

Praktikum ‚Objektorientierte Programmierung‘

Aufgabenblatt 8

Wie in der Vorlesung besprochen, empfehle ich die Compiler-Flags `-std=c++14` und `-Wall`. 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. 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 `C`. Die Methoden `POI` sind ab jetzt `POI`. Die Schnittstelle soll jetzt den Zugriff über den Indexoperator ermöglichen. Recherchieren Sie dazu *sorgfältig*. Der folgende Code muss fehlerfrei ausgeführt werden. Die letzten beiden Zeilen bedürfen einiger Überlegung.

```
    = C ();  
( 0 == "S" );  
( 1 == "M" M );  
( 2 == "O" );  
  
    3 ;  
    ( );  
(...)  
  
    -1 ;  
    ( );  
(...)  
  
0 = "G";  
( 0 == "G" );  
  
    3 = "G";  
    ( );  
(...)  
  
    -1 = "G";  
    ( );
```

```
(...)  
  
C = C ();  
( C 0 == "S" );
```

Die Funktion C ist dabei wie folgt definiert:

```
:: C  
:: = "M";  
:: = "S", "M", "O";  
:: C ( , 47, 11, , 3);  
;
```

Aufgabe 2:

Ändern Sie die Klasse C so, dass der folgende Code fehlerfrei ausgeführt wird:

```
:: = "S", "M", "O";  
:: C ("M", 47, 11, , 3);  
:: = "A", "D";  
:: C ("A", 23, 42, , 2);  
=  
( . N () == "A" );  
( . () == 23 );  
( . () == 42 );  
( . N O POI () == 2 );  
( 0 == 0 );  
( 1 == 1 );
```