

Kontrollskrivning 1, Programmeringsteknik för D/C

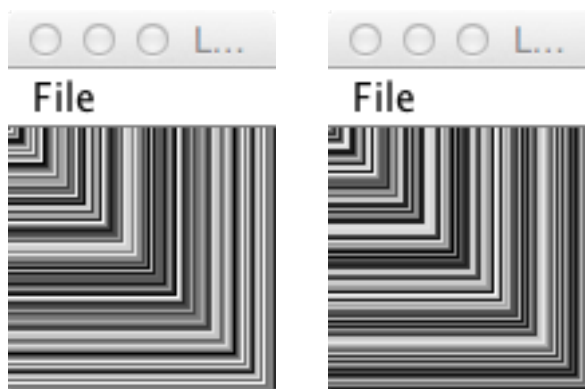
2014–10–14, 10.30–11.45

Anvisningar:

- Fyll i omslaget fullständigt, även lösta uppgifter och antalet inlämnade blad.
- Skriv bara på en sida av varje papper.
- Lämna bara in dina lösningar och omslaget, inte skrivningen och inte några kladdpapper.
- Tillåtet hjälpmedel: Java-snabbreferens.

Jag meddelar på kurshemsidan när rättningen är klar. Dina poäng på skrivningen anslås på institutionens anslagstavla (E-huset andra våningen, höger uppgång). Där meddelar jag också när och var du kan hämta ut din skrivning.

1. Skriv ett program som ritar linjer i ett `SimpleWindow`-fönster. Här visas resultatet av två exekveringar av programmet (fönstret har storleken 100×100 pixlar):



- För varje y-koordinat har man ritat en horisontell linje som börjar i vänsterkanten och går till fönstrets diagonal, sedan viker av uppåt och slutar i överkanten. Alla linjerna har bredden 1.
- Linjerna ritas med grå färg med slumpmässig intensitet, från helt svart till helt vitt.
- Man skapar fönstret med `new SimpleWindow(100, 100, "Lines")`. Programmet ska fungera även med andra "kvadratiska" fönster.
- `SimpleWindow`-operationen `moveTo(x,y)` flyttar pennan till punkten (x,y) utan att rita, operationen `lineTo(x,y)` ritar en linje.
- Linjefärgen anger man med operationen `setLineColor(color)`, där `color` är ett objekt av klassen `Color`. En grå färg skapar man med `new Color(i,i,i)`, där `i` ligger i intervallet $[0,255]$.
- Operationerna `getWidth()` och `getHeight()` tar reda på fönstrets bredd och höjd.

Vänd!

2. Antag att du tränar löpning och ska skriva ett program för att hålla reda på träningsresultat. Varje resultat beskrivs av ett objekt av klassen Result:

```
/** Skapar ett resultat med sträckan dist (meter) och tiden time (minuter)
    som gjorts datumet date (yyyy-mm-dd, till exempel "2014-01-15") */
Result(int dist, int time, String date);

/** Ändrar tiden till newTime */
void changeTime(int newTime);

/** Returnerar medelhastigheten i km/timme */
double getAverageSpeed();

/** Returnerar true om detta resultat har högre medelhastighet än
    resultatet other, annars false */
boolean isFasterThan(Result other);

/** Skriver ut uppgifter om resultatet på formen datum sträcka / tid,
    till exempel 2014-01-15 5000 m / 28 minuter
void print();
```

Implementera klassen.

3. Senare i kursen ska vi se hur man gör för att hålla reda på många objekt av samma slag. Nu nöjer vi oss med att se på ett exempelprogram som skapar och använder några resultatobjekt:

```
public class ExampleProgram {
    public static void main(String[] args) {
        Result r1 = new Result(5000, 28, "2014-01-15");
        Result r2 = new Result(5000, 20, "2014-01-18");
        Result r3 = r2;
        r3.changeTime(30);
        if (r1.isFasterThan(r2)) {
            System.out.println("Första snabbast");
        } else if (r2.isFasterThan(r1)) {
            System.out.println("Andra snabbast");
        } else {
            System.out.println("Lika snabba");
        }
    }
}
```

- Vilka variabler deklaras i main-metoden? Ange namn och typ.
- Hur många objekt skapas? (Räkna inte med String-objekt.)
- Vad skriver programmet ut?
- Rita en figur som visar vilka variabler och objekt som finns efter det att if-satsen exekverats. Ange också värdet av alla variabler och attribut. Ange värdena av referensvariabler med pilar.