

Praktikum ‚Objektorientierte Programmierung‘

Aufgabenblatt 2

Auch hier gilt: Wie in der Vorlesung besprochen empfehle ich die Compiler-Flags `-Wall -Wextra -Werror`. 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 Namespace genutzt werden. Definieren Sie einen eigenen Namespace. Die Anweisung `using namespace` darf nicht genutzt werden.

Aufgabe 1:

Entwickeln Sie in C++ eine Funktion

```
boolean is_digit(char c)
```

die genau dann `true` liefert, wenn das Zeichen `c` eine Ziffer ist.

Aufgabe 2:

Entwickeln Sie in C++ eine Funktion

```
boolean is_sign(char c, xxx sign)
```

die genau dann `true` liefert, wenn `c` ein Vorzeichen, also `'+'` oder `'-'` ist. Je nach Vorzeichen erhält `sign` den Wert `1` oder `-1`. Falls `c` kein Vorzeichen ist, bleibt `sign` unverändert. Den Platzhalter `xxx` ersetzen Sie geeignet. Ein Test sieht wie folgt aus:

```
int sign=23;  
assert(is_sign('-',sign));  
assert(sign==-1);
```

Ergänzen Sie weitere Tests.

Aufgabe 3:

Entwickeln Sie eine Funktion

```
int parse_int(const std::string& text)
```

die mit Hilfe der Funktionen `is_digit` und `is_sign` einen Text in eine ganze Zahl umwandelt. Der Text muss mit einer Ziffer oder einem der beiden Vorzeichen `'+'` oder `'-'` anfangen.

Außer dem Vorzeichen darf der Text nur Ziffern enthalten.

Stellen Sie sicher, dass eine Exception geworfen wird, wenn die Funktion ihren Vertrag nicht einhalten kann. Testen Sie ausgiebig - wie in der Vorlesung gezeigt. Möglicherweise enthält die Standardbibliothek von C++ bereits eine ähnliche Funktion: es ist natürlich nicht erlaubt, diese zu verwenden.

Aufgabe 4:

Entwickeln Sie eine Variante der Funktion `parse_int`, in der anstatt der Referenz ein Zeiger als Parameter verwendet wird.