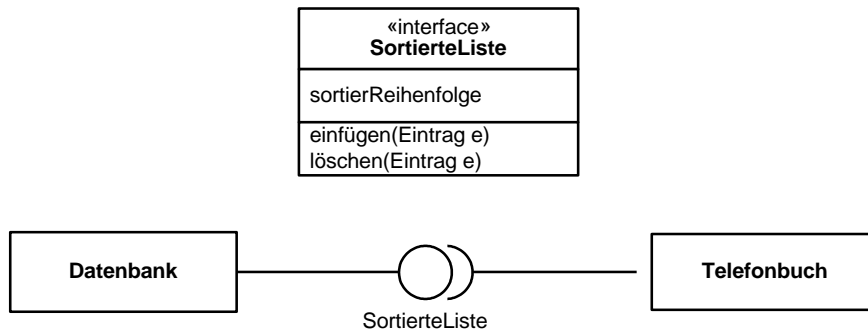


## Codebeispiele für Schnittstellen

### Kapitel 4: Klassendiagramm



### Umsetzung in C++

C++ verfügt über kein separates Konstrukt zur Definition von Schnittstellen. Stattdessen werden Schnittstellen als Klassen definiert, deren Operationen durch das Schlüsselwort `virtual` eingeleitet und formal durch die Zeichenkette `=0` abgeschlossen werden, auch wenn es sich dabei um keine Zuweisung handelt.

Diese Formulierung stellt sicher, dass innerhalb der Schnittstellenklasse keine Methoden hinterlegt werden können. Gleichzeitig können Klassen, die mit dieser Syntax ausgezeichnet wurden, keine Objekte erzeugen.

```

class SortierteListe {
    int sortierReihenfolge;
    public:
        virtual void einfuegen(Eintrag e)=0;
        virtual void loeschen(Eintrag e)=0;
};
class Datenbank : public SortierteListe {
    public:
        virtual void einfuegen(Eintrag e) {
            //...
        }
        virtual void loeschen(Eintrag e) {
            //...
        }
};
    
```

### Umsetzung in Java

Java unterstützt mit dem Schlüsselwort `interface` explizit die Möglichkeit, Schnittstellen zu definieren. Abgesehen von diesem Schlüsselwort entspricht die Definition einer Schnittstelle syntaktisch der einer Klasse.

Im Unterschied zur Klasse darf eine Schnittstelle jedoch nicht über Attribute, sondern lediglich über Konstanten verfügen. Außerdem dürfen für die deklarierten Operationen keine Methoden

hinterlegt werden. Alle deklarierten Operationen müssen mit der Sichtbarkeitseinschränkung `public` versehen sein.

Eine Klasse erklärt mit dem Schlüsselwort `implements`, dass sie eine Schnittstelle implementieren will. Darauf folgt durch Kommata getrennt die Aufzählung der durch die Klasse implementierten Schnittstellen.

```
interface SortierteListe {
    int sortierReihenfolge=0;
    public void einfügen(Eintrag e);
    public void löschen(Eintrag e);
}

class Datenbank implements SortierteListe {
    public void einfügen(Eintrag e) {
        //...
    }
    public void löschen(Eintrag e) {
        //...
    }
}
```