Hochschule Niederrhein	PE2 / SE2	Dozenten
Fachbereich 03	Übung 8	Rethmann / Brandt
Bachelor Informatik	SoSe 2021	Stockmanns / Quix

Vererbung und Polymorphie in C++

Lernziele

Hier soll das Thema *Polymorphie* vertieft und angewendet werden.

Aufgabe 8:

Eine Bank verwaltet Konten, wobei jedes Konto einer Person zugeordnet ist. Leiten Sie von der Klasse `Konto` zwei Klassen `Girokonto` und `Sparkonto` ab. Folgende Funktionen haben alle Kontenarten gemeinsam:

```
void hebeAb(int betrag, Datum d);
void zahleEin(int betrag, Datum d);
int kontostand();
string kontoauszug();
```

Bei einem Sparkonto darf nur dann Geld abgehoben werden, wenn der Kontostand dadurch nicht negativ wird. Bei einem Girokonto darf der Kontostand zwar negativ werden, aber nur bis zum angegebenen `dispo`-Wert, dem Kreditlimit. Werden diese Regeln verletzt, soll eine Exception geworfen werden.

Das Guthaben auf einem Sparkonto wird verzinst, beim Girokonto gibt es dagegen nur einen Sollzins, also einen Zinssatz für Überziehungen. Außerdem stellt das Girokonto eine weitere Methode bereit:

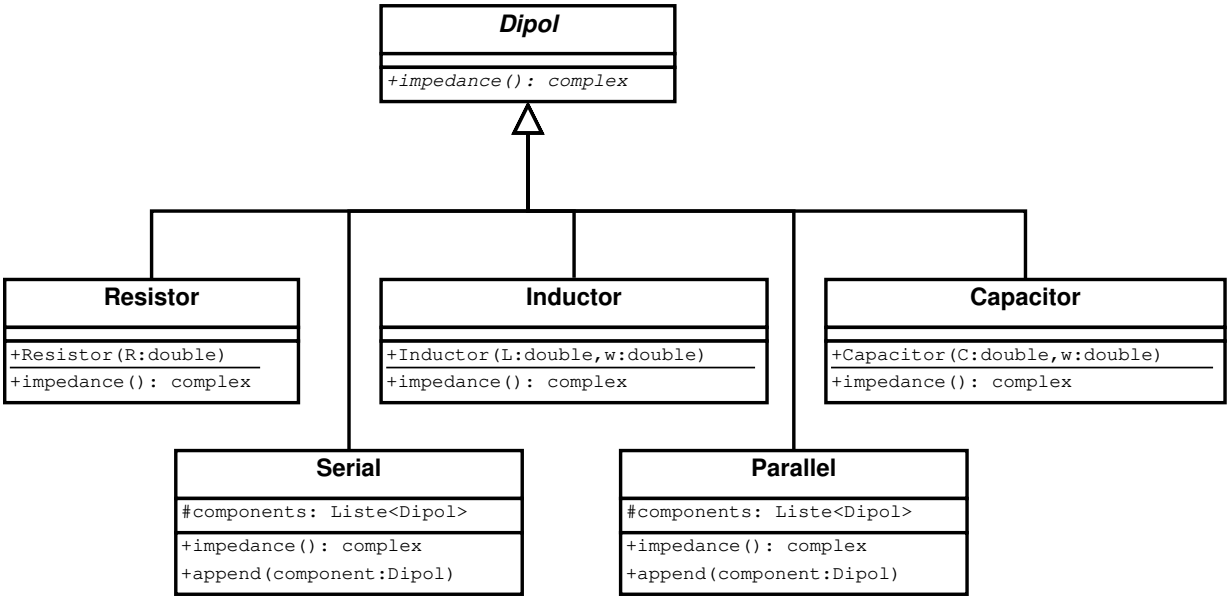
```
void ueberweise(string kontonr, int betrag, Datum d);
```

Implementieren Sie obige Klassen. Überlegen Sie sich, welche Attribute die Klassen haben und welche davon in der Basisklasse deklariert werden können. Welche der Methoden müssen als `virtual` deklariert und in einer Unterklasse überschrieben werden? Welche Methoden sind abstrakt? Wie werden Kontobewegungen vermerkt?

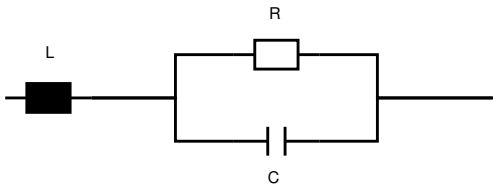
Die Klasse `Datum` stellen wir zur Verfügung. Wie immer gehört das Erstellen von Testtreibern und Testfällen zur Aufgabe, um die Korrektheit Ihrer Implementierung nachzuweisen. Die von uns implementierten Testfälle sind nicht ausreichend.

Zusatz-Aufgabe Z8:

Implementieren Sie eine Basisklasse `Dipol`, die eine rein virtuelle Methode `impedance` bereitstellt. Leiten Sie von dieser Klasse drei Klassen `Resistor`, `Inductor` und `Capacitor` ab, die jeweils die Methode `impedance` implementieren.



Nutzen Sie unsere generische Klasse `Liste` aus der Vorlesung, um eine Klasse `Serial` und eine Klasse `Parallel` zu implementieren, die eine Serien- bzw. Parallelschaltung von Zweipolen realisieren und ebenfalls die Klasse `Dipol` implementieren.



Die obige Schaltung könnte wie folgt realisiert werden:

```
int main(void) {
    Resistor r(20);
    Capacitor c(2e-6, 1e3);
    Parallel par;
    par.append(&r);
    par.append(&c);

    Inductor l(0.2, 1e3);
    Serial ser;
    ser.append(&l);
    ser.append(&par);
    cout << ser.impedance() << endl;
```

Wie ist das Programm `circuit` aus Zusatz-Aufgabe Z6 zu ändern, um den dort realisierten Schaltkreis mittels der hier implementierten Klassen zu modellieren?

Wie immer gehört das Erstellen von Testtreibern und Testfällen zur Aufgabe, um die Korrektheit Ihrer Implementierung nachzuweisen.