

Beispiel 1: Template Vector (24 Punkte)

Implementieren Sie ein Klassentemplate für Vektoren, wobei auch die Operatoren[], = und << sowie >> zu implementieren sind.

```
template <class T>
class Vektor {
private:
    T *elem;
    int vsize;
public:
    Vektor(int size);
    ~Vektor(void);
    T& operator[](int pos);
    Vektor<T>& operator=(Vektor<T>&v);
    friend ostream& operator<<(ostream &os, Vektor<T> &v);
    friend istream& operator>>(istream &is, Vektor<T> &v);
    void sort(Vektor<T>& v);
}
```

Testen Sie Ihr Klassentemplate mit den Datentypen Integer, Char sowie mit komplexen Zahlen.

```
class Complex {
private:
    int imag;
    int real;
public:
    friend ostream& operator<<(ostream &os, Complex &v);
    friend istream& operator>>(istream &is, Complex &v);
}
```

Achten Sie darauf, daß bei der Ein- und Ausgabe sowohl der Imaginär- als auch der Realteil zu berücksichtigen ist. Die Ein- und Ausgabe soll generell die Form haben:

Eingabe: **x. Element: wert**
 bzw. **x.Element: real: wert**
 imag: wert

Wobei x einen Wert von 1 bis vsize darstellt und wert der einzulesende Wert ist.

Ausgabe: **x. Element: wert**
 bzw. **x.Element: realwert + imagwert i**

Wichtig: Bereichsüber- und unterschreitungen sollen mittels Exception Handling abgefangen werden.

Achten Sie bei der Implementierung besonders darauf, daß ungültige Parameterwerte nicht zu ungewünschten Effekten (Absturz, Speicherkorruption) führen können. Testen Sie Ihr Modul entsprechend, um sicherzustellen, daß Sie dieses Ziel auch erreicht haben. Sie können Ihren Testtreiber hart codieren oder ein Testprogramm dazu schreiben.

Achten Sie bitte darauf, bei Ihrer schriftlichen Abgabe im Header Ihre Account-Nummer anzugeben, damit wir Ihre elektronische Abgabe finden können.

Abgabe: bis 31.5. 23:59 Uhr elektronisch (Verzeichnis uebung7 einrichten!) und am 1.6. bzw. 2.6. jeweils zu Beginn des Praktikums auf Papier (Listing der Quelldateien).