

Informatik IIa - Praktikum

Oliver Jack
Fachhochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Sommersemester 2010

Praktikum 6: Klassen

Operator-Overloading

Aufgabe 1 (Addition und Subtraktion zweier Vektoren):

- Betrachten Sie die Klasse `drei_d` aus der Vorlesung, die dreidimensionale Vektoren definiert.
- Ändern Sie die Klasse so, dass die Vektorkomponenten vom Typ `float` sind.
- Entfernen Sie aus der Klasse die Inkrement-Operatoren `++`.
- Erweitern Sie die Klasse um die Operatoren
 - `+` für die Addition zweier Vektoren
 - `-` für die Subtraktion zweier Vektoren
- Erweitern Sie die Klasse um eine Funktion, die den Betrag des Vektors ausgibt.

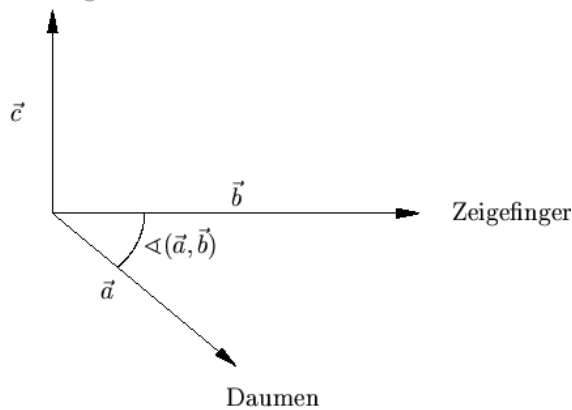
Aufgabe 2 (Das Kreuzprodukt von Vektoren):

- Erweitern Sie die Klasse `drei_d` um den Operator `*`, der das Kreuzprodukt zweier Vektoren berechnet.
- Implementieren Sie den Operator so, dass eine Verkettung von `*`-Operatoren möglich ist.

Das Kreuzprodukt zweier Vektoren $\vec{a} \times \vec{b}$ ist folgendermaßen definiert:

$$|\vec{c}| = |\vec{a}||\vec{b}| \sin(\angle(\vec{a}, \vec{b})) = \vec{a} \times \vec{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \times \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \det \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \\ \det \begin{pmatrix} a_3 & b_3 \\ a_1 & b_1 \end{pmatrix} \\ \det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

Veranschaulichung (Rechte Hand-Regel)
Mittelfinger



Aufgabe 3 (Bruchrechnung): Erstellen Sie eine Klasse zur Darstellung und arithmetischen Behandlung von Brüchen.

- Deklarieren Sie dazu eine Klasse `Bruch` mit einem Zähler und einem Nenner vom Typ `long`.
- Deklarieren Sie einen Konstruktor mit den zwei Parametern Zähler und Nenner, die die Default-Werte 0 und 1 erhalten.
- Deklarieren Sie die Operatoren `-` (unär), `+=`, `-=`.
- Deklarieren Sie die Operatoren `+`, `-`, `*`, `/`.
- Deklarieren Sie als `friend`-Funktionen die Ein- und Ausgabe-Operatoren `>>`, `<<`. Hinweis: Definieren Sie die Parameter der Operatoren wenn immer möglich als Referenzen auf konstante Objekte.

- f) Implementieren Sie die Klasse. Beachten Sie dabei, dass der Nenner eines Bruchs immer positiv sein soll (relevant bei Konstruktor und Ausgabe).
- g) Geben Sie eine Fehlermeldung aus, falls der Nenner gleich 0 ist.
- h) Rechenregeln für Brüche

$$\begin{aligned}\frac{a}{b} + \frac{c}{d} &= \frac{a*d + c*b}{b*d} \\ \frac{a}{b} - \frac{c}{d} &= \frac{a*d - c*b}{b*d} \\ \frac{a}{b} * \frac{c}{d} &= \frac{a*c}{b*d} \\ \frac{a}{b} / \frac{c}{d} &= \frac{a*d}{b*c}\end{aligned}$$

- i) Testen Sie Ihre Bruch mit folgendem Programm:

```
int main ()
{
    Bruch a(4,7), b(2);
    cout << endl << "Einige Testausgaben:"
         << endl << endl;
    cout << "a=" << a << endl;
    cout << "b=" << b << endl;
    cout << "a+b=" << (a + b) << endl;
    cout << "a-b=" << (a - b) << endl;
    cout << "a*b=" << (a * b) << endl;
    cout << "a/b=" << (a / b) << endl;
    a += Bruch(1,2);
    cout << "a+=1/2; a=" << a << endl;
    a -= Bruch(1,2);
    cout << "a-=1/2; a=" << a << endl;
    cout << "-b=" << -b << endl;
    cout << "Bitte einen Bruch eingeben:";
    cin >> a;
    cout << "Ihre Eingabe:" << a << endl;
    return 0;
}
```

Aufgabe 4 (Kernbruch (Optional)): Erweitern Sie die arithmetischen Operatoren der Klasse `Bruch`, in dem Sie das Ergebnis als Kern-Bruch zurückgeben.

- Normieren Sie den Bruch bei jeder Operation, in dem Sie den Kernbruch berechnen (Kürzen).
- Zur Erinnerung: Sie kürzen einen Bruch, in dem Sie Zähler und Nenner durch deren größten gemeinsamen Teiler `ggT` teilen.
- Größter gemeinsamer Teiler: Algorithmus von Euklid (steinalt, älter als 2000 Jahre)

$$\text{ggT}(a, b) = \begin{cases} a & \text{falls } b = 0 \\ \text{ggT}(b, a \bmod b) & \text{andernfalls} \end{cases}$$