



## Programmering för språkteknologer I

Markus Saers  
markus.saers@lingfil.uu.se  
9-2040  
stp.ling.uu.se/~markuss/vt09/pst1

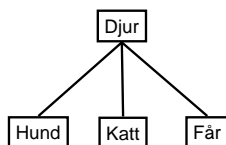


## Innehåll

Arv, arvshierarkier  
Gränssnitt  
Kapitel 9, 15



## Arvshierarki



## Arv

Låter oss skapa en basklass (Djur)  
Som sedan kan byggas ut (Får/Katt/...)  
Istället för att arbeta med specifika djur  
kan vi arbeta med det bredare konceptet  
Djur (konceptet Hund kan jaga alla Djur)



## Arv

Ibland orealistiskt att skapa objekt av basklassen

Hur skapar man ett "djur"?

Hur låter ett "djur"?

Abstrakta basklasser kan beskriva en del av konceptet, men lämna till de konkreta klasserna att implementera

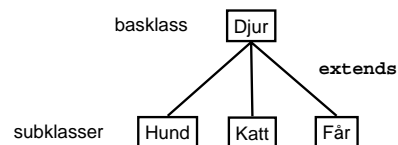
Alla djur kan låta

Hundar låter "voff"

Katter låter "mjau"



## Arvshierarki





## Abstraktioner

En klass som är abstract

```
public abstract class KlassNamn { ... }
```

Får ha abstrakta metoder

```
public abstract void metodNamn();
public abstract int funktionsNamn();
```

Kan ha konstruktor

```
public KlassNamn() { ... }
```

Som inte kan användas av andra än  
subklasserna!

Kan inte instansieras

Går inte att skapa objekt av klassen



## this/super

Nyckelordet **this** kan användas för att referera till attribut och metoder i det aktuella objektet

Nyckelordet **super** kan användas för att referera till attribut och metoder i det aktuella objektets basclass

Fungerar i princip lika

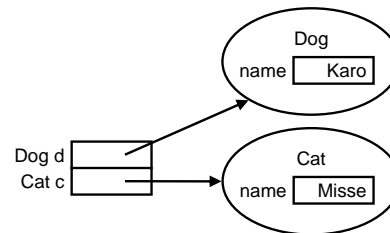


## Vad händer när en Hund jagar ett Djur?

```
Dog d = new Dog("Karo");
Cat c = new Cat("Misse");
```



## Vad händer när en Hund jagar ett Djur?



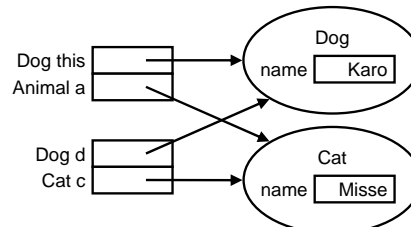
## Vad händer när en Hund jagar ett Djur?

```
Dog d = new Dog("Karo");
Cat c = new Cat("Misse");
d.chase(c);
```

```
public class Dog {
    public void chase(Animal a) {
        System.out.println(getName() +
            " chases " + a.getName());
    }
}
```



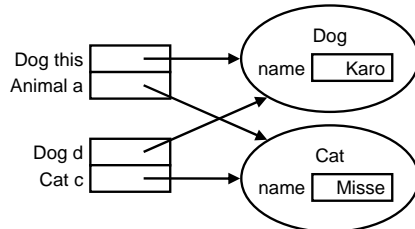
## Vad händer när en Hund jagar ett Djur?





## Vad händer när en Hund jagar ett Djur?

```
getName()
a.getName()
```



## Arv

Låter subclasserna ära

Metoder (konkreta och abstrakta)  
Attribut

Subklasserna har garanterat alla metoder som finns i basklassen

Eftersom man lyckades skapa ett objekt har alla abstrakta metoder i basklassen konkretiserats



## Flera roller?

Människor

"Djur"

Har sysselsättning  
(arbete/arbetslös/student)

Behöver (åtminstone) två baskoncept

Fysiologiskt

Socialt

Det går bara att ära från en klass i Java



## Hur ger man en klass flera roller?

Gränssnitt!

Definieras med nyckelordet interface

Betyder gränssnitt på engelska

Innehåller en lista med metodhuvuden

Klasser som "implementerar" gränssnittet måste innehålla implementationer av alla gränssnittets metoder



## Sociala varelser

```

public interface SocialBeing {
    String getSocialFunction();
    SocialBeing getBoss();
}
  
```



## Människor

```

public class Human extends Animal implements SocialBeing {
    private SocialBeing boss;
    private String socialFunction;
    public Human(String name) {
        super(name);
        socialFunction = "unemployed";
        boss = this;
    }
    public String getSocialFunction() {
        return socialFunction;
    }
    public SocialBeing getBoss() {
        return boss;
    }
    public void employ(SocialBeing boss, String function) {
        this.boss = boss;
        this.socialFunction = function;
    }
}
  
```