

Tentamen – obligatorisk del

Förklaringar

Den här delen av tentamen är obligatorisk och tillräcklig för betyget E

För att klara tentamen, måste studenten klara den här delen av tentamen. Om bara den här delen klaras, blir betyget på tentamen E. För att uppnå ett högre betyg, måste studenten samla ett tillräckligt antal poäng även på den extra delen av tentamen. Beroende på detta antal, kan betyget på tentamen bli D, C, B eller A.

För att klara den här delen av tentamen

För att klara den här delen av tentamen, måste studenten uppnå minst två tredjedelar av det totala antalet poäng på den här delen. Den här delen bidrar inte på något sätt till de högre betygen. Betyget på denna del är antingen P eller F.

Antalet poäng

Totalt: 33 poäng

För betyget P krävs minst: 22 poäng

Utforma noggrant dina svar, kodavsnitt och bilder

Formulera dina svar kortfattat och noggrant.

Koden ska utformas så att det lätt går att följa och förstå den. I vissa situationer kan lämpliga kommentarer bidra till förståelse. Små syntaktiska fel i koden kan eventuellt tolereras. Om delar i ett kodavsnitt inte kan exakt formuleras, kan möjligen en välutformad pseudokod bidra till lösningen. Man ska inte skriva mer kod än som behövs: om bara en metod krävs, behöver inte en hel klass skapas. All kod ska skrivas i Java.

När en vektor eller ett objekt ritas, ska det klart framgå vilka data som finns inuti denna vektor eller detta objekt. När en vektor eller ett objekt innehåller en referens, ska även den refererade resursen (ett objekt eller en vektor) ritas. Man ska förse alla referenser med relevanta beteckningar.

Uppgifter

Uppgift 1 (1 poäng + 1 poäng)

a)

```
int[]    u = new int[4];
u[0] = 4;
u[1] = 5;
int[]    v = new int[5];
v[v.length - 1] = 10;
v[v.length / 2] = 20;
```

Rita de skapade vektorerna och referenserna.

b)

```
int[]    u = new int[4];
u[0] = 4;
u[1] = 5;
int[]    v = new int[5];
u = v;
```

Hur ser de skapade vektorerna ut när det här kodavsnittet har utförts: rita vektorerna och motsvarande referenser.

Vad händer med den första vektorn när detta kodavsnitt har utförts?

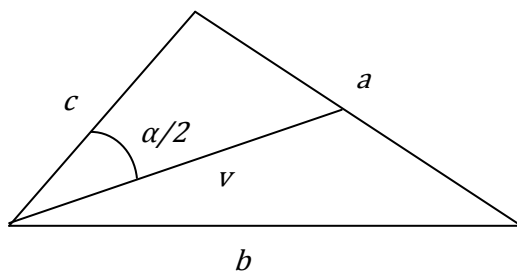
Uppgift 2 (1 poäng + 2 poäng)

10	20	30	40
5	10	15	20
4	3	2	1

- a) Skapa en tvådimensionell vektor (eng. *array*) som representerar data i den givna tabellen. Använd operatorm `new`.
- b) Rita den skapade vektorn, så att det framgår hur vektorn lagras i datorns minne. Både vektorns celler, motsvarande data och alla referenser ska finnas med. Det ska även finnas motsvarande beteckningar på referenserna.

Uppgift 3 (2 poäng + 1 poäng)

Längder av sidor i en triangel är a , b och c . Vinkeln som motsvarar sidan a är α , och längden av den motsvarande bisektrisen är v .



Längden av bisektrisen kan beräknas enligt följande formel:

$$v = (2 * b * c * \cos(\alpha / 2)) / (b + c)$$

- a) Skapa en statisk metod `bisektris`, som tar emot längderna b och c och vinkeln α , och returnerar längden av bisektrisen v .
- b) Anropa metoden `bisektris` för att bestämma längden av bisektrisen i fall att: $b = 4$, $c = 4$, $\alpha = \pi / 2$.

Uppgift 4 (2 poäng + 2 poäng)

- a) En statisk metod `max` tar emot en icke-tom, endimensionell vektor med heltal, och returnerar det största heltalet i vektorn. Metoden är både minneseffektiv och tidseffektiv. Skapa den metoden.
- b) En statisk metod `summa` tar emot en icke-tom, tvådimensionell vektor med heltal, och returnerar heltalens summa. Skapa den metoden.

Uppgift 5 (3 poäng)

En algoritm bestämmer det minsta elementet i en heltalsmängd. Denna algoritm kan beskrivas så här:

Algoritm: *min*

Förvillkor:

$$n \in \mathbb{N}, n \geq 1, X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{Z}$$

(\mathbb{N} – mängden av alla naturliga heltal, \mathbb{Z} – mängden av alla heltal)

Eftervillkor:

$$m \in \mathbb{N}, 1 \leq m \leq n: x_m = \text{minimum } X$$

```

min (n, X)
{
    i = 1
    m = 1
    while i < n
    {
        i++
        if (x_i < x_m)
            m = i
    }

    return x_m
}

```

Spåra denna algoritm i samband med följande mängd:

$$X = \{5, 8, 9, 4, 7, 6, 2, 10\}$$

Samla relevanta data i en tabell av följande form:

i	x_i	m	x_m

Uppgift 6 (2 poäng + 1 poäng)

En statisk metod `sifferSumma` tar emot en icke-tom teckensträng, som bara innehåller siffror. Metoden returnerar summan av alla dessa siffror.

a) Skapa metoden `sifferSumma`. Metoden ska vara minneseffektiv: man ska inte omvandla teckensträngen till motsvarande teckenvektor.

b) Anropa metoden `sifferSumma` på något sätt.

Uppgift 7 (3 poäng)

```

class V
{
    public static void main (String[] args)
    {
        StringBuilder str = new StringBuilder ("aaaa");
    }
}

```

```
        change (str);
        str.insert (0, "A");
        System.out.println (str);
    }

    public static void change (StringBuilder s)
    {
        s.append ("A");
        s = new StringBuilder ("bbbb");
        s.setCharAt (0, 'B');
        System.out.println (s);
    }
}
```

Vilken utskrift skapas när det här programmet exekveras?

Uppgift 8 (2 poäng + 1 poäng + 2 poäng)

En klass Punkt representerar en punkt i planet:

```
public class Punkt
{
    // punktens koordinater
    private double    x;
    private double    y;

    // Punkt initierar punkten utifrån givna koordinater
    public Punkt (double x, double y)
    {
        this.x = x;
        this.y = y;
    }

    // avstand returnerar avståndet mellan punkten och en given punkt
    public double avstand (Punkt p)
    {
        // koden här
    }
}
```

a) Implementera metoden avstand.

b) Skapa två punkter av typen Punkt, och anropa metoden avstand i samband med dem.

c) Rita de skapade punkterna i ett koordinatsystem. Ange på bilden de variabler som används i implementationen av metoden avstand (till exempel variablerna this.x, p.x, o.s.v.)

Uppgift 9 (2 poäng + 2 poäng)

Klassen Mangd representerar en heltalsmängd:

```
class Mangd
{
    private int[]    e = null;

    public Mangd (int[] e)
    {
        this.e = new int [e.length];
        for (int i = 0; i < e.length; i++)
            this.e[i] = e[i];
    }
}
```

a) Man skapar en vektor och ett objekt så här:

```
int[]    v = {1, 2, 3, 4, 5};
Mangd    m = new Mangd (v);
```

Rita den angivna vektorn och objektet av typen Mangd. Rita även motsvarande referenser och beteckna dem.

b) Rita vektorn, objektet och referenserna i fall att konstruktorn implementeras så här:

```
public Mangd (int[] e)
{
    this.e = e;
}
```

Uppgift 10 (1 poäng + 2 poäng)

Klasserna A och B implementeras så här:

```
class A
{
    protected int    n;

    public A (int n)
    {
        this.n = n;
    }

    public String toString ()
    {
        return "" + n;
    }
}

class B extends A
{
    public B (int n)
    {
        super (n);
    }

    public String toString ()
    {
        return "" + (1 + n);
    }
}

class AnvandAB
{
    public static void main (String[] args)
    {
        A[]    a = new A[4];
        a[0] = new A (4);
        a[1] = new B (4);
        a[2] = new A (10);
        a[3] = new B (10);

        for (int i = 0; i < a.length; i++)
            System.out.println (a[i]);
    }
}
```

a) Man lagrar i vektorn (som refereras av referensen) a både objekt av typen A och objekt av typen B. Varför är det möjligt?

b) Vilken utskrift skapas när det här programmet exekveras? Förklara!