

Praktikum ‚Objektorientierte Programmierung‘

Aufgabenblatt 5

Für alle Klassen müssen Deklaration und Implementierung getrennt sein. Bei der Übersetzung werden nicht nur Fehler, sondern auch *Warnungen* und *Verbesserungshinweise* angezeigt. Ignorieren Sie das nicht, sondern nutzen Sie diese Hinweise, um Ihren Code zu verbessern. *Ergreifen Sie aber nur Maßnahmen, die Sie auch verstehen.*

Aufgabe 1:

Das Gesetz von Demeter (engl. *Law of Demeter*) ist eine der vielen Regeln, die es in der objektorientierten Softwareentwicklung gibt. Die armen Menschen, die mit Ihrer Klasse `City` arbeiten wollen, müssen sich vorab noch über die Klasse `Position` informieren, obwohl diese für sie keinen Nutzen liefert. Sie wird nur im Konstruktor genutzt. Darum verbessern Sie Ihre Klasse `City` wie folgt:

a. Der Konstruktor, den Sie in der letzten Praktikumsaufgabe entwickelt haben, ist ab sofort `private`. Sie definieren einen öffentlichen Konstruktor

```
City(const std::string& name, int x, int y, std::string *pois, int length);
```

den Sie *geschickt* implementieren.

b. Bedauerlicherweise hat sich jetzt die Schnittstelle der Klasse geändert und Sie müssen den bestehenden Test-Code ändern. Vielleicht auch mal eine wertvolle Erfahrung. Sorgen Sie dafür, dass die Tests nichts an ihrem Umfang einbüßen.

Aufgabe 2:

Entwickeln Sie für die Klasse `Position` einen Kopierkonstruktor, wenn es notwendig ist.

Aufgabe 3:

Ergänzen Sie die Klassen `Position` und `City` um einen Destruktor, wenn es notwendig ist.

Aufgabe 4:

Besteht Ihre Klasse `City` den folgenden Test?

```
std::string pois[]={"Sauron", "Minas Morgul", "Orodruin"};
hfu::City mordor("Mordor", 47, 11, pois, 3);
auto copy = mordor;
mordor.setPOI(2,"Cirith Ungol");
assert(copy.getPOI(2)=="Orodruin");
```

Falls die Klasse den Test nicht besteht,

- machen Sie sich den Test klar,
- finden Sie die Ursache und
- passen Sie die Klasse geeignet an.