

```
cdSamling.put("Not going under", "Maria Arredondo");
```

Skriv noen programsetninger som:

- først ber om og leser inn et artistnavn fra tastatur
- deretter går gjennom HashMapen og skriver ut titlene til alle platene du har registrert med denne artisten. Hint: husk at du vil ha tak i både nøklene og verdiene når du løper gjennom HashMapen. Det kan derfor lønne seg å lage en Iterator over nøklene (platetitlene).

Du kan anta at programsetningene plasseres slik i programmet at de har tilgang til HashMapen `cdSamling`.

```
In tast = new In();
System.out.print("Artistens navn: ");
String artistnavn = tast.inWord("\n");

Iterator it = cdSamling.keySet().iterator();
while (it.hasNext()) {
    String tittel = (String) it.next();
    String navn = (String) cdSamling.get(tittel);
    if (navn.equals(artistnavn)) {
        System.out.println(tittel);
    }
}
```

Moral når du løper gjennom en HashMap:

- Hvis du bare er ute etter verdien, kan du bruke `it.values().iterator()`
- Hvis du er ute etter både nøkkelen og verdien, kan du bruke `it.keySet().iterator()` for å finne nøkkelen, og `get()` for å finne verdien.

Oppgave 11

I programmet nedenfor skal du lage en konstruktør til klassen Student. Konstruktøren skal ha studentens navn som parameter og skal initiere objektvariabelen **navn**.

Du skal også skrive en objektmetode **void økPoeng(int poeng)** i klassen Student som øker antall studiepoeng med parameterens verdi.

I klassen StudentTest skal du så sørge for å få initiere arrayen **uioStud** med 32000 Student-objekter. Studentobjektene i arrayen **uioStud** skal ha hvert sitt studentnavn **Stud-1, Stud-2, Stud-3, ..., Stud-32000**. Dermed skal f.eks. **uioStud[252]** peke på et Student-objekt hvor studentens navn er satt lik **Stud-253**.

Til sist skal du sette antall studiepoeng til 30 for hver av studentene **Stud-1, ..., Stud-25000**.

```
class Student {
    String navn;
    int antallStudiepoeng = 0;
    // Her skal du skrive konstruktøren

    Student (String navn) {
        this.navn = navn;
    }

    // Her skal du skrive objektmetoden økPoeng

    void økPoeng (int poeng) {
        antallStudiepoeng += poeng;
    }
}
```

```
class StudentTest {
    Student[] uioStud = new Student[50000];

    public static void main (String[] args) {
        // Her skal du opprette 32000 Student-objekter med navn Stud-1,
        // Stud-2,...,Stud-32000 og legge dem inn i uioStud-arrayen
        StudentTest st = new StudentTest();
        for (int i=0; i<32000; i++) {
            st.uioStud[i] = new Student("Stud-" + (i+1));
        }
        // Her skal du øke antall studiepoeng med 30 for studentene
        // med studentnavn Stud-1, Stud-2, ..., Stud-25000.

        for (int i=0; i<25000; i++) {
            st.uioStud[i].økPoeng(30);
        }
    } // end main
}
```

Oppgave 12

Filen **SpamOrd.txt** inneholder endel *spam-ord* (atskilt av blanke tegn), dvs ord som du ikke forventer å finne i seriøse eposter til deg, men som ofte forekommer i useriøse eposter (såkalt spam-epost).

Du skal lage et program som leser **SpamOrd.txt** og som deretter leser filen **Epost.txt** og sjekker om denne inneholder noen av spam-ordene. Programmet skriver til slutt ut på skjermen om filen **Epost.txt** inneholdt noen spam-ord eller ikke (du trenger ikke å skrive ut hvilke eller hvor mange spam-ord filen inneholdt).

Du kan anta at **SpamOrd.txt** maksimalt inneholder 200 ord. I denne oppgaven kan du godt legge all programkoden inn i main-metoden.

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

class Spam {
    public static void main (String[] args) {
        HashMap h = new HashMap();

        In spamfil = new In("SpamOrd.txt");
        while (!spamfil.lastItem()) {
            h.put(spamfil.inWord(), null);
        }

        In epostfil = new In("Epost.txt");
        boolean funnet = false;
        while (!epostfil.lastItem() && !funnet) {
            if (h.containsKey(epostfil.inWord()) funnet = true;
        }

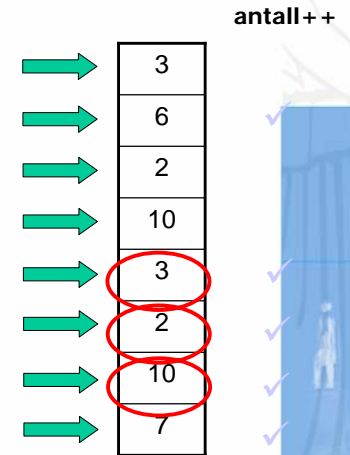
        if (funnet) System.out.println("Spam-ord !");
        else System.out.println("Ingen spam-ord");
    }
}
```

Oppgave 13 - Vanskelig

Skriv en metode som finner og skriver ut på skjerm hvor mange ulike verdier du har i en array og returnerer dette antallet.

Hint: løp gjennom alle elementene i arrayen, og for hvert element teller du det med (dvs øker antallet elementer med 1) kun dersom det er siste gang dette elementet forekommer i arrayen (det kan du sjekke med en ny løkke inni den ytre løkken).

Du kan anta at lengden av arrayen a er minst 1.



```
void finnAntallUlikeTall (int[] a) {
    int antall = 0;

    for (int i=0; i<a.length; i++) {
        int ai = a[i];
        boolean flere = false;

        int j = i+1;
        while (j < a.length && !flere) {
            if (a[j++] == ai) flere = true;
        }

        if (!flere) antall++;
    }

    System.out.println(antall);
}
```

Oppgave 14 - vanskelig

Et primtall er et heltall $k \geq 2$ som bare er delelig med 1 og seg selv. F.eks. er 5 et primtall fordi det ikke er delelig med noen av tallene 2, 3, 4 (med andre ord: vi kan ikke skrive 5 som et helt multiplum av noen av disse tallene), mens 4 ikke er et primtall fordi det er delelig med 2.

Det minste primtallet er 2. Hvis $k \geq 3$ er et heltall, så er k et primtall hvis det ikke er delelig med noen av tallene 2, 3, ..., $k-1$. For å sjekke om et tall er delelig med et annet i Java kan vi bruke operatoren `%`. Hvis a og b er to heltall, så vil $a \% b == 0$ hvis og bare hvis a er delelig med b .

Skriv ferdig metoden nedenfor som skriver ut på skjermen de n første primtallene. F.eks. skal kallet `skrivPrimtall(5)` skrive ut tallene 2, 3, 5, 7 og 11. Du kan anta at $n \geq 1$.


```
void skrivPrintall (int n) {
    System.out.println(2);

    int antall = 1;
    int k = 3;

    while (antall < n) {
        boolean erPrintall = true;
        for (int i=2; i<k; i++) {
            if (k % i == 0) erPrintall = false;
        }

        if (erPrintall) {
            System.out.println(k);
            antall++;
        }

        k++;
    }
}
```

