

Gränssnitt och typoberoende programmering

En abstrakt modell av en polylinje

En polylinje i planet

En polylinje är en geometrisk figur som består av en serie bundna linjesegment. Ändpunkter för dessa segment utgör polylinjens hörn. En polylinje är bestämd med sina hörn, sin färg och sin bredd. En tom polylinje saknar hörn.

En polylinjes hörn, färg, bredd och längd kan erhållas. Färgen och bredden kan ändras. En polylinjes utseende ändras även när dess hörnsekvens ändras. Det går att lägga till ett nytt hörn till polylinjen – antingen på slutet, eller framför ett hörn vars namn är givet. Det går även att ta bort ett hörn med ett givet namn.

Man kan iterera genom en polylinje: polylinjens hörn kan besökas och användas i tur och ordning.

En abstrakt modell av en polylinje ska skapas: man ska skapa ett gränssnitt som heter `Polylinje`.

En modell av en polylinje

Man underförstår att det finns en klass `Punkt`, som på ett passande sätt representerar en punkt i planet. Objekt av denna klass ska användas för att representera polylinjens hörn.

```
public interface Polylinje extends java.lang.Iterable<Punkt>
{
    Punkt[] getHorn ();

    String getFarg ();

    int getBredd ();

    double langd ();

    void setFarg (String farg);

    void setBredd (int bredd);

    public void laggtill (Punkt horn);

    void laggtillframfor (Punkt horn, String hornNamn);

    void taBort (String hornNamn);

    java.util.Iterator<Punkt> iterator ();
}
```

Uppgifter i samband med polylinjer

1. Skapa en klass `VPolylinje` som representerar en polylinje i planet, och som implementerar gränssnittet `Polylinje`. Förutom de metoder som specificeras i gränssnittet, ska även metoden `toString` (som returnerar polylinjens strängrepresentation) implementeras. Polylinjens hörn ska lagras i en vektor av den inbyggda typen.

2. Skapa en klass `NPolylinje` som representerar en polylinje i planet, och som implementerar gränssnittet `Polylinje`. Förutom de metoder som specificeras i gränssnittet, ska även metoden `toString` (som returnerar polylinjens strängrepresentation) implementeras. Polylinjens hörn ska lagras i en sekvens av länkade noder.

Klassen `NPolylinje` ska påbörjas så här:

```
public class NPolylinje implements Polylinje
{
    private static class Nod
```

Från boken Programmeringsprinciper i Java : Exempel och övningar (Fadil Galjic & Studentlitteratur)

```

{
    public Punkt    horn;
    public Nod      nastaNod;

    public Nod (Punkt horn)
    {
        this.horn = horn;
        nastaNod = null;
    }
}

private Nod    horn;
private String  farg = "svart";
private int     bredd = 1; // pixlar

public NPolylinje ()
{
    this.horn = null;
}

public NPolylinje (Punkt[] horn)
{
    if (horn.length > 0)
    {
        Nod  nod = new Nod (new Punkt (horn[0]));
        this.horn = nod;
        int   pos = 1;
        while (pos < horn.length)
        {
            nod.nastaNod = new Nod (new Punkt (horn[pos++]));
            nod = nod.nastaNod;
        }
    }
}

// ytterligare kod här
}

```

3. Rita ett objekt av typen `NPolylinje`. Objektets nodsekvens (med motsvarande hörn) ska finnas med i ritningen.

4. Skapa ett gemensamt testprogram för klasserna `VPolylinje` och `NPolylinje`. En referens av gränssnittet `Polylinje` ska användas för att referera till objekt av de implementerade klasserna, och för att aktivera de olika metoderna. Man kan göra enligt följande mönster:

```

Polylinje    polylinje = null;
polylinje = new VPolylinje ();           // (1)
// polylinje = new NPolylinje ();        // (2)

```

Beroende på den klass som ska testas, bortkommenteras antingen satsen (1) eller satsen (2).

5. Kan man iterera en polylinje så här:

```

for (Punkt horn : polylinje)
    System.out.println (horn);

```

Varför?

6. Skapa en statisk metod – i en särskild klass som heter `Polylinjer`, som tar emot en vektor med ett antal polylinjer av typen `Polylinje`, och returnerar den kortaste av de polylinjer i vektorn som är gula.

Använd denna metod tre gånger: i samband med en vektor med polylinjer av typen `VPolylinje`, i samband med en vektor med polylinjer av typen `NPolylinje` och i samband med en vektor som innehåller polylinjer av båda typerna.

Hur kan en vektor innehålla objekt av olika typer? Hur kan en och samma metod ta emot objekt av olika typer?