**Eigenschaften der Vererbung**

- eine Subklasse erbt Attribute und Methoden von ihrer Superklasse

- es entsteht eine Vererbungshierarchie, die theoretisch beliebig tief geschachtelt sein kann

- Ziel und Vorteil: bestehender Programmcode kann wieder verwendet werden (Reuse)

- durch die Mehrfachvererbung erbt eine Subklasse von mehreren Superklassen

- Superklassen sind Generalisierungen ihrer Subklassen

- Subklassen sind Spezialisierung ihrer Superklassen

Darstellung der Vererbung in UML ist ein einfacher Pfeil

**Vererbung in Java**

- Java unterstützt nur einfache nicht mehrfache Vererbung

- Syntax: class Subklasse extends Superklasse { }

- Alle Methoden und Attribute außer die Konstruktoren werden vererbt

- In der Subklasse kann die Funktionalität der Superklasse durch hinzufügen, überladen und überschreiben von Methoden verändert werden

**Die Superklasse Objekt**

- ist Wurzel der Klassenhierarchie in Java (Alle Klassen sind Subklassen von Objekt)

- deshalb haben allen Objekte die Methoden

Objekt(); parameterloser Konstruktor

toString(); Konvertierung eines Objekts in einen String

equals(), Vergleicht zwei Objekte mit einander

hashCode(), Berechnet den Hash Code eines Objekts

clone(); Kopiert Objekte

finalie(); Destruktor

**Methoden überschreiben**

- Methoden die in Superklassen definiert sind können in Subklassen überschrieben werden

- beim dynamischen binden sucht die JVM zunächst im Referenzdatentyp nach einer passenden Methode und sucht von dort nach oben weiter

- Überschreiben kann mit zwei Sichtbarkeitsmodifikatoren verhindert werden:

Private Methode ist in der Subklasse nicht sichtbar

Final Verhindert das explizite Überschreiben

Weitere Modifikatoren

Abstract - Klassen, die nicht ausreichend spezialisiert sind, dienen nur als Vorlage für Subklassen

- können keine Objekte erzeugen

- beinhaltet eine Klasse eine abstract Methode wird sie abstract

- beinhalten nur Methodenrumpf und keine Implimentierung

- Subklassen müssen die abstract Methoden implementieren

Darstellung in UML durch kursive Schrift

Final - Final Methoden werden nicht vererbt oder überschrieben

- Final Attribute sind Konstanten deren Wert sich nicht ändert

**Bedeutung von this und super**

- mit super kann per Punktnotation auf die Superklasse zugegriffen werden

- this ist eine Referenzvariable, die beim Objekterzeugen automatisch angelegt wird, und auf das Objekt verweist

- über die this-Referenz können eigene Methoden, Instanz- und Klassenattribute angesprochen werden

- mit Hilfe von this und super als Übergabeparameter können Konstruktoren verkettet werden

- this und super müssen dann als erste Anweisung im Konstruktor stehen

**Narrowing Cast**

- beschreibt den Wechsel von einer Sicht mit mehr Details zu einer mit weniger Details bei einem Objekt

- Objekte von Subklassen können in Referenzvariablen gespeichert werden, die vom Typ ihrer Superklasse sind

- dabei wird nur die Sicht auf die Objekte beschränkt, d.h. es sind nur Methoden und Attribute sichtbar, die in der Superklasse deklariert sind

- der Rest ist vorhanden, aber vorrübergehend ausgeblendet

- der narrowing cast ist ein wesentliches Prinzip des Vererbungskonzepts von Java und Voraussetzung für die Polymorphie

Widening Cast

- Umkehrung des narrowing cast

- Referenzen auf Objekte, die als Referenzvariable vom Typ der Superklasse sind, sollen als Referenzvariablen der Subklasse gespeichert werden

- ist eine unsicher Konvertierung, deshalb muss:

- sichergestellt werden, dass die Referenz eine Referenz auf Objekte der Subklasse ist

- instanceof muss true ergeben

- bei der Durchführung ein expliziter Cast gemacht werden



Polymorphie

- weiterer Grundsatz des objektorierntierten Programmierens

- bedeutet die Vielgestaltigkeit von Objekten

- basiert auf dynamischem Binden

- gleiche Methodenaufrufe rufen unterschiedliches Verhalten hervor