

**Skriftlig eksamen i
Grundlæggende Programmering**

IT-C, 17. januar 2001
Dansk udgave

Alle hjælpemidler er tilladt, dog ikke datamat.

Eksamen er skriftlig, fire timer, og bedømmes efter 13-skalaen.

Opgavesættet består af fire opgaver, der alle ønskes løst. De fire opgaver vægtes lige.

Hvis besvarelsen benytter klasser og metoder fra lærebogen, forelæsningsnoterne, eller forelæsningsplancherne, så behøver du ikke at skrive dem af, men du skal give en præcis henvisning med afsnitsnummer, sidetal, osv.

The exam questions are available also in English (separately). Eksamensspørgsmålene findes også på engelsk (på separate ark).

Et godt råd: Gennemlæs opgavesættet inden du begynder at besvare de enkelte opgaver.

Opgave 1

Opgave 1.1

Vis hvad nedenstående Java-program udskriver på skærmen når det køres. Pas nøje på antallet af linieskift og blanktegn (mellemrum):

```
class Opgave1_1 {
    public static void main(String[] args) {
        line(5); line(2); line(5);
        System.out.println();
        line(5); line(5); line(5);
    }

    static void line(int i) {
        switch (i) {
            case 0: System.out.println("  "); break;
            case 1: System.out.println(" *"); break;
            case 2: System.out.println(" * "); break;
            case 4: System.out.println("* "); break;
            case 5: System.out.println("* *"); break;
        }
    }
}
```

Opgave 1.2

Skriv en metode `static void show(int d)` som kan kaldes med en parameter `d` mellem 1 og 6, og som udskriver et billede af en side af en terning, hvor antallet af øjne er `d`. Benyt gerne metoden `line`. For eksempel skal kaldet `show(2)` skrive

```
*

```

```
*

```

og kaldet `show(3)` skal skrive

```
*

```

```
*

```

```
*

```

Opgave 1.3

Skriv et komplet Java-program `Opgave1_3` der udskriver en hul kasse som er 5 linier høj og 5 plusser bred:

```
+++++
+   +
+   +
+   +
+++++
```

Opgave 1.4

Skriv en metode `static void fill(char c, int n)` som givet parametrene `c` og `n` udskriver en kasse der er `n` linier høj og `n` plusser bred, hvor kassens indre er udfyldt med tegnet `c` i stedet for blanktegn (mellemrum).

For eksempel skal kaldet `fill('o', 5)` udskrive denne udfyldte kasse:

```
+++++
+ooo+
+ooo+
+ooo+
+++++
```

Opgave 2

Denne opgave handler om datoer i år 2001. Månederne nummereres 1–12 (januar–december) og ugedagene nummereres 0–6 (mandag–søndag). Den 1. januar 2001 var en mandag (ugedag 0). År 2001 er ikke et skudår, så februar har 28 dage, og alle problemer vedrørende skudår kan ignoreres.

Her er et skelet til et program der regner på datoer i år 2001:

```
class Opgave2 {
    static int[] days =
        { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 };

    static int monthlen(int m) { ... }

    static String dayname(int wd) { ... }

    static int dayno(int m, int d) { ... }

    static int weekday(int m, int d) { ... }

    static void prmonth(int m) { ... }
}
```

Tabellen `days` indeholder 13 tal, sådan at `days[m]` er det samlede antal dage i året til og med måned `m`. For eksempel er `days[2]` lig med 59, nemlig antal dage i januar og februar tilsammen.

I delspørgsmålene nedenfor skal du færdiggøre metoderne `monthlen`, `dayname`, `dayno`, `weekday` og `prmonth`. Ved besvarelsen af hvert delspørgsmål må du gerne benytte metoder fra de forudgående delspørgsmål (også selvom du ikke har besvaret de forudgående delspørgsmål).

Opgave 2.1

Færdiggør metoden `monthlen`, sådan at kaldet `monthlen(m)` returnerer antal dage i måned nummer `m`. Benyt tabellen `days`. For eksempel skal kaldet `monthlen(2)` give 28, nemlig antallet af dage i februar 2001.

Opgave 2.2

Færdiggør metoden `dayname`, sådan at kaldet `dayname(wd)` returnerer navnet på ugedagen med nummer `wd`. (Du bestemmer selv om det skal være på dansk eller engelsk; for eksempel kan `dayname(0)` returnere strengen "mandag" eller "Monday" eller "Mon").

Opgave 2.3

Færdiggør metoden `dayno`, sådan at kaldet `dayno(m, d)` returnerer antallet af dage siden 1/1 (i år 2001). For eksempel skal `dayno(1, 1)` give 0, `dayno(1, 17)` skal give 16, `dayno(1, 31)` skal give 30, `dayno(2, 1)` skal give 31, `dayno(12, 1)` skal give 334, og `dayno(12, 31)` skal give 364.

Opgave 2.4

Færdiggør metoden `weekday`, sådan at kaldet `weekday(m, d)` returnerer nummeret på ugedagen for dag `d` i måned `m` (i år 2001). For eksempel skal `weekday(1, 1)` give 0 (mandag); `weekday(1, 17)` skal give 2 (onsdag); `weekday(2, 1)` skal give 3 (torsdag); `weekday(12, 1)` skal give 5 (lørdag), og `weekday(12, 31)` skal give 0 (mandag). Vink: Benyt modulus-operatoren (%), idet $0\%7$ er 0, $16\%7$ er 2, $31\%7$ er 3, og $334\%7$ er 5.

Opgave 2.5

Færdiggør metoden `prmonth` sådan at kaldet `prmonth(m)` udskriver en kalender for måned `m` (i år 2001).

For eksempel skal kaldet `prmonth(1)` udskrive kalenderen for januar 2001:

```
Mon 1
Tue 2
Wed 3
Thu 4
Fri 5
Sat 6
Sun 7
Mon 8
...
Wed 17
Thu 18
...
Wed 31
```

Tilsvarende skal kaldet `prmonth(2)` udskrive kalenderen for februar 2001:

```
Thu 1
Fri 2
...
Wed 28
```

Opgave 3

Denne opgave handler om kurser og studerende på IT-højskolen. Kurser repræsenteres som objekter af klassen `Course`, vist nedenfor. Et kursus har et navn (`name`) og en bestemt ugedag (`weekday`) som er et heltal: 0 betyder mandag, 1 betyder tirsdag, ..., 6 betyder søndag.

Der er to typer af studerende: ITC-studerende (som har et studienummer og følger nul eller flere kurser), og enkeltfagsstuderende (som ikke har noget studienummer og følger netop ét kursus).

Den abstrakte klasse `Student`, vist nedenfor, beskriver det der er fælles for alle studerende: den studerende har et navn (feltet `name`), og man kan spørge hvor mange kurser den studerende følger (med metoden `getCourseCount`). Den studerendes kurser er nummereret 0, 1, ..., så man kan bede om den studerendes kursus nummer 0, 1, ... (med metoden `getCourse`), og endelig kan man udskrive alle oplysninger om den studerende (med metoden `print`).

```
class Course {
    String name;
    int weekday;

    Course(String name, int weekday)
    { this.name = name; this.weekday = weekday; }

    int getDay() { return weekday; }

    public String toString() { return name; }
}

abstract class Student {
    String name;
    abstract int getCourseCount();
    abstract Course getCourse(int i);
    abstract void print();
}

... erklæring af klassen SingleCourseStudent ...

... erklæring af klassen ITCStudent ...
```

Delspørgsmålene nedenfor går ud på at færdiggøre klasserne `SingleCourseStudent` og `ITCStudent` så de kan bruges som vist i dette program:

```
class Opg3 {
    public static void main(String[] args) {
        Course gp = new Course("Introductory Programming", 2);
        Course dbd = new Course("Design of User Interfaces and Data", 4);
        Course dbw = new Course("Database-based Web Publishing", 4);

        Student s1 = new SingleCourseStudent("Svend Bergstein", gp);

        Course[] courses1 = { gp, dbd, dbw };
        Student s2 = new ITCStudent("Birthe Weiss", courses1, 4711);
        Student s3 = new ITCStudent("Jan Trøjborg", courses1, 666);

        s1.print();
        s2.print();
        s3.print();
    }

    static boolean dayClash(Student s) { ... }
}
```

Opgave 3.1

Erklær klassen `SingleCourseStudent` som en subklasse af `Student`, med et felt `course` (af type `Course`), og med en konstruktør der tager navn og kursus som argumenter, så den kan kaldes som vist i `main`-metoden ovenfor.

Der skal også være passende implementeringer af metoderne `getCourseCount` og `getCourse`. Metoden `getCourse` skal blot returnere den studerendes (ene) kursus, uanset hvilket argument den kaldes med.

Opgave 3.2

Erklær klassen `ITCStudent` som en subclasses af `Student`, med nye felter `courses` (af type `Course[]`) og `studentId` (af type `int`), og med en konstruktor der tager navn, kursustabel og studienummer, så den kan kaldes som vist i `main`-metoden ovenfor.

Der skal også være passende implementeringer af metoderne `getCourseCount` og `getCourse`. Kaldet `s.getCourseCount()` skal returnere antallet af kurser som studerende `s` følger. Kaldet `s.getCourse(i)` skal returnere kursus nummer `i` for studerende `s`, hvor de mulige værdier for `i` er `0`, `1`, ..., op til antallet af kurser minus én. Hvis den studerende `s` for eksempel følger 3 forskellige kurser, så skal man kunne kalde `s.getCourse(i)` for `i` lig med `0`, `1` og `2`.

Opgave 3.3

Skriv en implementering af metoden `print` i såvel `SingleCourseStudent` som `ITCStudent`. De to implementeringer skal være skrevet så `main`-metoden vist ovenfor giver denne udskrift når den køres:

```
Svend Bergstein, single-course student:  
Introductory Programming
```

```
Birthe Weiss, id 4711:  
Introductory Programming  
Design of User Interfaces and Data  
Database-based Web Publishing
```

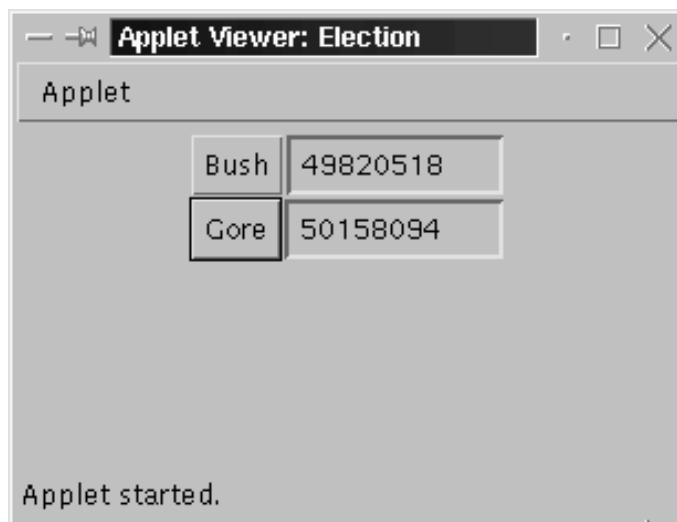
```
Jan Trøjborg, id 666:  
Introductory Programming  
Design of User Interfaces and Data  
Database-based Web Publishing
```

Opgave 3.4

Skriv metoden `static boolean dayClash(Student s)` i klassen `Opg3`. Metoden skal returnere `true` hvis den studerende har to kurser der ligger på samme ugedag, ellers skal den returnere `false`. Vink: Benyt metoderne `getCourseCount` og `getCourse` fra `Student`-klassen.

Opgave 4

Denne opgave omhandler en applet til stemmeoptælling. Appletten har to knapper mærket 'Bush' og 'Gore' og to tilsvarende tekstfelter (med plads til 10 tegn) som viser hvor mange gange der har været trykket på hver af de to knapper. Her er et billede af appletten:



Her er et skelet til Applet-klassen:

```
import java.applet.Applet; import java.awt.*; import java.awt.event.*;

public class Election extends Applet {
    int cntBush = 0, cntGore = 0;
    Button butGore, butBush;
    TextField txtGore, txtBush;

    public void init() {
        Panel p = new Panel();
        ...
        add(p);
    }

    class BushListener implements ActionListener { ... }

    class GoreListener implements ActionListener { ... }
}
```

Kommentarerne `/* a */`, `/* b */` osv. har ingen effekt; de er kun indsat for at kunne navngive programpunkter i delspørgsmålene nedenfor.

Opgave 4.1

Skriv Java-kode til programpunkt `/* b */` i metoden `init` til at oprette de fire komponenter, give appletten en passende layout-manager, og sætte komponenterne på appletten, så den får det udseende der er vist i figuren ovenfor.

Vis med en tegning hvordan du forestiller dig eventuelle hjælpe-Paneler og komponenternes placering inden for disse.

Vink: De fire komponenter kan hensigtsmæssigt anbringes på et Panel `p` der forsynes med `GridLayout` eller `BorderLayout` (eventuelt med ekstra hjælpe-Paneler). Tekstfelterne må ikke kunne redigeres af brugeren.

Opgave 4.2

Skriv indmaden til de to lytterklasser `BushListener` og `GoreListener` (ved programpunkterne `/* c */` og `/* d */`), sådan at tryk på en knap medfører at det tilsvarende stemmetal forøges med én og vises i det tilsvarende tekstfelt. Felterne `cntBush` og `cntGore` skal indeholde de aktuelle stemmetal.

Skriv også de ordrer der skal tilføjes til `init` for at oprette lytterne og knytte dem til knapperne.

Opgave 4.3

Tilføj yderligere et tekstfelt `status` med plads til 30 tegn, og placér det efter panelet `p`. Når der er trykket på en knap skal det nye tekstfelt vise en af følgende tre tekster: 'Bush fører med x stemmer' eller 'Gore fører med y stemmer' eller 'Bush og Gore står lige', hvor x og y skal være den aktuelle forskel i stemmetal. I eksemplet vist på foregående side skulle teksten altså være 'Gore fører med 337576 stemmer'. Beskriv nøje alle nødvendige tilføjelser og rettelser til appletten, og hvor de skal laves (hvilke programpunkter).

Opgave 4.4

Lav appletten så statusteksten også er korrekt når forskellen på stemmetallene er én. Statusteksten skal i så fald være for eksempel 'Gore fører med 1 stemme', ikke 'Gore fører med 1 stemmer'.

Opgave 4.5

Tilføj et søjlediagram med to liggende søjler der viser de forholdsmæssige stemmetal for Bush og Gore. Placér det efter de øvrige komponenter på appletten. Canvasset som indeholder grafikken skal være 250 pixels bredt og 50 pixels højt. Skalér søjlerne så søjlen for den førende kandidat udfylder hele bredden (hvis begge kandidater har nul stemmer skal der ikke vises noget). Vink: Lav en subklasse af `Canvas` som en indre klasse i `Election`, sådan at canvassets `paint`-metode kan benytte felterne `cntBush` og `cntGore` i `Election`-klassen.

Her er et eksempel på applettens udseende tidligt i optællingen:

