

Praktikum ‚Objektorientierte Programmierung‘

Aufgabenblatt 7

Wie in der Vorlesung besprochen, empfehle ich die Compiler-Flags `-Wall` und `-Wextra`. Um möglichst im Einklang mit der Vorlesung zu sein, sollten Sie dafür sorgen, dass Ihr Compiler den C++14-Standard unterstützt. In keiner der Aufgaben darf der *globale* Namensraum genutzt werden. Definieren Sie einen eigenen Namensraum. Die Anweisung `using namespace` darf nicht genutzt werden.

Für alle Klassen müssen Deklaration und Implementierung getrennt sein. Nutzen Sie Include-Guards. 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:

Ändern Sie die Schnittstelle der Klasse `Position`. Die Methode `compare` ist ab jetzt `private`. Zur Schnittstelle ergänzen Sie die Operatoren `==`, `!=`, `<`, `>`, `<=` und `>=`. Diese Änderungen ziehen weitere Änderungen nach sich – wie das bei Änderungen der Schnittstelle so ist: Die Tests müssen ebenso wie die `sort`-Funktion angepasst werden. Nehmen Sie diese Anpassungen vor.

Hinweis: Sie müssen sechs Operatoren implementieren. Wem das unnötig erscheint und wer etwas Recherche nicht scheut, nutzt den (in der Vorlesung nicht besprochenen) Spaceship-Operator.

Aufgabe 2:

a. Ändern Sie die Klasse `Position` so, dass der folgende Code

```
hfu::Position position("Mordor", 47, 11);  
std::cout<< position <<std::endl;
```

den folgenden Text ausgibt:

```
Name: Mordor, x = 47, y = 11
```

b. Ändern Sie die Klasse `City` – selbstverständlich redundanzarm – so, dass der folgende Code

```
std::string pois[]={"Sauron", "Minas Morgul", "Orodruin"};  
hfu::City city("Mordor", 47, 11,pois,3);  
std::cout<< city <<std::endl;
```

den folgenden Text ausgibt:

```
Name: Mordor, x = 47, y = 11  
Sauron  
Minas Morgul  
Orodruin
```