

## Tentamen i Programmeringsteknik I 2009-10-16

- Skrivtid: 14:00-17:00
- Hjälpmedel: Lewis & Loftus, Java Software Solutions eller Skansholm, Java Direkt med Swing. Litteraturen får inte innehålla överdrivet mycket anteckningar.
- Tentamen består av fyra uppgifter och maxpoängen är 24.
- För att bli godkänd på tentamen (betyg 3 eller G) krävs minst 15 poäng.
- För att få betyg VG på tentamen krävs dels minst 20 poäng.
- För att få betyg 4 på tentamen krävs dels minst 18 poäng.
- För att få betyg 5 på tentamen krävs dels minst 21 poäng.
- Programmen skall vara skrivna i Java om inget annat anges.
- Programmen skall vara skrivna med vettig layout och vara kommenterade när så behövs.
- Skriv varje uppgift på nytt papper.
- Skriv inte på baksidan.
- Lägg uppgifterna i rätt ordning.
- Skriv namn på alla papper.
- Använd inte rödpenna!
- Läs uppgifterna noga så du vet vad som skall göras.
- Gör inte mer än det som efterfrågas.
- Skriv uppgiftens nummer i övre högra hörnet på varje blad.
- Oklarheter: Fråga! Frågorna kan vara svåra men ska inte vara svårbegripliga.

*Lycka Till* önskar Anna

## Uppgift 1

### En enkel klass

Skriv en klass, `Vindkraft`, som förenklat beskriver ett vindkraftverk.

Ett vindkraftverk kan i vår modell beskrivas med sitt namn, storleken på generatoren (i kW) samt radien på rotorbladen.

Klassen ska innehålla två konstruktörer, en standardkonstruktor som inte tar några inparametrar och en överlagrad konstruktor som har tre inparametrar.

Klassen har två metoder:

**toString** Metoden returnerar en sträng som innehåller värdena på instansvariablerna hos ett objekt.

**prodEnergi** Metoden ska beräkna och returnera mängden energi som produceras under ett tidsintervall. Metoden har två inparametrar, `tid` och `medelVind`, som båda är reella tal. Energin beräknas enligt följande formel:

```
energi = rotorRadie*generatorStorlek*tid*medelVind
```

Skriv slutligen en main-metod som gör följande:

- Skapar ett `Vindkraft`objekt med hjälp av den överlagrade konstruktorn. Du får själv välja lämpliga värden till instansvariablerna.
- Skriver ut värdena på objektets instansvariabler.
- Läser in `tid`- och `medelVind`-värden.
- Beräknar mängden producerad energi med hjälp av metoden `prodEnergi`.
- Skriver ut mängden producerad energi.

(6p)

## Uppgift 2

### Klassen Krets

Klassen `Krets` beskriver resistorer i en elektrisk krets. Klassen har två instansvariabler, en array av `double` som representerar resistansen hos ett antal resistorer i kretsen, och ett heltal `antal` som visar hur många resistorer som är inkopplade i kretsen för närvarande. Klassen har följande metoder:

- En konstruktor med en heltalsparameter som skapar arrayen så lång som parametern anger.
- En metod `lasIn` som frågar efter och läser in värden på resistorer, dvs värden till arrayen. Antalet värden som läses in ska anges i parameterlistan.
- En metod `serie` som beräknar den totala resistansen  $R$  om alla resistorer är seriekopplade:  
$$R = R_1 + R_2 + \dots + R_n$$
om  $n$  resistorer finns i kretsen. Metoden skriver ut den totala resistansen med lämplig ledtext.
- En metod `kopplaUr` som tar bort en resistor från kretsen. Index på den resistor som ska tas bort skickas i parameterlistan. Metoden returnerar inget värde.
- En metod `parallell` som beräknar den totala resistansen  $R$  om alla resistorer är parallellkopplade:  
$$1/R = 1/R_1 + 1/R_2 + \dots + 1/R_n$$
om  $n$  resistorer finns i kretsen. Metoden skriver inte ut något utan returnerar värdet på den totala resistansen.

Skriv klassen `Krets` med de metoder som beskrivs ovan. Du behöver inte skriva någon `main`-metod som testar klassen.

(6p)

### Uppgift 3

#### Klassen Vinkel

Antag att du har en klass `Vinkel` för att hantera vinklar. En vinkel bestäms av hur många grader, minuter och sekunder vinkeln avviker från en noll-linje, t ex x-axeln i ett koordinatsystem. Riktning moturs ger positiv vinkel och medurs ger negativ vinkel. En minut är en sextiondel av en grad, och en sekund är en sextiondel av en minut. Klassen `Vinkel` har följande utseende:

```
import java.util.Scanner;
public class Vinkel {
    private int grader;
    private int minuter;
    private int sekunder;

    public Vinkel() {
        grader = minuter = sekunder = 0;
    }

    public Vinkel(int g, int m, int s) {
        grader = g;
        minuter = m;
        sekunder = s;
    }

    public ...lasIn(...) { //läser in vinkelns grader, minuter och sekunder
        ...
    }

    public...skrivUt(...) { //skriver ut vinkelns grader, minuter och sekunder
        ...
    }

    public...addera(...) { //adderar två vinklar. Inget skrivs ut
        ...
    }
    public...subtrahera(...) { //subtraherar två vinklar. Inget skrivs ut
        ...
    }

    public static void main(String [] arg) {
        ...
    }
}
```

- a) Definiera metoderna `lasIn`, `skrivUt` och `addera`. Parameterlistor och returtyp får du bestämma själv, men metoden `addera` skall inte skriva ut något. Du kan anta att metoden `subtrahera` redan finns definierad.
- b) Skriv ett huvudprogram, `main`, som skapar minst tre vinklar och sedan läser värden till vinklarna med metoden `lasIn`. Programmet skall sedan addera två av vinklarna och skriva ut resultatet. Därefter skall den tredje inlästa vinkeln subtraheras från resultatet av additionen. Resultatet av subtraktionen skall skrivas ut.

Exempel: Om vinklarna

19 grader, 10 minuter och 50 sekunder,

20 grader, 16 minuter och 33 sekunder,

8 grader, 20 minuter och 13 sekunder

lästs in blir resultatet av additionen av de två första vinklarna:

39 grader, 27 minuter och 23 sekunder.

Resultatet av subtraktionen mellan detta värde och det sist inlästa blir:

31 grader, 7 minuter och 10 sekunder.

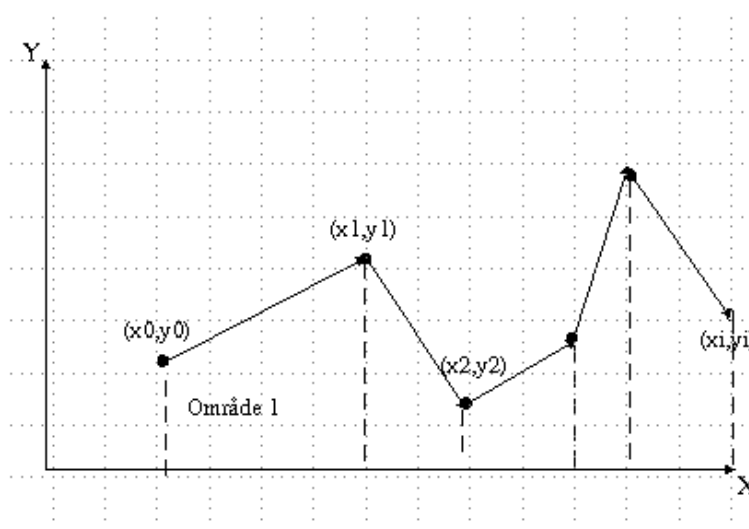
Inga nya metoder får läggas till.

(6p)

## Uppgift 4

### Klassen PolySet

Klassen `PolySet` kan användas för att behandla *polygontåg*, dvs. en följd av punkter i planet. Instansvariablerna ska vara en array med x-koordinater, en array med y-koordinater, samt antalet punkter som finns inlagda. x-koordinaterna ligger lagrade i stigande ordning i sin array.



Din uppgift är att skriva några metoder till klassen `PolySet`. Du kan anta att någon annan skriver resten av klassen. Skriv följande metoder:

- En metod `maxY` söker reda på och returnerar det maximala y-värdet för någon punkt i polygontåget.
- En metod `riktningskoefficient` som beräknar riktningkoefficienten för ett visst delområde, se figuren ovan. Området där riktningkoefficienten skall beräknas anges som ett tal, exempelvis 1, och skickas som en parameter till metoden. Om parameterns värde är 1 avses den räta linjen mellan talparen  $(x_0, y_0)$  och  $(x_1, y_1)$ , om parameterns värde är 2 avses linjen mellan talparen  $(x_1, y_1)$  och  $(x_2, y_2)$ , osv. Riktningkoefficienten mellan två punkter, t ex punkterna  $(x_0, y_0)$  och  $(x_1, y_1)$  fås med formeln:

$$\text{riktningkoefficienten} = (y_1 - y_0) / (x_1 - x_0)$$

- En metod `integral` som beräknar arean under polygontåget och returnerar detta värde. Arealen beräknas genom att man summerar areorna för alla delintervallen, enligt följande formel (som ger arean för område 1 i figuren):

$$\text{delarea} = (y_1 + y_0) \cdot (x_1 - x_0) \cdot 0.5$$

(6p)