



JAVAUTVECKLING – LEKTION 12

© 2016

Mahmud Al Hakim

mahmud.al.hakim@nackademin.se

www.alhakim.se

AGENDA

Arv (inheritance)

Subklasser

Abstrakta klasser och metoder

VAD ÄR ARV (INHERITANCE)

Arv är en central del av objektorientering.

Subklassen är den nya klassen som ärver av den redan existerande klassen.

Superklassen är den klass som man ärver av.

Klassen Object är alla klassers superklass.

I Java ärver en klass alltid från en och endast en klass.

Källa: <http://www.programmera.net/java/inheritance.php>

INLEDANDE EXEMPEL — KLASSEN HUS

```
public class Hus {  
    double längd;  
    double bredd;  
    int antalvåningar;  
    public double yta() {  
        return längd * bredd * antalvåningar;  
    }  
}
```

TESTPROGRAM

```
public class HusTest {
    public static void main(String[] arg) {
        Hus h = new Hus();
        h.längd = 10;
        h.bredd = 15;
        h.antalVåningar = 3;
        System.out.println("Yta: " + h.yta());
    }
}
```

KLASSEN SKOLA ÄRVER FRÅN HUS

```
public class skola extends Hus {
    int antalKlassrum;

    public double antalKlassrumPerVån() {
        return (double) antalKlassrum / antalVåningar;
    }
}
```

TESTPROGRAM

```
public class HusTest {  
    public static void main(String[] arg) {  
        Skola s = new Skola();  
        s.längd = 10;  
        s.bredd = 15;  
        s.antalvåningar = 5;  
        s.antalklassrum = 10;  
        System.out.println("Yta: " + s.antalklassrumPerVån());  
    }  
}
```

ABSTRAKTA KLASSER OCH METODER

En klass kan deklarerars som abstrakt

Inga objekt får skapas av en abstrakt klass.

Även subclasser får vara abstrakta.

En abstrakt klass får, förutom vanliga variabler och metoder, också innehålla abstrakta metoder.

Abstrakta metoder har ingen implementering.

UPPGIFT 17.2 – KLASSEN PUNKT

```
class Punkt {
    private double x, y;
    public Punkt(double xx, double yy) {
        x=xx; y=yy;
    }
    public double avstånd(Punkt p) {
        // ger avståndet mellan denna punkt och punkten p
        return Math.sqrt((x-p.x)*(x-p.x)+(y-p.y)*(y-p.y));
    }
}
```

TESTPROGRAM

```
public static void main(String[] args) {
    Punkt p1 = new Punkt(1,1);
    Punkt p2 = new Punkt(1,5);
    System.out.println("Avståndet mellan p1 och p2 är "
        + p1.avstånd(p2));
}
```

UPPGIFT 17.2 – KLASSEN FIGUR

```
public abstract class Figur {
    public Punkt startpunkt;
    public Figur(double x, double y) {
        startpunkt = new Punkt(x,y);
    }
    public abstract double area();
}
```

UPPGIFT 17.2 – KLASSEN CIRKEL

```
class Cirkel extends Figur {
    private double radie;
    public Cirkel(double rad, double x, double y) {
        super(x,y);
        radie=rad;
    }
    @Override
    public double area() {
        return Math.PI*radie*radie;
    }
}
```

TESTPROGRAM

```
public static void main(String[] args) {  
    // Figur f = new Figur();    // FEL!  
    Cirkel c = new Cirkel(10,1.0,2.0);  
    System.out.println("Area: " + c.area());  
}
```