

Beispiel 1: Klasse „Queue“ (24 Punkte)

Gegeben ist die Spezifikation der Klasse TQueue zur Darstellung von FIFO (First In First Out) Listen von Elementen der Klasse TQueueElem. Zusätzlich können die Elemente der Queue nach einem vorgegebenen Schema sortiert werden (implementieren Sie einen einfachen Sortieralgorithmus, z.B. Bubble Sort).

Die Klasse TQueue soll die Elemente intern in einer ringförmig doppelt verketteten Liste mit Dummy-Element speichern. Das Dummy-Element dient als Anker für die Liste und ist notwendig, um die leere Liste als Klasse repräsentieren zu können. Das Feld *data* darf nur im Dummy-Element den Wert NULL haben. Im Destruktor der Klasse TQueue muß der Speicher aller in der Queue verbliebenen Elemente freigegeben werden. Ihr Programm soll kein „Memory Leak“ beinhalten!

Das Wissen um die Sortierreihenfolge innerhalb der Listenelemente soll im Listenelement selbst verborgen bleiben. Aus diesem Grund ist im Beispiel der Wert eines Elements nach der Initialisierung absichtlich nur mehr zu Ausgabezwecken verfügbar.

```
enum TQueueResult {Success, qOutOfMem, qEmpty, qNULL};

typedef int BOOL;

class TQueueElem {
protected:
    int value;
    ...

public:

    TQueueElem();
    TQueueElem(int initValue);
    /* Initialisiere das Element mit initValue */

    ~TQueueElem();

    BOOL IsBiggerThan(TQueueElem *otherElem) const;
    /* Gebe TRUE zurück, wenn das Empfängerobjekt
       größer ist als otherElem */

    int operator>(TQueueElem *otherElem);
    int operator<(TQueueElem *otherElem);
    int operator==(TQueueElem *otherElem);

    BOOL IsEqual(TQueueElem *otherElem) const;
    /* Gebe TRUE zurück, wenn das Empfängerobjekt
       gleich otherElem ist */
};
```

```

        void CopyToStr(char *str, int strlen) const;
        /* Kopiere eine ASCII Repräsentation der Nutzdaten
           (value) in den übergebenen String str */
    }

class TQueue {
protected:
    TQueue *next;    /* Zeiger auf nächstes Element */
    TQueue *prev;   /* Zeiger auf vorheriges Element */
    TQueueElem *data; /* Zeiger auf eigentl. Element */
    ...

public:
    TQueue(void);
    ~TQueue(void);

    TQueueResult Enqueue(TQueueElem *elem);
    /* Füge elem in die Queue ein */

    TQueueResult Dequeue(TQueueElem* &elem);
    /* Hole das älteste Element (wenn Sort aufgerufen
       wurde: das nächste in Sortierreihenfolge) aus
       der Queue heraus und gebe es in elem zurück */

    void Sort(void);
    /* Sortiere die Liste */

    int NofElems(void);
    /* Gebe die Zahl der Elemente in der Liste
       zurück */
}

```

Implementieren Sie die geforderten Klassen in einem vom Hauptprogramm getrennten Modul.

Achten Sie bei der Implementierung besonders darauf, daß ungültige Parameterwerte nicht zu ungewünschten Effekten (Absturz, Speicherkorruption) führen können. Testen Sie Ihr Modul entsprechend, um sicherzustellen, daß Sie dieses Ziel auch erreicht haben. Sie können Ihren Testtreiber hart codieren oder mit Hilfe von E/A Streams gestalten.

Achten Sie bitte darauf, bei Ihrer schriftlichen Abgabe im Header Ihre Account-Nummer anzugeben, damit wir Ihre elektronische Abgabe finden können.

Abgabe: bis 26.11. 23:59 Uhr elektronisch (Verzeichnis uebung3 einrichten!) und am 27.11. am Institut auf Papier (Listing der Quelldateien).