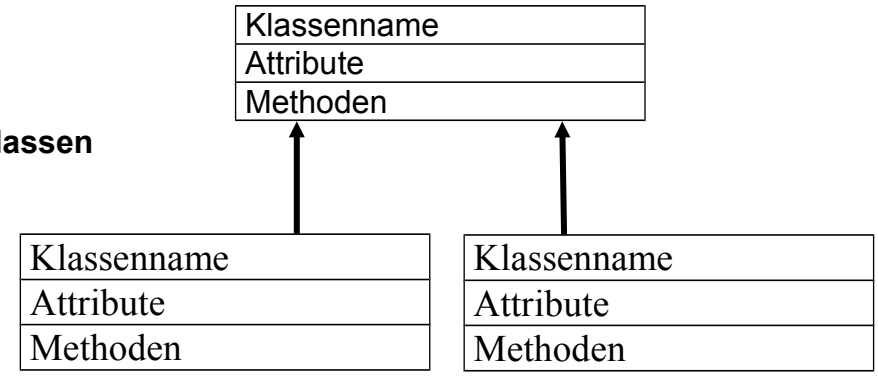
Das Prinzip der Vererbung

Eines der wichtigsten Konzepte im Zusammenhang mit Klassen ist die Möglichkeit, eine Klasse durch Vererbung von einer anderen Klasse abzuleiten. Sinn der Vererbung ist es, Klassenhierarchien aufzubauen, in denen die Gemeinsamkeiten von Klassen einer Ebene in die Klassen der vorangehenden Ebene ausgelagert werden. Klassenhierarchien zeichnen sich daher üblicherweise dadurch aus, dass die oberen Ebenen ein immer höheres Abstraktionsniveau haben, während die nachfolgenden Ebenen sich durch zunehmende Spezialisierung auszeichnen.

Einfache Vererbung

- **Basis-Klasse**
- **abgeleitete Klassen**



Es gelten folgende Regeln:

- alle Attribute werden an die abgeleiteten Klassen weitergegeben (vererbt)
- alle Methoden werden an die abgeleiteten Klassen weitergegeben (vererbt)
- Zugriffssteuerung:

Attribute oder Methoden der Basis-Klasse sind:	Funktionen der Objekte der Basis-Klasse	Funktionen der Objekte der abgeleiteten Klasse	alle Funktionen, die Objekte der Klasse kennen
public:	Zugriff möglich	Zugriff möglich	Zugriff möglich
protected:	Zugriff möglich	Zugriff möglich	Zugriff nicht möglich
private:	Zugriff möglich	Zugriff nicht möglich	Zugriff nicht möglich

- Konstruktoren / Destruktoren
  - werden an die abgeleiteten Klassen vererbt
  - jede abgeleitete Klasse kann eigene Konstruktoren und Destruktoren haben
  - Reihenfolge der Aufrufe:
    - Konstruktor der Basis-Klasse
    - Konstruktor der abgeleiteten Klasse
    - Destruktor der abgeleiteten Klasse
    - Destruktor der Basis-Klasse
  - der Standardkonstruktor der Basisklasse wird automatisch vom Compiler aufgerufen
  - reicht der Standardkonstruktor der Basisklasse nicht, um die geerbten Elemente zu initialisieren, so ruft man den entsprechenden Konstruktor der Basisklasse explizit auf.

- **Syntax in C++:**
  - **class** KLASSENNAME : **public** BASIS-KLASSENNAME
    - Bsp.:
```
class GIROKONTO : public KONTO
{
    private:
        double dGrenzbetrag;
}
```

Methoden überladen

- eine Methode der Basis-Klasse kann in der Klasse selber oder einer abgeleiteten Klasse überladen werden
- die überladene Methode muss dazu folgende Bedingungen erfüllen:
  - selber Name wie die zu überladene Methode
  - veränderte Übergabeparametertypliste

(Beispiel einer Überladung: Standardkonstruktor und weiterer Konstruktor in einer Klasse.)

Methoden überschreiben

- eine Methode der Basis-Klasse kann in einer abgeleiteten Klasse überschrieben werden
- die überschriebene Methode der abgeleiteten Klasse muss dazu folgende Bedingungen erfüllen:
  - selber Name wie in der Basis-Klasse
  - selber Rückgabetyp wie in der Basis-Klasse
  - selbe Übergabeparametertypliste wie in der Basis-Klasse

  die Methode der Basisklasse kann danach nur noch über die vollständige Qualifizierung mit ihrem Klassennamen aufgerufen werden.

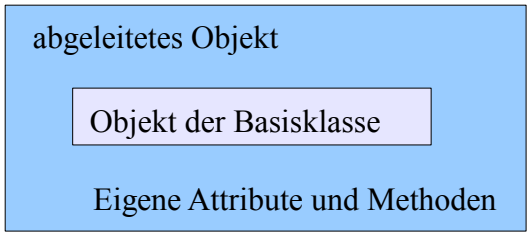
(Beispiel setKontostand() wird in GIROKONTO überschrieben, da der Grenzbetrag als zusätzliches Attribut der abgeleiteten Klasse bei jeder Änderung überprüft werden muss.)

Aufgabe:
Erstellen Sie die Klassenhierarchie und Testen Sie die Zugriffsmöglichkeiten auf die einzelnen Elemente.

Polymorphismus (=Vielgestaltigkeit) und virtuelle Methoden

Unter Polymorphie versteht man zum einen die oben erwähnte Möglichkeit Methoden zu überschreiben:
eine Methode – mehrere Implementierungen
Zum anderen versteht man darunter auch die Behandlung eines Objekts einer abgeleiteten Klasse als Objekt der Basisklasse:
ein Objekt – mehrere Datentypen

Was darunter zu verstehen ist, wird im Folgenden erläutert:
In Objekten einer abgeleitete Klasse sind die geerbten Elemente (Attribute und Methoden) aus der Basisklasse als Unterobjekt vorhanden, so dass das abgeleitete Objekt als ein Objekt der



Dies bedeutet, dass alle Objekte der abgeleiteten Klassen in einem Array von Objekten der Basisklasse oder noch besser in einem Array von Zeigern auf Objekte der Basisklasse abgelegt werden können. Um dabei auch noch Zugriff auf die „richtigen“ Methoden zu haben, verwendet man abstrakte (virtuelle) Methoden.

Virtuelle Methoden

eine abstrakte bzw. rein virtuelle Methode

Beispiel:

```
class KONTO{... double getKontostand(void);                    // gibt den Kontostand zurück
                virtual void setKontostand(double dBetrag)=0;  //  ändert den Kontostand
            ...}
class GIROKONTO: public KONTO{...virtual setKontostand(){...}...}   //abgeleitete Klasse
class SPARKONTO: public KONTO{...virtual setKontostand(){...}...}   //abgeleitete Klasse
int main(void)
{
        KONTO *pKonten[2];

        pKonten[0]= new SPARKONTO(50);
        pKonten[1]=new GIROKONTO(120);
        pKonten[0]->setKontostand();          // Aufruf der Methode in SPARKONTO
        pKonten[1]->setKontostand();          // Aufruf der Methode in GIROKONTO
        delete []Konten;
}
```

Durch das Schlüsselwort virtual wird für abgeleitete Klassen, die diese Methode überschreiben, die Version aus der abgeleiteten Klasse aufgerufen, auch dann, wenn auf das Objekt über einen Zeiger vom Typ der Basisklasse zugegriffen wird.(späte Bindung – erst zur Laufzeit)

Abstrakte(rein virtuelle) Methode

Eine abstrakte Methode besitzt keinen Definitionsteil. Sie gibt nur die Schnittstelle vor, die an andere Klassen vererbt wird. Erkennbar ist dies an der Zuweisung der Null.

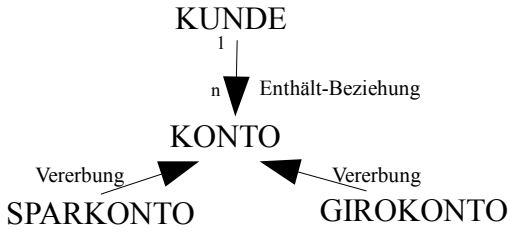
Abstrakte Klasse

Man spricht von einer abstrakten Klasse, wenn sie mindestens eine rein virtuelle Methode enthält.

- von einer abstrakten Klasse lassen sich keine Objekte bilden.
- wird eine abstrakte Methode in der abgeleiteten Klasse nicht überschrieben, so ist sie auch dort abstrakt und damit ist auch die abgeleitete Klasse abstrakt.

Aufgabe:

1) Erstellen Sie KONTO als abstrakte Klasse. Erstellen Sie eine Klasse KUNDE. Verwalten Sie die Konten für die Kunden in einem Feld von bis zu 4 Zeigern auf KONTO. Zur Laufzeit wird entschieden, ob ein Girokonto oder ein Sparkonto angelegt wird.



2) Geben Sie das komplette UML-Klassendiagramm für dieses Projekt an.
3) Welche Probleme tauchen beim Löschen einzelner Konten eines Kunden auf? Welche Datenstruktur könnte bei der Speicherung der Konten im Gegensatz zum Array von Vorteil sein?
4) Die STL (Standard Template Library) in C++ stellt Datenstrukturen zur Verfügung. Erarbeiten Sie sich selbstständig die Benutzung der Datenstruktur Liste aus der STL für die Verwaltung der Konten.