

Praktikum ‚Objektorientierte Programmierung‘

Aufgabenblatt 6

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.

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 Klasse `City` so, dass der folgende Test übersetzt und ausgeführt wird:

```
hfu::City city("Mordor", 47, 11, nullptr, 0);
assert(city.getNumberOfPOIs() == 0);

try {
    hfu::City city("Mordor", 47, 11, nullptr, 23);
    assert(false);
} catch (...) {}

hfu::City empty("Mordor", 47, 11);
assert(empty.getNumberOfPOIs() == 0);
```

Aufgabe 2:

Ergänzen Sie die Klasse `City` um eine Methode `add`, so dass der folgende Test übersetzt und ausgeführt wird:

```
std::string pois[] = {"Sauron", "Minas Morgul"};
hfu::City city("Mordor", 47, 11, pois, 2);
city.add("Orodruin");
assert(city.getNumberOfPOIs() == 3);
assert(city.getPOI(2) == "Orodruin");
hfu::City empty("Mordor", 47, 11);
empty.add("Orodruin");
assert(empty.getNumberOfPOIs() == 1);
assert(empty.getPOI(0) == "Orodruin");
```

Aufgabe 3:

Ergänzen Sie die Klasse `City` um eine Methode `remove`, so dass der folgende Test übersetzt

und ausgeführt wird:

```
std::string pois[]={"Sauron", "Minas Morgul", "Orodruin"};  
hfu::City city("Mordor", 47, 11, pois, 3);  
hfu::City copy1=city;  
assert(true==copy1.remove("Orodruin"));  
assert(copy1.getNumberOfPOIs()==2);  
assert(false== copy1.remove("Shire"));  
hfu::City copy2=city;  
assert(true==copy2.remove("Sauron"));  
assert(copy2.getNumberOfPOIs()==2);  
assert(copy2.getPOI(0)=="Minas Morgul");  
  
std::string saurons[]={"Sauron", "Sauron", "Sauron"};  
hfu::City same("Mordor", 47, 11, saurons, 3);  
assert(true==same.remove("Sauron"));  
assert(same.getNumberOfPOIs()==0);
```

Achten Sie darauf, dass die Klasse City für die Verwaltung der POIs nicht mehr Platz als nötig verwendet.