

## INF1000 eksamen V05

## Oppgave 1 - Kortsvarsoppgave

(a)

```
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?

Svar: 2 ganger

2007-04-30

2

## Oppgave 1 (b)

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);
    }
}
```

Svar: a = 9, b=10

2007-04-30

3

## Oppgave 1 (c)

```
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");
        }
    }
}
```

Hva skrives ut på skjermen?

Svar:

t er nå:abc

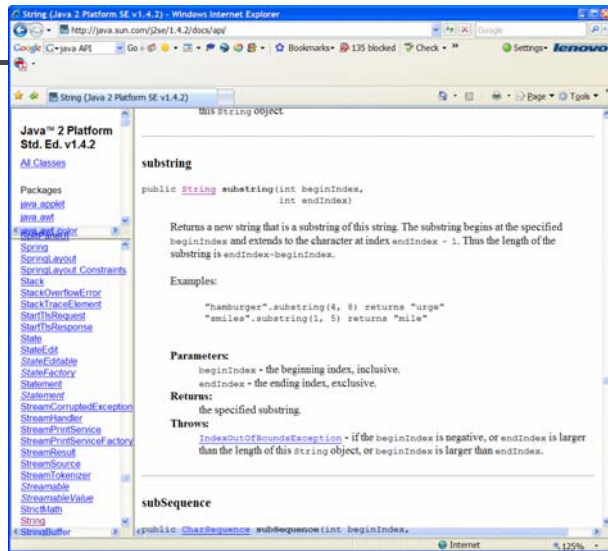
u er nå:defghi

deF finnes ikke

s ender med klm

2007-04-30

4



2007-04-30

5



## Oppg. 1 (d)

Hva skrives ut på skjermen?

Svar:  
verdien til z er 6  
like  
sann  
d = 22

```
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);
    }
}
```

2007-04-30

6



## Oppgave 1 (e)

Hva skrives ut på skjermen?

Svar:  
k=6

```
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);
        }
    }
}
```

2007-04-30

7



## Oppgave 1 (f)

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

Svar:  
produktet av alle verdiene i array a

```
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];
}
```

2007-04-30

8

## Oppgave 1 (g)

```
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; Svar:  
10    }  
11 ]
```

Finne og korriger alle feil i denne koden.

- Linje 1 og 11 feil bruk av parentes.
- Linje 2 navn er deklartert som en int, men i linje 7 blir navn satt lik en String.

2007-04-30

9

## Oppgave 2 – class Geografikunnskaper

```
import java.util.*;  
class Geografikunnskaper {  
    HashMap LandOversikt = new HashMap();  
  
    void leggTilLand(String navn, int innbyggertall) {  
        Land land = new Land(innbyggertall);  
        LandOversikt.put(navn, land);  
    }  
  
    void leggTilByILand(String navnLand, String navnBy,  
                        int innbyggertall, boolean hovedstad) {  
        boolean finnes = LandOversikt.containsKey(navnLand);  
        if (!finnes) {  
            Land land = new Land(0);  
            LandOversikt.put(navnLand, land);  
        }  
        Land nland = (Land) LandOversikt.get(navnLand);  
        nland.leggTilBy(navnBy, innbyggertall, hovedstad);  
    }  
}
```

2007-04-30

10

## Oppgave 2 – class Geografikunnskaper

```
public static void main(String args[]) {  
    Geografikunnskaper geo = new Geografikunnskaper();  
    geo.leggTilLand("Norge", 4525000);  
    geo.leggTilLand("Sverige", 9040000 );  
    geo.leggTilByILand("Norge", "Oslo", 530000, true);  
    geo.leggTilByILand("Norge", "Moss", 26000, false);  
    geo.leggTilByILand("Norge", "Narvik", 19000, false);  
    geo.leggTilByILand("Norge", "Hamar", 26500, false);  
    // Anta at programmet senere skal utvides slik at elever  
    // kan legge inn informasjon om nye land og byer ved bruk  
    // av meny og inntasting på skjermen. NB! Dette skal ikke  
    // lages nå.  
    // Her skal du sørge for at alle byer i alle land blir  
    // skrevet ut på skjermen.  
}  
} // Slutt på class Geografikunnskaper
```

2007-04-30

11

## Oppgave 2 – class Land

```
class Land{  
    int innbyggertall;  
    HashMap Byer = new HashMap();  
  
    Land(int innbyggertall) {  
        this.innbyggertall = innbyggertall;  
    }  
  
    void leggTilBy(String navn, int innbyggertall,  
                  boolean hovedstad) {  
        By by = new By(innbyggertall, hovedstad);  
        Byer.put(navn, by);  
    }  
  
    void skrivUtAlleLand() {  
        // Denne metoden skal du skrive i oppgave 2b  
    }  
} // Slutt på class Land
```

2007-04-30

12

## Oppgave 2 – class By

```
class By{
    int innbyggertall;
    boolean hovedstad;

    By(int innbyggertall, boolean hovedstad) {
        this.innbyggertall = innbyggertall;
        this.hovedstad = hovedstad;
    }

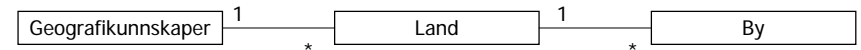
    void skrivUt() {
        // Denne metoden skal du skrive i oppgave 2b
    }
} // Slutt på class By
```

2007-04-30

13

## Oppgave 2 a

- Lag UML klassesdiagram for systemet

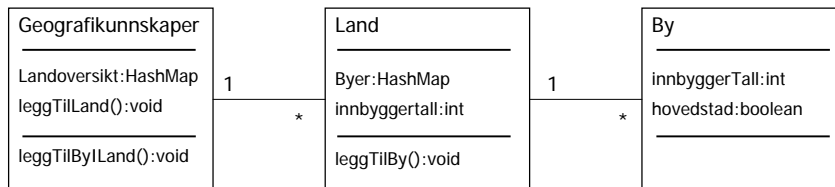


2007-04-30

14

## Oppgave 2 a

- Eller:



2007-04-30

15

## Oppgave 2b

- Lag en rutine som kan skrive ut:

```
Norge    4525000
*****
Oslo     530000
Moss     260000
Narvik   190000
Hamar    265000
*****
Sverige  9040000
```

- For land uten registrerte byer skrives kun landsinfo.

2007-04-30

16



## Oppgave 2 b – class By

```
void skrivUt(){
    System.out.println(innbyggertall);
}
```

2007-04-30

17



## Oppgave 2 b – class Land

```
void skrivUtAlleLand(){
    System.out.println(innbyggertall);
    System.out.println("*****");
    Iterator it = Byer.keySet().iterator();
    while (it.hasNext()){
        String nøkkel = (String) it.next();
        System.out.print(nøkkel + " ");
        By b = (By) Byer.get(nøkkel);
        b.skrivUt();
    }
}
```

2007-04-30

18



## Oppgave 2 b – class Geografikunnskaper

```
void skrivUt(){
    Iterator it = LandOversikt.keySet().iterator();
    while (it.hasNext()){
        String nøkkel = (String) it.next();
        System.out.println(nøkkel + " ");

        Land land = (Land) LandOversikt.get(nøkkel);
        land.skrivUtAlleLand();
        System.out.println("*****");
    }
}
```

2007-04-30

19



## Oppgave 2 b - main

```
geo.skrivUt();
```

2007-04-30

20

## Oppgave 3 – 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  $3/8 = 0,375$ .

## Oppgave 3 (a)

```
double[] symbolFrekvens(String sekvens){
    int a = 0;
    int t = 0;
    int c = 0;
    int g = 0;
    String SEKVENS = sekvens.toUpperCase();
    int n = SEKVENS.length();
    for (int i = 0; i < n; i++){
        char s = SEKVENS.charAt(i);
        switch (s) {
            case 'A': a++;
                       break;
            case 'T': t++;
                       break;
            case 'C': c++;
                       break;
            case 'G': g++;
        }
    }
    double[] frekvens = {(double) a/n, (double) t/n,
                        (double) c/n, (double) g/n};
    return frekvens;
}
```

## Oppgave 3 (b)

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.

## Oppgave 3 (b)

```
class DNasekvens{
    String navn;
    String sekvens;
    double[] frek = new double[4];
    int lengde;

    double[] symbolFrekvens(String sekvens){
        // SE OPPGAVE 3A
    }

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



## Oppgave 3 (c)

Eksempel på filen DNA.txt.:

```
137
AY1231 AAATCAGAAG
AY5432 GGAATCCAGTAAAA
AY3234 GGAGTCGATGA
AY3423 GGAGTCGATCCCAGA
...
```

Lag en metode

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

som leser inn alle DNA-sekvensene i en array bestående av **DNasekvens-objekter**. Denne arrayen returneres fra metoden.



## Oppgave 3 (c)

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