

Arv m.m. i Java: sub- och superklasser

Alla klasser har klassen `Object` som superklass:

```
package people;
class Person {
    String name;
    public void setName(String n) {
        name = n;
    }
    public String toString() {
        return name;
    }
}
```

Vi kan skapa en subclass – det reserverade ordet är `extends` – till klassen `Person` på detta sätt:

```
package people;
class Student extends Person {
    int creditPoints;
    String majorSubject;
    public void setCreditPoints(int p) {
        creditPoints = p;
    }
    public int getCreditPoints() {
        return creditPoints;
    }
    public void setMajorSubject(String s) {
        majorSubject = s;
    }
    public String toString() {
        return name + " (" + majorSubject + ") ";
    }
}
```

Notera att `toString`-metoden omdefinieras i subclassen. Detta innebär att den superklassens specifikation skrivs över. Man säger att subclassens `toString`-metod ”overrides” den i superklassen.

Vi hade också kunnat skriva "class Person extends Object" i första exemplet. Det hade betytt samma sak.

Arvshierarkin måste vara av trädstruktur. En subclass har en bestämd (omedelbar) superklass.

Abstrakta och finala klasser

Abstrakta klasser: Om man har en klass som man bara vill ha för att andra andra klasser skall ärva från den, kan man deklarera den som `abstract`. Det betyder att det inte är tillåtet att skapa instanser av den.

Finala klasser: Med det reserverade ordet `final` anger man att en klass är maximalt specifik i den bemärkelsen att subclasser till den inte får införas.

Interface (gränssnitt)

Ett gränssnitt liknar i mycket en abstrakt klass, men kan bara innehålla metoder som är abstrakta, d.v.s. sådana som saknar implementationsdel. Detta är exemplet från laboration 3:

```
package simpleSorting;
import java.util.*;
public interface LinkedListOfStringsSorting {
    public LinkedList<String> sort(LinkedList<String> input);
}
```

Man säger att klasser **implementerar** (implements) ett gränssnitt. Man får skriva på detta sätt (exempel från laboration 3):

```
public class InsertionSort implements LinkedListOfStringsSorting {
```

Förutom abstrakta metoder får gränssnitt innehålla finala klassvariabler. En `final` variabel är en vars värde inte kan ändras.

En klass får implementera ett godtyckligt antal gränssnitt. Det är en väsentlig skillnad gentemot arv, då en klass ärver från maximalt en omedelbar superklass.