Dr. Matthias Grabert
Claudia Fischer

C++ mit Data Mining Anwendungen (SS 2004)

Abgabetermin: 24. Juni 2004

Das Zeigerkonzept in C++ (und C) hat einige Schwächen, dass vor allen unerfahrenen Programmierern Probleme bereiten kann:
Es ist z.B. möglich Zeiger zu inkrementieren und dekrementieren, auch wenn sie nicht auf ein Array zeigen. Ein anderes Problem ist die Tatsache, dass Pointer und die Objekte, auf die sie zeigen, voneinander unabhängig existieren. Dies kann dazu führen, dass das Objekt nicht mehr existiert, der Pointer aber noch vorhanden ist und einen von null verschiedenen, scheinbar noch gültigen Wert hat.
Im folgenden sollen Sie deshalb zwei Klassen für (relativ) sichere Pointer schreiben.

Eine Basisklasse für Pointer (3 Punkte)

Implementieren Sie dazu zuerst eine Basisklasse für Pointer, von der später beide Unterklassen abgeleitet werden sollen. Diese Klasse kann abstrakt sein, d.h. es muss nicht möglich sein, Objekte dieser Klassen zu instanziieren.
Wir gehen vereinfachend davon aus, dass die Pointer nur auf einfache Datentypen zeigen, also nicht auf Objekte oder Arrays. Natürlich ist es in diesem Fall sinnvoll diese Klasse (und auch die weiteren) als Template zu implementieren.
In der Klasse sollten folgende Methoden vorhanden sein:

- ein Default- Konstruktor,
- ein Konstruktor, der eine Variable vom Typ T übergeben bekommt,
- ein Dereferenzierungsoperator **operator***,
- eine Methode, die überprüft, ob der Zeiger ein NULL- Pointer ist, dies soll dann der Fall sein, wenn er auf keine Variable zeigt und
- zwei Operatoren **operator==** und **operator!=**, die überprüfen, ob zwei Pointer auf dieselbe Variable zeigen.

Eine abgeleitete Klasse (3 Punkte)

Erweitern Sie nun diese Basisklasse, so dass jeder Zeiger seinen eigenen Speicherplatz zugewiesen bekommt. Es können also keine zwei Pointer auf denselben Speicherplatz zeigen. Realisieren Sie dies, indem die Variable, auf die der Zeiger zeigt, bei einer Zuweisung kopiert wird. In der abgeleiteten Klasse sollen folgende Methoden vorhanden sein:

- ein Default- Konstruktor,
- ein Konstruktor, der eine Variable vom Typ T übergeben bekommt,
- ein Kopierkonstruktor,
- ein Destruktor,
- ein Zuweisungsoperator **operator=**, dem ein Objekt dieser Klasse übergeben wird und
- ein weiterer Zuweisungsoperator **operator=**, dem ein Zeiger auf eine Variable vom Typ T übergeben wird.

Eine andere abgeleitete Klasse (4 Punkte)

Eine andere Möglichkeit, das zweite der oben genannten Probleme zu beheben, ist es, die Anzahl der Zeiger, die auf dieselbe Variable zeigen, zu zählen und den Speicherplatz dann freizugeben, wenn kein Zeiger mehr darauf zeigt. Leiten Sie also eine weitere Klasse von der Basisklasse ab, die dies realisiert. In dieser Klasse sollte es wiederum folgende Methoden geben:

- ein Default- Konstruktor,
- ein Konstruktor, der eine Variable vom Typ T übergeben bekommt,
- ein Kopierkonstruktor,
- ein Destruktor und
- ein Zuweisungsoperator **operator=**, dem ein Objekt dieser Klasse übergeben wird

Schreiben Sie zusätzlich zwei kleine Programme, um diese Klassen zu testen.

Viel Erfolg!
Achtung: Jeder, der die Klausur mitschreiben will, muss sich auf der Vorlesungsseite dazu anmelden !!