

Übung 11

Templates

A 11.1: Funktions-Templates

(7)

- (a) Schreiben Sie ein *Funktions-Template* `invert`, das zu einem übergebenen numerischen Wert den negativen Wert zurückgibt. Testen Sie das Funktions-Template mit den Typen `int`, `double` und `short`.
- (b) Überladen Sie das *Funktions-Template* so, daß `invert` auch für `char[5]` aufgerufen werden kann. Die Funktion soll dann den *invertierten String* (d.h. aus "abcde" wird "edcba") zurückliefern.

A 11.2: Klassen-Templates

(13)

In dieser Aufgabe soll eine einfache *Queue* (Warteschlange) implementiert werden. Der Begriff "Warteschlange" ist dabei ganz anschaulich wie die Schlange an der Supermarkt-Kasse zu verstehen: Elemente werden am Ende der Warteschlange angefügt (d.h. neu Kunden stellen sich an) und werden am Anfang der Warteschlange entnommen (d.h. ein Kunde hat seine Waren bezahlt und verläßt die Schlange).

- (a) Schreiben Sie ein *Klassen-Template* `Queue`, das **beliebige** Elemente an eine Warteschlange anhängen bzw. vom Anfang der Warteschlange entnehmen kann. Prüfen Sie zusätzlich ab, ob die Warteschlange beim Entnehmen eines Elements leer ist und geben Sie eine entsprechende Fehlermeldung aus.
- (b) Testen Sie die `Queue` mit Elementen aus drei **beliebigen** Klassen (z.B. die Klassen `Student` oder `Vektor`). Fügen Sie insgesamt mind. 15 Elemente in die `Queue` ein, wobei zwischendurch Elemente auch wieder entnommen werden sollen, um Überläufe zu verhindern.

Die eigentliche `Queue` soll durch ein Array der Größe 10 implementiert werden. Überprüfen Sie dieses Array auf Überläufe (d.h. mehr als 10 Elemente in der `Queue`) .