

Aufgabenblatt 7

Testat im Praktikum bis 17.5.2019.

Aufgabe 7.1 (Fifo – First In First Out)

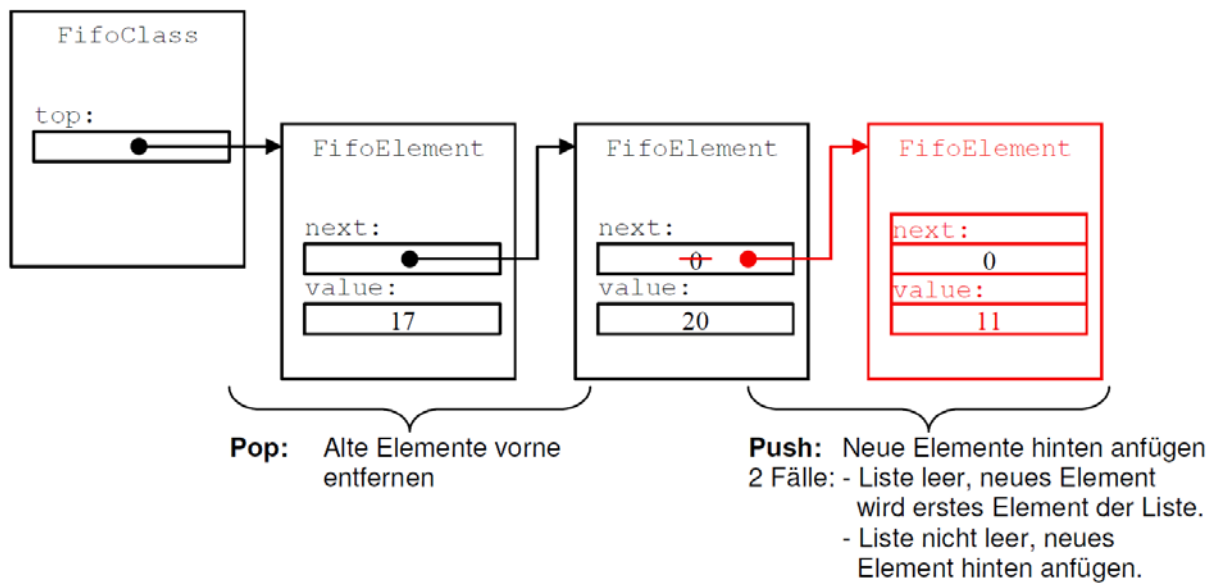
Erstellen Sie Klassenschablone **Fifo**, die einen Fifo-Speicher für Objekte vom Typ **T** mit den Standardmethoden und mindestens folgenden öffentlichen Methoden realisiert:

Fifo()	erzeugt ein leeres Fifo
~Fifo()	Destruktor
Fifo& operator<< (const T&)	schreibt eine Kopie des T-Objektes in das Fifo ein. Dieselbe Funktionalität soll durch eine push -Methode bereitgestellt werden
Fifo& operator>> (T&)	liest ein T-Objekt aus dem Fifo aus. Dieselbe Funktionalität soll durch eine pop -Methode bereitgestellt werden
operator int () const	gibt den Füllstand des Fifos zurück

Ausnahmen sollen jeweils durch Werfen einer Zeichenkettenkonstante abgefangen werden. Neben unbekannten Ausnahmen sind insbesondere Ausnahmen für Fehler bei der dynamischen Speicherbeschaffung ^{>a} Fehler beim Öffnen von nichtexistierenden Dateien, sowie der Fifo-Unterlauf zu fangen.

Die einzelnen Elemente des Fifos sollen Objekte einer von Ihnen zu schreibenden Klasse **FifoElement** sein, die als verzeigerte Liste verkettet den Fifo im Speicher dynamisch abbilden. Nachstehendes Diagramm zeigt schematisch den Aufbau, sowie die grundlegenden Fifo-Operationen anhand eines Fifos für integer-Zahlen

^{>a}**new** wirft bei Fehlern Ausnahmeobjekte vom Typ **bad_alloc**, wenn der Header **new** eingebunden ist. In so einem Fall soll die Ausnahme lokal gefangen und stattdessen eine passende Fehlermeldung weitergeworfen werden



Quelle: Prof. Dr. B. Lang, HS Osnabrück

Die Fifo-Klasse muss mit dem Testprogramm `FifoClassTest.cpp` durchgetestet werden und die Ausgabe `Ausgabe.FifoClassTest` produzieren. Dem Testprogramm entnehmen Sie ebenfalls ggf. zu erstellende öffentliche Methoden der Fifo-Klasse.

Zusatzaufgabe: Überladen Sie den `<<`-Operator zusätzlich so, dass er als Argument auch Fifo-Objekte (als Referenz) akzeptiert und dessen Elemente in `*this` einfügt.

Hinweise: Für die Bearbeitung ist die Verwendung der STL nicht erlaubt. Vielmehr soll der Fifo mit einer von Ihnen geschriebenen dynamischen Speicherverwaltung versehen werden. Achten Sie darauf, dass Kopierkonstruktor und Zuweisungsoperator tiefe Kopien erzeugen und keine Speicherlecks entstehen.