



Eksamen høsten 2003

Den store bøygen i INF1000 er de obligatoriske oppgavene.

Hovedformålet med eksamen er å skille ut de som har skjønt litt om programmering fra de som ikke har skjønt noe.

For de som har løst de obligatoriske oppgavene uten mye hjelp, bør eksamen være ganske enkel.

2004

Oppgave 1.1

Hvilke utsagn er riktige om en *variabel deklarerert i en objektmetode*?

- Før vi foretar en tilordning til variabelen har den ingen verdi.
- Den kan ha en aksessmodifikator (f eks `private` eller `public`).
- Andre objektmetoder i samme klasse har tilgang (aksess) til variabelen.
- Objektmetoder i andre klasser har tilgang (aksess) til den via prikk-notasjon.

2004

NB! Les oppgaveteksten *nøye!*

... i en **objektmetode**?

Variable i metoder er alltid utilgjengelige utenfra. Derfor er det også uaktuelt med aksessmodifikator.

Som alle variable er metodevariable uten verdi i starten. De kan initieres ved deklarasjonen:

```
int x = 0;
```

eller ved en tilordning senere.

2004

Svaret på oppgave 1.1 er altså

- Før vi foretar en tilordning til variabelen har den ingen verdi.
- Den kan ha en aksessmodifikator (f eks `private` eller `public`).
- Andre objektmetoder i samme klasse har tilgang (aksess) til variabelen.
- Objektmetoder i andre klasser har tilgang (aksess) til den via prikk-notasjon.

2004

Oppgave 1.2

Hvilke utsagn er riktige om en *objektvariabel*?

- Før vi foretar en tilordning har den ingen verdi.
- Den kan ha en aksessmodifikator (f eks `private` eller `public`).
- Objektmetoder i samme klasse har tilgang (aksess) til variabelen.
- Objektmetoder i andre klasser har tilgang (aksess) til den via prikk-notasjon dersom variabelen er deklartert som `public`.

2004

Vi har følgende aksessmodifikatorer:

- `private` skjult for alle andre klasser
- `protected` skjult for alle andre klasser (unntatt *subklasser*)
 - kun tilgjengelig for klasser i samme *pakke*
- `public` åpen for alle

Dere vet ikke hva pakker og subklasser er; det kommer i INF1010. Dere trenger bare å vite:

- `private` skjult
- `public` åpen

2004

Svaret på oppgave 1.2 er da:

- Før vi foretar en tilordning har den ingen verdi.
- Den kan ha en aksessmodifikator (f eks `private` eller `public`).
- Objektmetoder i samme klasse har tilgang (aksess) til variabelen.
- Objektmetoder i andre klasser har tilgang (aksess) til den via prikk-notasjon dersom variabelen er deklartert som `public`.

2004

Oppgave 1.7

Anta at vi har et program hvor en av klassene blant annet har følgende objektvariabeldeklarasjon:

```
HashMap personer = new HashMap();
```

Klassen inneholder blant annet metoder for å legge inn objekter av klassen `Person` (med en passende nøkkel, f eks `personnummer`) i `HashMap`-en og for å løpe gjennom alle `Person`-objektene i `HashMap`-en. Sistnevnte metode, som skal kalle på en metode `SkrivUt()` i hvert av objektene i `HashMap`-en, ser slik ut:

```
void skrivAlle () {  
    Iterator liste = personer.values().iterator();  
    while (liste.hasNext()) {  
        ...  
        b.skrivUt();  
    }  
}
```

Innholdet i `while`-løkken ovenfor er ikke ferdig utfyllt. Hvilke(n) av følgende alternativer kan vi erstatte ... med slik at metoden virker slik den skal?

- `Person b = it.next();`
- `Person b = (Person)it.next();`
- `Person b = liste.next();`
- `Person b = (Person)liste.next();`
- `Bil b = (Bil)it.next();`
- Ingen av alternativene ovenfor

2004

NB! En del spørsmål vil være formet slik at de sjekker forståelse. Målet er at de som kopierer blindt fra læreboken eller lysark, skal tabbe seg ut.

De fleste eksemplene kaller iteratoren for it men i dette eksemplet heter den liste.

Det er mange eksempler med Bil i læreboken, men de har ingenting med dette eksemplet å gjøre.

Derimot demonstrerer eksemplene at man må typekonvertere når man bruker en iterator.

2004

Svaret på oppgave 1.7 blir da:

- Person b = it.next();
- Person b = (Person)it.next();
- Person b = liste.next();
- Person b = (Person)liste.next();
- Bil b = (Bil)it.next();
- Ingen av alternativene ovenfor

2004

Oppgave 1.12

Anta at følgende program utføres:

```
class Studentregister {
    public static void main (String[] arg) {
        Student s = new Student("Ole", "Karl Johans gt 1");
        Student p = new Student("Marit", "Karl Johans gt 2");
        System.out.println(s.fåNavn()+" og "+p.fåNavn());
    }
}

class Student {
    String navn = "Grete";
    String adresse = "Blindernveien 3";
}

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

String fåNavn () {return navn;}
}
```

2004

Hva blir utskriften på skjermen?

- Grete og Grete
- Ole og Ole
- Marit og Marit
- navn og navn
- Ole og Marit
- s.fåNavn() og s.fåNavn()
- Marit og Ole
- Ingen av alternativene over

2004

Vurderinger

Det er ingen klassevariable (angitt med static) i denne oppgaven, kun vanlige objektvariable.

Deklarasjonene

```
String navn = "Grete";  
String adresse = "Blindernveien 3";
```

utføres først hver gang et nytt objekt lages, men overskrives så av det som skjer i konstruktøren:

```
Student (String navn, String adresse) {  
    this.navn = navn;  
    this.adresse = adresse;  
}
```

Metoden fåNavn er en vanlig objektmetode og utføres «inne i» det objektet som angis i kallet:

```
s.fåNavn() og p.fåNavn()
```

2004

Det riktige svaret på oppgave 1.12 blir da:

- Grete og Grete
- Ole og Ole
- Marit og Marit
- navn og navn
- Ole og Marit
- s.fåNavn() og s.fåNavn()
- Marit og Ole
- Ingen av alternativene over

2004

Oppgave 2.2

Tenk deg at du skal lage et datasystem for en idrettsklubb. Systemet skal inneholde diverse informasjon om medlemmene i klubben. For hvert medlem skal systemet inneholde opplysninger om: navn, adresse, fødselsnummer (11 siffer), telefonnummer, statsborgerskap, yrke og sivil stand (ugift/gift/samboer). Kan et slikt system tas i bruk uten konsesjon?

- Ja
- Nei, fordi følgende paragrafer i personvernet er relevante (har du ikke paragrafnumrene, kan du si hvilke problemer de relevante paragrafene omhandler):

2004

De viktigste prinsippene ved slike vurderinger er:

§8 Det skal være saklig grunn for registeret.

§9 Sensitive opplysninger kun hvis den registrerte samtykker. §2 forteller hva som er sensitivt.

§19 Den registrerte skal vite om registeret.

§33 Man må ha konsesjon når det lagres sensitive opplysninger.

I dette registeret har vi:

- Fødesels- og personnummeret regnes som sensitivt.
- Statsborgerskap, yrke og sivilstand er muligens sensitive opplysninger.

(De er neppe heller aktuelle for et idrettslag.)

Konklusjon

Konsesjon er nødvendig pga §8, §9 og §33.

2004

```

void ordreløkke () {
    boolean fortsatt = true;
    while (fortsett) {
        switch (lesOrdre()) {
            case 1: leggInnSted(); break;
            case 2: fjernSted(); break;
            case 3: lagOversikt(); break;
            case 4: fortsatt = false;
        }
    }

    void leggInnSted () {
        // Les navn, nettadresse og beskrivelse fra terminal,
        // opprett nytt Nettsted-objekt og legg det inn i
        // HashMap-en 'nettsteder' med adressen som nøkkel.
    }

    void fjernSted() {
        // Les en tekststreng fra terminal. Gå så gjennom alle
        // nøklene i HashMap-en 'nettsteder' for å finne alle de
        // nettstedene med nettadresse som enten inneholder
        // tekststrengen eller er identisk med tekststrengen.
        // For hvert slikt nettsted skal programmet spørre
        // brukeren om det skal slettes, og hvis brukeren svarer
        // ja, så fjernes nettstedet fra HashMap-en 'nettsteder'.
        // Les f. eks tekststrengen "ufo.no", så skal programmet
        // i tur og orden presentere for brukeren hvert nettsted
        // med "ufo.no" i sin nettadresse og spørre om
        // nettstedet skal fjernes fra registret.
    }
}

```

2004

Oppgave 2.4 A

Gjør ferdig følgende:

```

void leggInnSted () {
    // Les navn, nettadresse og beskrivelse fra terminal,
    // opprett nytt Nettsted-objekt og legg det inn i
    // HashMap-en 'nettsteder' med adressen som nøkkel.

    System.out.println("Navn: ");
    String navn = tastatur.inWord("\n");
    System.out.println("Adresse: ");
    String adresse = tastatur.inWord("\n");
    System.out.println("Beskrivelse: ");
    String beskr = tastatur.inWord("\n");
}

```

2004

Oppgave 2.4

Gitt følgende program:

```

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

class NettstedProg {
    public static void main (String[] arg) {
        NettstedOversikt no = new NettstedOversikt();
        no.ordreløkke();
    }
}

class NettstedOversikt {
    In tastatur = new In();
    HashMap nettsteder = new HashMap();

    int lesOrdre () {
        int k = 0;
        while (k < 1 || k > 4) {
            System.out.println("1: Legg inn nytt nettsted");
            System.out.println("2: Fjern nettsted");
            System.out.println("3: Lag oversikt over nettsteder");
            System.out.println("4: Avslutt");
            k = tastatur.inInt();
        }
        return k;
    }
}

```

2004

```

void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.
}

class Nettsted {
    private String navn;
    private String nettadresse;
    private String beskrivelse;

    Nettsted (String navn, String nettadresse,
              String beskrivelse) {
        this.navn = navn;
        this.nettadresse = nettadresse;
        this.beskrivelse = beskrivelse;
    }

    void skrivUt () {
        System.out.println("Navn: " +navn);
        System.out.println("Nettadresse: " +nettadresse);
        System.out.println("Beskrivelse: " +beskrivelse);
    }
}

```

2004

Oppgave 2.4 C

Gjør ferdig følgende:

```
void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.
}
```

2004

Oppgave 2.4 B

Gjør ferdig følgende:

```
void fjernSted() {
    // Les en tekststreng fra terminal. Gå så gjennom alle
    // nøklene i HashMap-en 'nettsteder' for å finne alle de
    // nettstedene med nettadresse som enten inneholder
    // tekststrengen eller er identisk med tekststrengen.
    // For hvert slikt nettsted skal programmet spørre
    // brukeren om det skal slettes, og hvis brukeren svarer
    // ja, så fjernes nettstedet fra HashMap-en 'nettsteder'.
    // Les f eks tekststrengen "uio.no", så skal programmet
    // i tur og orden presentere for brukeren hvert nettsted
    // med "uio.no" i sin nettadresse og spørre om
    // nettstedet skal fjernes fra registeret.
    System.out.println("Adresse: ");
    String adresse = tastatur.inWord("\n");
}
}
```

2004

Denne bør være triviell om man har skjønt HashMap.

```
void leggInnSted () {
    // Les navn, nettadresse og beskrivelse fra terminal,
    // opprett nytt Nettsted-objekt og legg det inn i
    // HashMap-en 'nettsteder' med adressen som nøkkel.
    System.out.println("Navn: ");
    String navn = tastatur.inWord("\n");
    System.out.println("Adresse: ");
    String adresse = tastatur.inWord("\n");
    System.out.println("Beskrivelse: ");
    String beskr = tastatur.inWord("\n");
    nettsteder.put(adresse,
        new Nettsted(navn, adresse, beskr));
}
}
```

2004

Denne bør også være triviell utifra eksempler gitt i kurset:

```
void lagOversikt () {
    // Løp gjennom alle nettstedene som er lagret og skriv
    // ut innholdet av hver dem.
    Iterator it = nettsteder.values().iterator();
    while (it.hasNext()) {
        Nettsted ns = (Nettsted)it.next();
        ns.skrivUt();
    }
}
}
```

2004

Løsningen blir:

```
void fjernSted() {
    // Les en tekststreng fra terminal. Gå så gjennom alle
    // nøktene i HashMap-en 'nettsteder' for å finne alle de
    // nettstedene med nettadresse som enten inneholder
    // tekststreng eller er identisk med tekststreng.
    // For hvert slikt nettsted skal programmet spørre
    // brukeren om det skal slettes, og hvis brukeren svarer
    // ja, så fjernes nettstedet fra HashMap-en 'nettsteder'.
    // Les f. eks tekststreng "uio.no", så skal programmet
    // i tur og orden presentere for brukeren hvert nettsted
    // med "uio.no" i sin nettadresse og spørre om
    // nettstedet skal fjernes fra registeret.
    System.out.print("Adresse: ");
    String adresse = tastatur.innord("\n");
    Iterator it = nettsteder.keySet().iterator();
    while (it.hasNext()) {
        String adr = (String)it.next();
        if (adr.indexOf(adresse) >= 0) {
            System.out.print("Skal "+adr+" fjernes? ");
            String svar = tastatur.innord();
            if (svar.charAt(0) == 'j' || svar.charAt(0) == 'J')
                it.remove();
        }
    }
}
```

2004

Første spørsmål

Det første spørsmålet man stiller seg når man skal programmere, er:

«Hva skal gjøres?»

Svar: Spør brukeren om en nettadresse. Gå gjennom alle nettsteder og fjern de som matcher.

Andre spørsmål

Neste spørsmål er:

«Hvordan skal jeg få det til?»

- Be brukeren om en nettadresse: Trivielt.
- Gå gjennom alle nettstedene: Vanlig opplegg med en Iterator.
- Hvordan sjekke hvilke som matcher: Med String-metoden indexOf.
- Hvordan fjerne en post i HashMap-en: Med metoden remove.

(De to siste har ikke vært forelest, men er beskrevet i læreboken.)

2004

Dette er en oppgave som ikke er helt triviell. Man må sjekke to og to verdier mot med hverandre og så bytte dem om hvis de står i gal rekkefølge. Løsningen blir da:

```
void SkrivUtSortert2 (int a, int b, int c) {
    int tmp;

    if (a > b) {
        // Bytt om a og b:
        tmp = a; a = b; b = tmp;
    }
    // Nå vet vi at a <= b.

    if (b > c) {
        // Bytt om b og c:
        tmp = b; b = c; c = tmp;
    }
    // Nå vet vi at a <= c && b <= c.

    if (a > b) {
        // Bytt om a og b:
        tmp = a; a = b; b = tmp;
    }

    System.out.println(a);
    System.out.println(b);
    System.out.println(c);
}
```

2004

Oppgave 2.5

(Denne oppgaven bør tas til slutt.) Metoden nedenfor skal – uten bruk av array – skrive ut verdiene til de tre parametrene på hver sin linje i stigende rekkefølge. Metoden skal først bytte om på verdiene i a, b og c slik at $a \leq b \leq c$, og deretter skrives innholdet av a, b og c ut (i den rekkefølgen, og på hver sin linje). Fyll inn innholdet i metoden:

```
void SkrivUtSortert2 (int a, int b, int c) {
    }
}
```

2004

Andre tips

Hjelpemidler

Det er lov å ta med alle trykte og skrevne hjelpemidler til eksamen.

Tidsbruk

Fordel tiden jevnt. Om en oppgave teller 25%, bør dere jobbe 45 minutter med den og så gå over til en annen.

Er noe uklart i oppgaven?

Vi forelesere vil gå «trøsterunde» i lokalene etter ca 1 time. Da er det anledning til å spørre om det du måtte være usikker på.

2004