

Möglichkeit 1: List-Template mit Iterator (24 Punkte)

Die Übung 9 soll um einen Iterator erweitert werden (neue Elemente sind kursiv):

Implementiert werden soll eine einfach verkettete Liste, die beliebige Daten verwalten kann.

Ein einzelnes Element der Liste wird durch Objekte der Klasse **Knoten** dargestellt. Diese ist intern zu **Liste** und vereinbart Daten- und Verweisteil des Listenknotens.

```

template <class Element>
class Liste {
    friend class Knoten;
    friend class Iterator;

public:
    Liste();
    Liste(const Liste<Element>& l);
    ~Liste();
    Liste<Element>& Einfuegen(const Element& elem);
    // Element am Listenende einfuegen
    Liste<Element>& Entfernen(Element& elem);
    // Element am Listenanfang entfernen
    Liste<Element>& AlleEntfernen();
    Liste<Element>& operator=(const Liste<Element>& l);
    int AnzElems() const;
    int IsEmpty() const;

private:
    Knoten* head;
    Knoten* last;
    int anzElems;
};

class Knoten {
    Element elem;
    Knoten* next;
};

```

```

class Iterator {
public:
    Iterator(const Liste<Element>& ol);
    ~Iterator();
    Iterator& reset();      // Iterator wird an den Listenanfang gesetzt
    Iterator& operator++();
    // in der Liste wird um Element vorwärts gegangen
    Element& item();
    /* Liefert eine Referenz auf das Listenelement, auf das der Iterator
    zeigt */
    int istAmEnde() const;
    // Abfrage, ob der Iterator am Listenende angelangt ist
private:
    Liste<Element>::Knoten* posPtr;
    const Liste<Element>* pListe;
};

```

Zur einfachen Verwaltung der Liste wird jetzt ein Iterator definiert, der auf ein Element der Liste verweist. Bei der Erzeugung eines Iterator-Objekts bezieht sich dieses Objekt auf eine bestimmte Liste.

Möglichkeit 2: Container Liste mit Iterator (24 Punkte)

Implementieren Sie die Aufgabenstellung der Möglichkeit 1 mittels einer Container-Klasse.

Sie können bei dieser Übung zwischen der Implementierung der Möglichkeit 1 und der Möglichkeit 2 wählen.

Alle Ausnahmesituationen, die auftreten könnten sind mittels exception handling (try, throw, catch) abzufangen!

Abgabe: bis 2.2.2000 elektronisch (Verzeichnis uebung10 einrichten!) wie üblich und auf Papier (Listing der Quelldateien) am Institut.