Övning 3.

Syfte

På denna övning behandlas *arv* och *polymorfism.*

Uppgift 1

Klassen Employee nedan används för att beskriva anställda i ett företag:

```
public class Employee {
    private String name;
    private String address;
    private int  appointmentNo;
    private double payRate; //lön per månad
    private static int noOfEmployees = 0;
    public Employee(){
        noOfEmployees =  noOfEmployees +1;
        appointmentNo =  noOfEmployees;
    }
    public Employee(String name, String address, double payRate) {
        this();
        this.name = name;
        this.address = address;
        this.payRate = payRate;
    }
    public void setName(String name) {
        this.name =  name;
    }
    public String getName() {
        return name;
    }
    public int getAppointmentNo () {
        return appointmentNo;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getAddress() {
        return address;
    }
    public void setPayRate (double payRate) {
        this.payRate = payRate;
    }
    public double getPayRate() {
        return payRate;
    }
}
```

Dock har företagsledningen upptäckt att denna klass har vissa brister eftersom man för alla chefer förutom månadslönen infört en månatlig bonus, samt för vissa högre chefer även tillgång till ett hembiträde.

Skriv en klass Boss som på lämpligt sätt utökar klassen Employee för att avbilda de särskilda förmånerna som cheferna i företaget har.

Uppgift 2 *Uppgiften reviderad 090917*

På laboration 1 skrev ni två klasser för att avbilda vanliga 6-sidig tärningar respektive tärningar med godtyckligt antal sidor. Ni skall nu göra något liknande.

a) Definiera en generell klass Die för tärningar med godtyckligt antal sidor. Klassens konstruktor skall ta antalet sidor som inparameter, och det skall finnas accessmetoder för att läsa av antalet sidor, antalet ögon efter senaste kastet, samt en metod som kastar tärningen. Alla instansvariabler skall vara privata. Definiera sedan en klass Die6, för sexsidiga tärningar, som subklass till Die. Hur bör klassens konstruktor se ut? Vilka metoder är nödvändiga i subklassen – och vilka kan ärvas från basklassen? Skapa också en klass DoubleDie, som är en 6-sidig tärning vars sidor har värdena 2, 4, 8, 16, 32 och 64 (en sådan tärning används t.ex. i *backgammon*). Vilken av de två andra klasserna kan vara en lämplig basklass till DoubleDie? Tips: Överskugga en av basklassens metoder på listigt sätt!

b) Skriv en metod

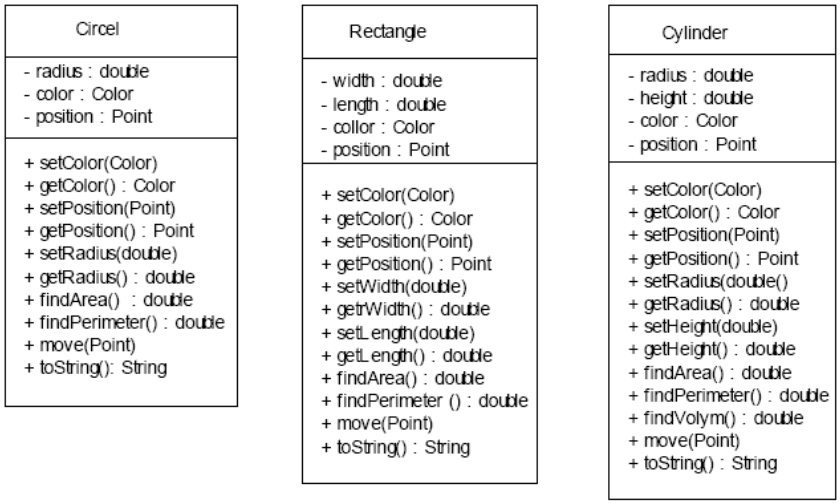
public static int getValue(Die[] theDice)

som tar ett fält med ett okänt antal tärningar av godtycklig typ, dvs. av typerna Die, Die6 eller DoubleDie, och returnerar det sammanlagda värdet av tärningarna. Det samanlagda värdet beräknas genom att addera värdet på tärningarna av typerna Die och Die6 och därefter multiplicera detta värde med värdet som finns på var och en av de tärningar som är av typen DoubleDie.

c) Skriv ett lämpligt huvudprogram för att testa klasserna och metoden.

Uppgift 3

I ett program som handhar geometriska objekt har vi identifierat följande klasser:



Som synes har klasserna en hel del gemensamt. Gör en implementering av klasserna Circel, Rectangle och Cylinder i vilken arv utnyttjas. Sträva efter att återanvända så mycket kod som möjligt.

Anmärkning: Den elegantaste lösningen fås om man använder en *abstrakt klass*. Detta begrepp har emellertid ännu inte behandlats på kursen, varför en lösning utan användning av abstrakta klasser skall göras.

Efter att abstrakta klasser behandlats på föreläsning 10, rekommenderas ni omarbeta er lösning med användning av en abstrakt klass.