

SI-pass 3 OOP

Handledare: Oscar Söderlund

Mail: soscar@student.chalmers.se

1. I Java kan man enkelt representera rektanglar med hjälp av en standardklass vid namn Rectangle, men det bryr vi oss inte om, för nu ska vi göra vår egen (bättre) rektangelklass!

Klassen skall heta Rectangle, och objekt av denna klass skall representera rektanglar med hjälp av bredd, höjd, samt koordinaterna för rektangelns övre vänstra hörn. Samtliga numeriska värden skall lagras som heltal.

Klassen skall tillhandahålla:

- En parameterlös default-konstruktor, som skapar en rektangel med bredden och höjden 50, samt koordinaterna (0,0).
- En konstruktor som låter användaren (dvs. programmeraren som använder klassen) själv specificera rektangelns utseende.
- Metoder för att avläsa kvadratens höjd och bredd, samt koordinater.
- En metod för att avläsa om rektangeln är kvadratisk.
- En metod, equals(Rectangle r), som avgör om två rektanglar är identiska.
- En metod, intersectsWith(Rectangle r), som avgör om två rektanglar överlappar.

2. Skriv en klass, SixDie, vars syfte är att simulera en sexsidig tärning. Klassen skall tillhandahålla en metod för att avläsa vilken sida av tärningen som för tillfället är uppåt, samt en metod för att kasta tärningen.

Tips: Math.random()

3. Betrakta följande javakod:

```
1: int a;  
2: a = 5;  
3: int b = a;
```

Betrakta sedan hur exekveringen av dessa rader kan illustreras med avseende på variablernas innehåll och arbetsminnets utseende:



Betrakta nu följande javakod:

```
1: Car myCar;  
2: myCar = new Car();  
3: Car anotherCar = myCar;
```

Gör en liknande illustration över hur arbetsminnet ser ut under exekveringen av dessa kodrader. Vad innehåller myCar och anotherCar-variablerna för värden?

Tentauppgifter

Här ser du ett felaktigt program. Avsikten är att det skall läsa in en text och kontrollera att texten bara innehåller siffror. Programmet innehåller tre olika fel. Det finns ett syntaxfel. Om man rättar detta fel och provkör programmet visar det sig att det innehåller ytterligare två fel: ett logiskt fel som gör att programmet inte utför det som det skall och ett fel som kan resultera i ett exekveringsfel. Ange vilka de tre felen är och ge för varje fel förslag på hur man kan skriva för att rätta felet!

```
import javax.swing.*;
public class FinnTreFel {
    public static void main (String[] arg) {
        String t = JOptionPane.showInputDialog("Ett tal?");
        for (int i=1; i<=t.length(); i++)
            if (t.charAt(i) >= '0' && <= '9')
                JOptionPane.showMessageDialog(null,"Talet är OK");
            else
                JOptionPane.showMessageDialog(null,"Inget tal");
        System.exit(0);
    }
}
```

(6 p)

Ett personnummer, t.ex. 561231-4913, består av tio siffror. Efter de sex första siffrorna skall det finnas ett minustecken. Den näst sista siffran är udda för män och jämn för kvinnor. Den sista siffran är en kontrollsifфра. Den beräknas på följande sätt: De nio första siffrorna i personnumret multipliceras omväxlande med 2 och 1, den första med 2 den andra med 1 den tredje med 2 osv. Siffrorna i de värden man då får adderas. I personnumret ovan blir det $(1+0)+6+2+2+6+1+8+9+2 = 37$. (Observera att 10 räknas som $1+0$.) Kontrollsiffran bestäms sedan av att den summa man fått plus kontrollsiffran skall vara jämnt delbar med 10. I exemplet blir alltså kontrollsiffran lika med 3.

Skriv en klass Personnummer som innehåller ett personnummer. Det skall finnas en konstruktor som får en String som parameter. Konstruktorn skall kontrollera att parametern innehåller ett korrekt personnummer (inklusive minustecknet). Det skall också finnas en metod toString som returnerar det aktuella personnumret som en String (med minustecken) samt två metoder ärMan respektive ärKvinna vilka ger en boolean som anger om personen är en man eller en kvinna.

(10 p)

Ett s.k. anagram får man om man utgår från ett ord eller en mening och kastar om bokstäverna så att ett nytt ord eller en ny mening framträder. Av ordet metoanrop kan man t.ex. bilda anagrammet peta ond orm. Skriv en klassmetod ärAnagram som undersöker om en text innehåller ett anagram av en annan text. De två texterna ges som parametrar till metoden. Resultatet skall ha typen boolean. Tips: Innan texterna jämförs kan man ta bort alla blanka tecken och se till att endast små (eller stora) bokstäver används.

(10 p)