

Programmieren - Blatt 5

Diskussion:

- *Vorbereitung Aufgabe 1 – Klassen:*
 - Was ist eine Klasse? Was ist ein Objekt? Was sind Membervariablen und Membermethoden?
 - Was bedeutend die Keywords **public** und **private**?
 - Was ist ein Konstruktor? Was ist wenn man keinen Konstruktor schreibt?
 - Was ist ein Destruktor? Muss man immer einen Destruktor schreiben?
 - Wie kann man auch bei Klassen Deklaration und Definition trennen?
- *Vorbereitung Aufgabe 2 – Vererbung:*
 - Wie wird eine Klasse B als Unterklasse einer anderen Klasse A definiert? Wie sieht dann die Definition des Konstruktors von B aus?
 - Schauen Sie sich die Datei *test_overloading.cpp* an. Was passiert, wenn man das Programm laufen lässt? Warum? Was hätten Sie erwartet? Wie kann man das gewünschte Verhalten erreichen?
 - Auf der Homepage finden Sie im Material Dateien *A.h*, *B.h* und *main.cpp*. Schauen Sie sich die Dateien an und geben Sie für jede Zeile in der Funktion `B::foo()` und der `int main()` an, ob der jeweilige Befehl funktioniert und warum (bzw. warum nicht).
- *Vorbereitung Aufgabe 3 – Überladung von Operatoren:*
 - Was meint man mit der Überladung von Operatoren? (Kann man auch Funktionen überladen?)
 - Was sind die “3 Basic Rules” bei der Operator-Überladung?
 - Warum sollte man z.B. nicht nur `+`, sondern auch `+=` überladen (oder umgekehrt)?
 - Was machen die folgenden Zeilen Code?

```
1 std::ostream& operator<<(std::ostream& os, const Complex& c)
2 {
3     // write obj to stream
4     os << c.get_real() << ", " << c.get_imag();
5     return os;
6 }
```

(Infos: <http://stackoverflow.com/questions/4421706/operator-overloading/4421708#4421708>)

Präsenz-Aufgabe 1 (Klassen)

In dieser Aufgabe soll eine Art Inventur-Liste für Computer angelegt werden.

Schreiben Sie eine Klasse `Computer`, welche

- die Größe des Arbeitsspeichers, die Größe der Festplatte und den aktuellen Zustand (ein/aus) beinhaltet,
- einen Konstruktor hat, der Arbeitsspeicher- und Festplattengröße als Argumente nimmt. Beim Anlegen eines Computers soll der Zustand immer auf "aus" gesetzt werden.
- Funktionen `turn_on` und `turn_off` hat, mit denen der aktuelle Zustand geändert werden kann,
- eine Funktion `void print_info()` hat, die alle Informationen (Arbeitsspeicher, Festplatte, Zustand) zum Computer ausgibt.

Legen Sie im Hauptprogramm verschiedene Computer an, geben Sie jeweils die Informationen darüber mittels `print_info` aus und testen Sie auch Ihre Funktionen `turn_on`, `turn_off`.

Präsenz-Aufgabe 2 (Klassen, Vererbung)

1. Leiten Sie von Ihrer Klasse `Computer` eine Klasse `Notebook` ab, die zusätzlich die Information verwaltet, ob der Laptop an das Stromnetz angeschlossen ist oder auf Batterie läuft.

Schreiben Sie Funktionen zum Ändern der Stromversorgung.

Schreiben Sie auch hier eine Funktion `void print_info()`, welche nun zusätzlich noch die Information über die Stromversorgung ausgeben soll.

Legen Sie im Hauptprogramm verschiedene Notebooks an und testen Sie Ihre Funktionen.

2. Rufen Sie im Hauptprogramm die folgende Funktion für verschiedene Computer und Notebooks auf:

```
void
my_current_computer(Computer& computer){
    cout << "I'm working on a ";
    computer.print_info();
}
```

Was für ein Verhalten beobachten Sie? Haben Sie das erwartet?

3. Legen Sie nun im Hauptprogramm einen Vektor `vector<Computer>` an und füllen Sie ihn mit verschiedenen Computern und Notebooks. Rufen Sie `print_info` für alle Einträge des Vektors auf. Was sehen Sie? Wieso?

Was passiert, wenn Sie einen `vector<Computer*>` verwenden?

Präsenz-Aufgabe 3 (Klassen, Operator-Überladung)

Bearbeiten Sie Aufgabe 1 auf dem Übungsblatt 8 von Herrn Lachmann, also:

- a) Schreiben Sie eine Klasse `Complex` zur Verwaltung komplexer Zahlen mit den benötigten Member-Variablen und Konstruktoren, sowie einer geeigneten Ausgabe-Funktion.
- b) Überladen Sie die Operatoren `+=`, `+`, `*=`, `*`.
- c) Testen Sie Ihre Klasse in einem Hauptprogramm.
- d) * Fügen Sie noch eine Funktion `absval()` hinzu, welche den Betrag einer komplexen Zahl berechnet und testen Sie auch diese in Ihrem Hauptprogramm.