

## Tentamen i Objektorienterad programmering

Fredagen 15 mars 2013, 14.00 – 18.00.

Jourhavande lärare: Björn von Sydow, tel 0722/391401.

Inga hjälpmedel.

Lösningar till uppgifterna behöver ej kommenteras. Ej heller behöver nödvändiga importers anges. Triviala syntaxfel tolereras utan poängavdrag vid rättningen. Skriv läsligt!

Skrivningen kan ge maximalt 40 p; 20 p ger med säkerhet godkänt. 27 p ger betyget 4 och 33 p betyget 5.

1. (a) Definiera en funktion

```
public static int maxabs(int[] a)
```

som returnerar det största absolutbeloppet av talen i a. Se nedan för exempel. Du får anta att a har längd minst ett. (3 p)

- (b) Definiera en klass Uppgift1 som innehåller funktionen maxabs från (a) samt en main-rutin, som testar maxabs genom att läsa ett antal tal från kommandoraden, placera dessa i ett fält samt anropa maxabs och skriva ut resultatet. Exempel på körning:

```
> java Uppgift1 1 4 -3 9 2
9
> java Uppgift1 7 -10 5
10
```

(3 p)

- (c) Rita en principskiss över datorns minne när ditt program körs med det första exemplet i (b) ovan och exekveringen hunnit så långt att man just ska returnera värdet 9 i funktionen maxabs. Skissen ska ha med minnesceller för alla variabler och illustrera referenser med pilar till andra minnesceller. (3 p)

- (d) Lägg till en funktion

```
public static double[] scale(int[] a)
```

som returnerar en skalad variant av a: talen har gjorts om till typen double och dividerats med en gemensam konstant så att det största talet har absolutbelopp 1. Om a innehåller talen 7, -10 och 5, så ska scale(a) innehålla 0.7, -1.0 och 0.5. Du behöver inte lägga till något test av scale till main. (3 p)

2. (a) I ett program som administrerar studenters studieresultat förekommer en klass `Course`, vars objekt ger information om en viss kurs' namn och poängtal. Klassens utseende är följande<sup>1</sup>:

```
public class Course {
    private String name;
    private int credits;

    public Course(String name, int credits) {...}

    public String getName() {...}

    public int getCredits() {...}
}
```

Komplettera klassen med kropparna för konstrueraren och de två metoderna. (2 p)

- (b) I samma program finns en klass `Student`, som lagrar information om en student och dennas avklarade kurser. Klassen har följande metoder:

- `public String getName()`.  
Returnerar studentens namn.
- `public void add(Course c)`.  
Lägger till informationen att studenten klarat kursen `c`.
- `public int getNrPassed()`.  
Returnerar antalet avklarade kurser.
- `public Course getCourse(int i)`.  
Returnerar den `i`:te godkända kursen; för `i=0` fås den först godkända kursen, osv.
- `public int totalCredits()`.  
Returnerar det totala antalet avklarade poäng, dvs summan av poängtalerna för alla avklarade kurser.

Definiera klassen `Student`. Välj själv lämpliga parametrar till konstrueraren. (6 p)

3. Linuxkommandot `head` fungerar på följande sätt:

- Det kan ta ett filnamn som kommandoradsargument och skriver då ut de tio första raderna från denna fil i terminalfönstret. Innehåller filen färre än tio rader skrivs hela filen ut.
- Det kan också ta flera filnamn som kommandoradsargument. Då skrivs de (högst) tio första raderna från varje fil ut, med en rubriklad för varje fil, som i följande exempel:

```
> head fil1.txt fil2.txt
==> fil1.txt <==
Här följer de första tio raderna ur fil1.txt
==> fil2.txt <==
Här följer de första tio raderna ur fil2.txt
>
```

Skriv ett Java-program som fungerar på samma sätt som detta Linuxkommando. (6 p)

**Anm:** Objekt ur klassen `File` skapas med en konstruerare som tar filnamnet (en `String`) som parameter.

---

<sup>1</sup>Vi noterar att i detta system måste poängtalet vara ett heltal. Tyvärr kan alltså denna kurs, som ger 7.5 poäng, inte hanteras av programmet

4. En svartvit bild kan representeras med en matris av heltal, dvs ett objekt av typen `int[][]`. I en bild `image` anger `image[x][y]` bildens ljushet i den pixel som ligger `x` pixlar från bildens vänsterkant och `y` pixlar från dess överkant. Värdena i matrisen ligger mellan 0 och 255, där 0 betyder att motsvarande pixel i bilden är svart, 255 betyder att pixeln är vit och mellanliggande värden representerar olika nyanser av grått. Vi ska i denna uppgift implementera en form av filtrering av sådana bilder, som brukar användas för att anonymisera ansikten i fotografier. Vi betraktar följande tre varianter av ett foto av Barack Obama, där den vänstra bilden är originalet, en bild med  $349 \times 492$  pixlar.



- (a) I mittenbilden har ett kvadratisk *block* av pixlar i bilden ändrats. Blockets övre vänstra hörn är pixeln (100,160) och blocket har storlek  $50 \times 50$  pixlar. I detta block har alla pixlar getts samma värde, medelvärdet av värdet av alla pixlar i blocket innan ändringen.

Definiera en rutin

```
public static void blockpixelate(int[][] image, int x, int y, int side)
```

som gör en sådan ändring i bilden `image`, i ett block som har övre vänstra hörnet i  $(x,y)$  och storlek `side` $\times$ `side` pixlar. (4 p)

- (b) I den högra bilden har rutinen från (a) upprepade gånger använts på block av storlek  $20 \times 20$  pixlar över hela bilden.

Definiera en rutin

```
public static void pixelate(int[][] image, int side)
```

som gör en sådan filtrering. Parametern `side` anger också här blockstorleken. Det är tillåtet att, som i bilden ovan, låta bilden vara oförändrat i ett smalt band nertill och/eller till höger om bildens höjd och/eller bredd inte är en multipel av `side`. (4 p)

5. Ett hierarkiskt filsystem består dels av *filer*, dels av *kataloger*. En katalog innehåller i sin tur filer och andra kataloger.

Objekten i klassen `java.io.File` representerar namn på sådana filer och kataloger. Klassen innehåller en konstruerare

```
public File(String name)
```

samt bland annat följande metoder:

- `public File[] listFiles()`, som returnerar `null`, om det aktuella `File`-objektet är en fil och som returnerar en lista av dess filer och kataloger om det är en katalog.
- `public String getPath()` som returnerar det fullständiga relativa filnamnet för filen i förhållande till den aktuella katalogen, t ex

```
labbar/lab3/Main.java
```

- `public String getName()` som returnerar bara själva filnamnet; i exemplet ovan `Main.java`.

Skriv ett program `FindFile` som tar ett filnamn på kommandoraden och listar alla filer med detta namn som finns i någon underkatalog. Tänkbart exempel på körning:

```
> java FindFile Main.java
labbar/lab1/Main.java
labbar/lab2/Main.java
lectures/lect1/code/Main.java
>
```

(6 p)