

Aufgabenblatt 3 Überladen von Operatoren

Aufgabe 1

Gehen Sie von der gegebenen Skizze der Klasse Vektor3d aus:

```
class Vektor3d {  
    float x, y, z;  
public:  
    Vektor3d(float a, float b, float c) { x = a; y = b; z = c; }  
    Vektor3d() { x=0; y=0; z=0; }  
};
```

- Implementieren Sie den parameterbehafteten Konstruktor so, dass der parameterlose Konstruktor überflüssig wird. Benutzen Sie die Listenschreibweise für die Initialisierung des parameterbehafteten Konstruktors
- Überladen Sie den Operator + als friend Funktion. $v3=v1+v2$ bedeutet: $x3=x1+x2$, $y3=y1+y2$, $z3=z1+z2$ (binärer Operator +) mit $v1=(x1,y1,z1)$, $v2=(x2,y2,z2)$, $v3=(x3,y3,z3)$
- Überladen Sie den Operator – als friend Funktion: $v3=v1-v2$ bedeutet: $x3=x1-x2$, $y3=y1-y2$, $z3=z1-z2$ (binärer Operator -)
- Ergänzen Sie als weitere Varianten für Überladung die Verknüpfungen: $\text{Vektor3d} * \text{float}$ ergibt Vektor3d und $\text{Vektor3d} * \text{Vektor3d}$ (inneres Produkt) ergibt float.
- Überladen Sie den Vergleichsoperator als friend Funktion. Übergeben Sie die Parameter per Referenz. Achten Sie auf minimale Zugriffsrechte.
- Überladen Sie den Präfixoperator ++: $++v1=(++x1,++x2,++x3)$
- Überladen Sie den Postfixoperator ++: $v1++=(x1++,x2++,x3++)$
- Überladen Sie den Operator << bezüglich der Klasse Vektor3d wobei $\text{cout} << v1$ bedeutet: v1 wird in geeignetem Format auf der Konsole ausgegeben. Das Resultat der binären Verknüpfung mit dem Operator << zwischen einem Objekt cout (vom Typ ostream) und v1 vom Typ vektor3d ist cout (vom Typ ostream)! Für was brauchen wir das?
- Wählen Sie einen weiteren Operator, den Sie für Vector3d überladen. Am besten wählen sie einen praktisch sinnvollen Fall.
- Überlegen Sie sich einen Testplan. Jede Funktionalität sollte getestet sein. Dokumentieren Sie Ihre Tests. Testfall: Was wollen Sie testen? Test: wie wollen Sie das testen. Was erwarten Sie als Ergebnis des Tests? Was beobachten Sie als Ergebnis des Tests? Fassen Sie alle Tests in einem Dokument zusammen.

Aufgabe 2

Ergänzen Sie die TaskQueue in Aufgabenblatt 2, Aufgabe 2 um einen privaten Zuweisungsoperator, um die Zuweisung zu verhindern. Ergänzen Sie den vorhandenen Testplan um geeignete Testfälle.

Die Programme sind in Felix abzugeben. Dokumentierten Sie die C++-Texte, z.B mit Kommentaren. Der Abgabetermin ist im Felix Kurs hinterlegt.