C++ Kurs

Teil 7

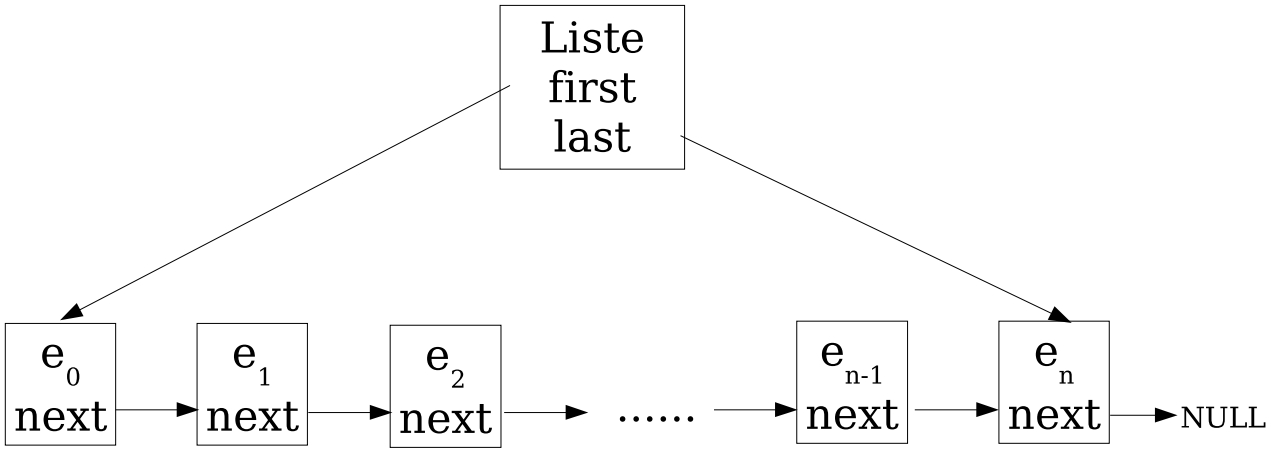
Stefan Westerfeld <stefan@space.twc.de>

Verkettete Listen

- Ein std::vector<T> hat folgende Eigenschaften
  - das Einfügen/Löschen eines Elementes am Ende ist schnell
  - das Einfügen/Löschen eines Elementes in der Mitte ist langsam, weil alle nachfolgenden Elemente einmal kopiert werden müssen
- Will man schnell Elemente in der Mitte einfügen/löschen können, empfiehlt sich stattdessen eine std::list<T>
- Im folgenden soll eine eigene Liste implementiert werden, um die Vor- und Nachteile zu verstehen

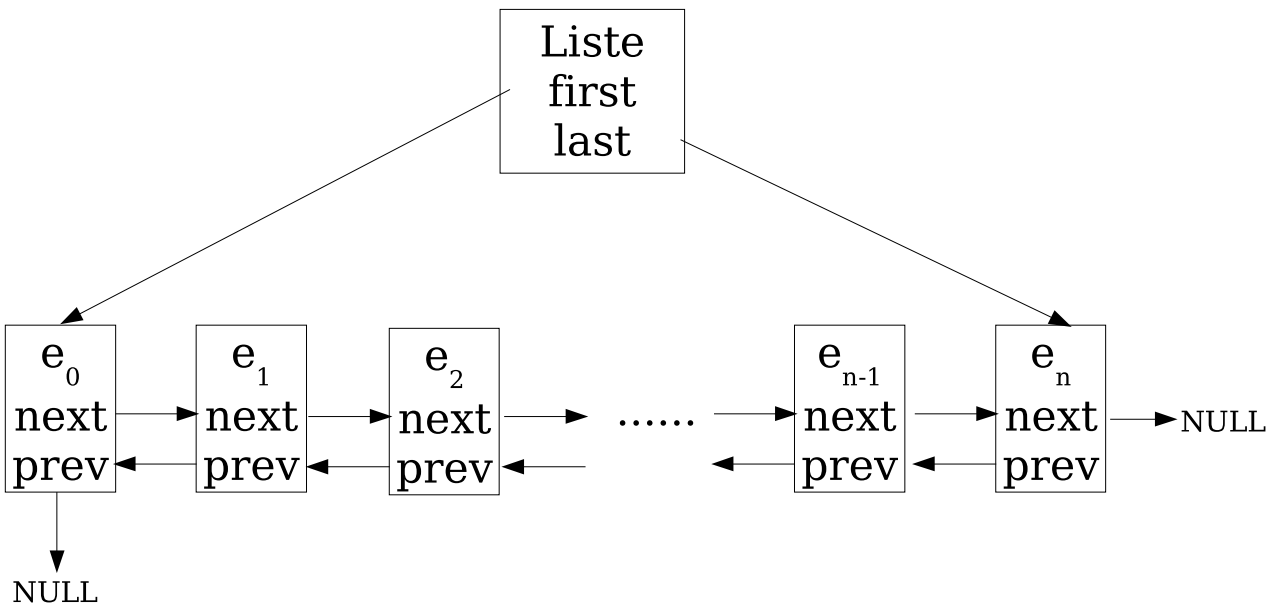
Eine Liste im Speicher

- Eine einfach verkettete Liste besteht aus Elementen, die jeweils das nächste Element kennen



Eine Liste im Speicher

- Eine doppelt verkettete Liste besteht aus Elementen, die jeweils das nächste/vorige Element kennen



Dynamisches allokieren der Elemente

- Wenn ein neues Listenelement hinzukommt, müssen wir neuen Speicher für das Element anfordern – dies geht mit new

```
{
    [...]
    Element *e = new Element();
    [...]
}
```

- Mit dem Operator new allokierte Objekte werden nie automatisch freigegeben, sondern müssen explizit mit delete gelöscht werden

```
{
   [...]
   Element *e = new Element();
   [...]
}  // e wird nicht freigegeben
```

```
{
   [...]
   Element e;
   [...]
}  // e wird freigegeben
```

Freigeben der Elemente

- Mit new allokierte Elmente müssen, wenn sie nicht mehr benötigt werden, mit delete wieder freigegeben werden, z.B.

```
// Element in Liste einfügen
{
    [...]
    Element *e = new Element();
    [...]
}

// Element aus Liste entfernen
{
    Element *e;
    // e auf das richtige Element setzen, notwendige
    // Schritte zum entfernen ausführen
    [...]
    delete e; // gibt Speicher frei
}
```

Aufgabe

Entwickle und teste eine Klasse für eine doppelt verkettete Liste in der Strings (Typ: `std::string`) gespeichert werden können.

Falls Zeit bleibt: Implementiere sortieren mit dem rekursiven Algorithmus MergeSort. Um eine Liste zu sortieren:

1. teile die Liste in zwei Teillisten mit (annähernd) gleicher Länge

2. sortiere die Teillisten mit Mergesort

3. füge die Teillisten zusammen, wobei die Sortierung erhalten bleiben soll