

INF1000: noen avsluttende ord

Arne og Fredrik



Pensum

- Det som er gjennomgått på forelesningene
- Stoff som er behandlet i oppgaver/obliger
- Notat om Informasjonsteknologi, vitenskap og samfunnsmessige virkninger



Oversikt (for deg som ikke er helt sikker på å få A)


- Pensumoversikt i store trekk
 - variabler, typer og tilordning
 - inn- og ut-lesning
 - løkker og if
 - tekster
 - metoder (parametere og retur av verdi)
 - klasser og objekter
 - mengder (array og HashMap)
 - UML
 - IKT og samfunn, Lov om behandling av persondata
- Sjekkliste
- Vanlige feil på eksamen
- Eksamenstips



variabler, typer og tilordning

- En enkelvariabel beskriver én ting/én egenskap
- Har tre (fem) egenskaper:
 - Et navn
 - En type
 - En verdi
 - (+ en nummerert plass /adresse i lageret)
 - (+ en størrelse, antall bit)

```
int i ;  
i = 4;  
i = i+1;
```



The diagram shows a light blue arrow pointing from the code to a yellow-bordered box containing the letter 'i' and the number '5'.

Sjekkliste: hva du absolutt bør kunne

- Bruk av klasser
 - Hvilke begreper som er naturlige å representere som klasser i et konkret tilfelle (NB: ofte flere "riktige" svar)
 - hvordan lage konstruktører og konstruktørens rolle
 - hvordan lage objekter av klasser
 - forskjellen på klassemetoder (-variable) og objektmetoder (-variable)
 - Hvordan (og når en kan) aksessere variable og metoder i andre objekter/klasser.

OMS 2007

27-Nov-07

OMS 2007

Sjekkliste *forts.*

- Lesing av tall/tekst/blanding av tall og tekst fra fil
 - Vite hvordan man åpner en fil for lesing
 - Vite hvordan man lager en løkke for å lese fra fil
 - Beherske bruk av inInt(), inWord(), inWord("\n"), osv
 - Beherske bruk av hasNext(), lastItem() og endOfFile()
- Skrivning av tall/tekst/blanding av tall og tekst til fil
 - Vite hvordan man åpner og lukker en fil for skriving
 - Vite hvordan man lager en løkke for å skrive til fil
 - Beherske bruk av out() og outln()

OMS 2007

27-Nov-07

OMS 2007

litt mer om: inn- og ut-lesning

- Bruk easyIO
 - kan bruke standard Java IO, men ingen pluss ved sensur
 - Behøver ikke ta med 'ekstra' som bare er for å rette opp problemer i easyIO - nå forøvrig rettet.
 - (som inLine() på 'slutten av' hver linje i fila)
- Husk alltid å lukke en fil:
- Les en fil:

OMS

```
void lesAbonnenter(String fil){
    In abon = new In(fil);

    while (abon.hasNext()) { // mulig : (!abon.lastItem())
        String fNum = abon.inWord();
        lisensBetalere.put(fNum, "Ikke brukt");
    }
    abon.close();
} // end lesAbonnenter
```

27-Nov-07

OMS 2007

Sjekkliste *forts.*

- Bruk av arrayer
 - Hvordan deklarere og opprette en array
 - Hvordan legge inn verdier i en array
 - Hvordan ta ut verdier av en array
 - Hvordan gå gjennom alle verdier i en array
- Bruk av HashMaper
 - Hvordan deklarere og opprette en HashMap.
 - Hvordan legge nøkkel/verdi inn i en HashMap
 - Hvordan ta ut nøkkel/verdi av en HashMap
 - Hvordan gå gjennom alle nøkler/verdier i en HashMap

OMS 2007

27-Nov-07

OMS 2007

Sjekkliste *forts.*

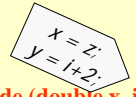
- Bruk av metoder
 - Hvordan kalle på en metode
 - Hvordan lage en metode
 - uten parametre/med parametre
 - uten returverdi/med returverdi
 - Bruk metoder for å
 - splitte opp koden i mer håndterlige biter
 - lette skrivingen av programmet
 - lette lesingen av programmet
 - lette feilretting av programmet

Verdien til parametrene kopieres over til metoden

- Når vi kaller metoden (bruker navnet i en annen metode), så overføres verdiene til de parameterne som ble brukt i kallet slik:

```
public static void main (String[] args) {
    A aa = new A()
    int i = 17;
    double z = 3.14;
    int j = aa.minMetode(z, i +2);
}

class A{
    int minMetode (double x, int y)  {
        // nå kan x og y brukes med de verdier de fikk i kallet
        x = x *2;
        return ((int) x) + y;
    }
}
```



De verdiene som ble brukt ved kallet, **blir kopiert over i parameterene før setningene i metoden blir utført.**

Sjekkliste *forts.*

- Forskjellen mellom elementære typer og klasser
 - elementære typer: int, double, char, boolean,
 - Kan ikke lage objekter av dem
 - klasser: String, HashMap, etc + våre egendefinerte klasser
 - Kan lage objekter av dem

Sjekkliste *forts.*

IT og samfunn

- Lov om behandling av persondata (som fremstilt på forelesningene)
- Er det sider ved det edb-baserte registeret du prøver å lage som kan tenkes å rammes av personvernlovgivingen?

En enkel klasse

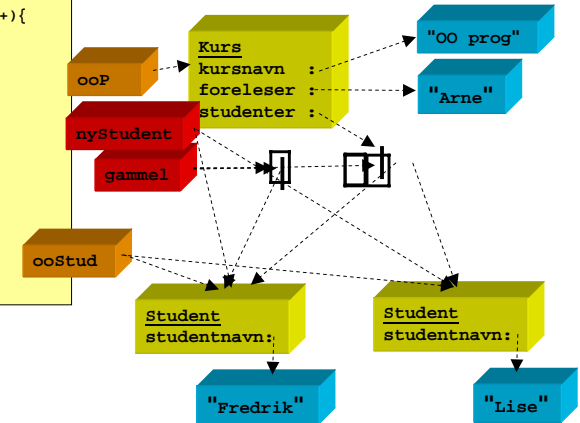
- Klasse for et kurs
- Har navnet på foreleser
- En array med studenter

- Vi skal se på
 - Konstruktøren
 - Objekter
 - Metoder

OMS 2007

```
class Kurs {
    String kursnavn;
    String foreleser;
    Student[] studenter;
    Kurs(String kursnavn, String foreleser){
        this.kursnavn = kursnavn;
        this.foreleser = foreleser;
        this.studenter = new Student[0];
    }
    void leggTilStudent(Student nyStudent){
        Student[] gammel = studenter;
        studenter = new Student[gammel.length+1];
        for(int i=0; i<gammel.length; i++){
            studenter[i] = gammel[i];
        }
        studenter[studenter.length-1] = nyStudent;
    }
}
class Student{
    String studentnavn;
    Student(String studentnavn){
        this.studentnavn = studentnavn;
    }
}
```

```
class Program{
    public static void main(String[] args){
        Kurs ooP = new Kurs("OO prog", "Arne");
        Student ooStud = null;
        ooStud = new Student("Fredrik");
        ooP.leggTilStudent(ooStud);
        ooStud = new Student("Lise");
        ooP.leggTilStudent(ooStud);
    }
}
```



```
class Kurs {
    < ... >
    boolean erStudent(String navn){
        for(int i=0; i<studenter.length; i++){
            if(studenter[i].studentnavn.equals( navn )){
                return true;
            }
        }
        return false;
    }
}
```

OMS 2007

```
class Program{
    public static void main(String[] args){
        < ... >
        if(ooP.erStudent("Lise")){
            System.out.println("Funnet!");
        }
    }
}
```

OMS 2007



Eksamenstips

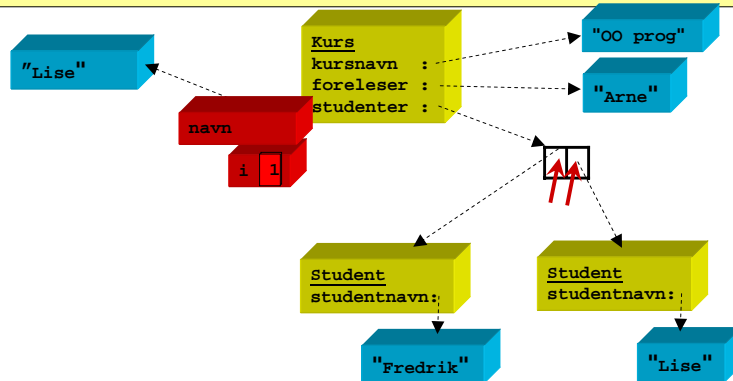
- Prøv å svare noe (fornuftig) på **alle** oppgavene!
- Ikke gruble for lenge etter "den beste" løsningen. Kommer du på en bedre løsning senere, beskriv den kort.
- Ved behov kan du gjøre forutsetninger som ikke er i strid med oppgaven og sunn fornuft.
 - Du kan forutsette at et filnavn er kjent hvis ikke annet står
 - Du kan *ikke* forutsette at max antall personer er 10 dersom det ikke står noe om det i oppgaven
 - Er du i tvil, så spør oss når vi går rundt på eksamen ca kl 10.

OMS 2007

27-Nov-07

OMS 2007

```
class Kurs {
  < ... >
  boolean erStudent(String navn){
    for(int i=0; i<studenter.length; i++){
      if(studenter[i].studentnavn.equals( navn )){
        return true;
      }
    }
    return false;
  }
}
```



Eksamenstips *forts.*

- Inf 1000 er et kurs i objektorientert programmering, **ikke** primært i Java:
 - Små syntaktiske feil, småfeil i metodenavn, osv spiller mindre rolle
 - Ville programmet ditt vært et nyttig første forsøk å utsette kompilatoren for?
- Java-konstruksjoner som ikke er forelest eller som ikke har vært tatt opp i oppgaver vil dere ikke få behov for på eksamen.
- En viss personlig frihet i hvordan man tegner UML-diagrammer (klasse/objekt-diagrammer) er tillatt.
 - Men meningen må komme klart fram! Ikke slurv når du tegner dem, det straffer seg siden når du skal bruke dem.

OMS 2007

27-Nov-07

OMS 2007



Eksempler på oppgave-formuleringer

- Ta utgangspunkt i et uferdig program og utvide med nye klasser, variable og metoder.
- Lage et program "fra scratch".
- Drøfte fordeler og ulemper med ulike løsninger
- Drøfte samfunnsmessige virkninger (f.eks. vil det være lov å sette et bestemt program i drift?)
- Hva skjer når denne løkka kjører? Hva er verdien av 'resultat' ?
- Flervalgsoppgaver med null, en eller flere riktige svar.

OMS 2007

27-Nov-07

OMS 2007

Eksamenstips: Se på prøveeksamen

- 1) 1 hvor stor (2p)
- 2) 10 ja/nei (20p)
- 3) 4 hvor mange (16p)
- 4) Verdi etter ++ etc. (4p)
- 5) Skriv enkel metode (20p)
- 6) Skriv en metode (20p)
- 7) Finn ut hva program gjør (10p)
- 8) Forstå hva et program gjør, noe vanskeligere (15p)
- 9) UML-oppgave (20p)
- 10) Det 'store' uspesifiserte programmet (50p)
- 11) Forstå et ganske vanskelig program (15p)
- 12) Skriv en litt vanskelig metode (30)
- 13) Lov om personopplysninger (20p)

Vanlige feil vi ser

- Glemmer å initialisere en variabel /gi den en verdi
 - Tror at den inneholder 'riktig' verdi fordi den har et 'riktig' navn
- Returnerer ikke en verdi fra en metode som krever det
- Greier ikke å nytte parametrene i en metode.
 - deklarerer f.eks. feilaktig parametere om igjen inne i metoden
- Når er det en peker og når er det verdien direkte
 - Peker: objekter av egne klasser og f.eks HashMap og String
 - Direkte verdi: int, long, double, boolean (basale typer)
 - array er spesiell:
 - Basaltyper (int, double) : Peker til et array-objekt med direkte verdier
 - Objekter : Peker til et array-objekt som igjen har pekere til objektene
- Programmet henger ikke sammen
 - Lager metoder som aldri blir kalt
 - Sier ikke **new** på egne klasser

Hvor foregår eksamen?

Haslehallen - tennissenter (Hasle stasjon på rute 5: Vestli-banen).

De med spesielle ordninger (Astma,...) sitter på Blindern - se hjemmesida.

Tid: 0900 - 1200



Noen praktiske ting rundt eksamen

- Husk å ta med legitimasjon og gyldig semesterkort på eksamen
- Ta med: Læreboka, obliger og notater
- Foreleserne går en runde som starter ca kl 1000. Da har dere mulighet til å spørre om ting som dere oppfatter som uklare i oppgaven o.l. Vi kan dessverre ikke svare på spørsmål av typen "ser dette riktig ut?" eller "ville du ha gjort det slik?"
- Bruk hele eksamenstiden ! Bli du sliten så ta heller en lang luftetur.
- Sensuren blir offentliggjort ca. 4-5 uker etter eksamen. Se forøvrig studWeb.



Fortsatt noe du lurer på?

Mange av dere har sikkert **fortsatt** endel spørsmål før eksamen. Er du en av dem?

Ta kontakt! Er det korte spørsmål kan vi gjerne ta det pr epost. Er det større spørsmål (kanskje knyttet til et program dere sitter med) så be om et møte.

Vi kan kontaktes på epost:

inf1000@ifi.uio.no

LYKKE TIL 3. DESEMBER !!