

1 Praktikum: Erstellen Sie ein C++ Programm zur Speicherung von Tweets

Sie erstellen ein kleines Twitter-Programm. In diesem Programm lesen Sie kleine Tweets von der Tastatur ein. Diese werden mit Datum und Uhrzeit in eine gekettete Liste speichert. Alle bisherigen Tweet-Nachrichten sollen nach der Eingabe angezeigt werden.

In der Aufgabenstellung erhalten Sie Lösungshinweise und Testbeispiele. Vervollständigen Sie die gegebenen Programmteile und fügen weitere Tests hinzu. Gehen Sie unbedingt in der Reihenfolge der Aufgabenstellung vor.

Die Aufgabenstellung gliedert sich in drei große Arbeitsschritte. Zu jedem Arbeitsschritt legen Sie bitte ein Projekt an, damit Sie die einzelnen Entwicklungsschritte im Praktikum erklären und zeigen können.

1.1 Aufgabenstellung

- Der erste Teil der Aufgabe besteht darin, dass Sie eine Klasse bereit stellen, die die Datenstruktur einer dynamisch, doppelt verketteten Liste zur Verfügung stellt. Um die Aufgabe zunächst zu vereinfachen gehen wir von der Speicherung von integer-Werten (als key) aus. Zur Speicherung der Daten benötigen wir eine Klasse Node. Die Klasse List dient zur Verwaltung der Datenstruktur und die Klasse Node zu Speicherung der Inhalte.

Wir benötigen folgende Dateien:

Node.h	Headerdatei der Klasse Node
Node.cpp	c++ Quelle der Klasse Node
List.h	Headerdatei der Klasse List
List.cpp	c++ Quelle der Klasse List
main.cpp	Testprogramm für die Klasse List

Diese Dateien liegen als Download auf dem ILIAS-Server (Praktikum 1 - C++ Dateien: Aufgabe 1a - zip-archivierte Dateien).

Kopieren Sie die Dateien in ein neues C++ Projekt. Verändern Sie **nicht** die Headerdateien und auch nicht die Datei "Node.cpp". Fügen Sie den fehlenden Code in die Quelldateien ein (List.cpp und main.cpp).

In einem Objekt der Klasse List gibt es mindestens ein Objekt der Klasse Node. Das erste Objekt der Klasse Node wird im Konstruktor der Klasse List erzeugt (Anker: head_tail). Dieses Objekt zeigt auf den **Anfang und gleichzeitig auf das Ende** der Liste. Auf den Wert key in diesem Objekt wird nicht zugegriffen.

In der C++ Quelle der Klasse List sind einige Methoden nur allgemein sprachlich in der zu erfüllenden Funktion beschrieben. Erzeugen Sie den notwendigen C++ Quellcode.

Die main.cpp enthält ein Testprogramm mit dem die Methoden der Klasse geprüft werden können. Dort müssen Sie weitere Testbeispiele einfügen, um alle möglichen Fälle testen zu können.

Bevor Sie weitermachen: Testen Sie mit Hilfe des Debuggers, ob der Dekonstruktor korrekt arbeitet. Dies führen Sie im Praktika Ihrem Betreuer vor.

2. Der zweite Teil der Aufgabe besteht darin, dass Sie die Klasse List und die Klasse Node je zu einer Template Klasse umbauen.

Gehen Sie wie folgt vor:

Kopieren Sie alle Dateien in ein neues Projekt. Anschließend wird der Programmteil von Node.cpp in die Datei Node.h kopiert (unterhalb der Klassendefinition). Anschließend kopieren Sie den Programmteil von List.cpp in List.h und löschen dann die Dateien Node.cpp und List.cpp. Jetzt bauen Sie beide Klassen zu Template Klassen um.

Hinweis zu friend ... im ursprünglichen List.h-Teil der Klasse List: Hier muss zusätzlich noch ein:

```
template <typename T>
friend std::ostream & operator << (std::ostream & stream, const List<T> & Liste);
```

eingesetzt werden.

Wählen Sie für die Template-Klasse List den Datentyp <int> im main-Programm.

3. Der letzte Teil der Aufgabe besteht nun darin, die ursprüngliche Aufgabe zu realisieren. Kopieren Sie dazu wieder alle Dateien vom 2. Projekt in ein neues Projekt außer die Datei main.cpp. Dafür fügen Sie die Datei tweet.h und main.cpp, die Sie im Downloadbereich des ILIAS-Server (Praktikum 1 - C++ Dateien: Aufgabe 1c - zip-archivierte Dateien) finden, ein.