Dirk Praetorius
Michele Ruggeri

Wintersemester 2014/15
07.01.2015

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 13

Aufgabe 13.1. Schreiben Sie die Klassendefinition zu einer Klasse `Polynomial` zur Speicherung von Polynomen vom Grad $n \in \mathbb{N}$, die bezüglich der Monombasis dargestellt sind, d.h.

$$p(x) = \sum_{j=0}^{n} a_j x^j.$$

In der Klasse soll neben dem dynamischen Vektor $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ der Koeffizienten `(double*)` auch der Grad $n \in \mathbb{N}$ gespeichert werden. Außerdem soll die Klasse über folgende Funktionalität verfügen:

- Destruktor, Konstruktor zum Allokieren des Null-Polynoms mit Grad n, Copy-Konstruktor,
- Zuweisungsoperator,
- Zugriff auf die Koeffizienten des Polynoms mittels `[ ]`, d.h. für $0 \le j \le n$ `p[j]` liefert a_j,
- die Möglichkeit, zwei Polynome p und q mit der Syntax `r=p+q` addieren zu können,
- die Möglichkeit, zwei Polynome p und q mit der Syntax `r=p*q` multiplizieren zu können,
- die Möglichkeit, ein Polynom p mittels `p(x)` bei $x \in \mathbb{R}$ auszuwerten bzw. mittels `p(k,x)` die k-te Ableitung von p bei $x \in \mathbb{R}$ auszuwerten.

Implementieren Sie die Konstruktoren und den Destruktor der Klasse, sowie den Zuweisungsoperator und den Koeffizientenzugriff mittels `[ ]`.

Aufgabe 13.2. Für $k \ge 0$ ist die k-te Ableitung $p^{(k)}$ eines Polynoms p wieder ein Polynom. Implementieren Sie die Möglichkeit, die k-te Ableitung von einem Polynom p mittels `p(k,x)` auszuwerten, wobei $x \in \mathbb{R}$ und $k \ge 0$. Für $k = 0$ sei der Aufruf `p(x)` erlaubt.

Aufgabe 13.3. Die Summe zweier Polynome ist wieder ein Polynom. Implementieren Sie für die Klasse `Polynomial` aus Aufgabe 13.1 die nötige Funktionalität, um zwei Polynome p und q mittels `r=p+q` zu addieren. Eine Zahl vom Typ `double` ist auch ein Polynom. Implementieren Sie die Möglichkeit eine Zahl a zu einem Polynom p mittels `r=a+p` zu addieren.

Aufgabe 13.4. Das Produkt zweier Polynome ist wieder ein Polynom. Implementieren Sie für die Klasse `Polynomial` aus Aufgabe 13.1 die nötige Funktionalität, um zwei Polynome p und q mittels `r=p*q` zu multiplizieren. Eine Zahl vom Typ `double` ist auch ein Polynom. Implementieren Sie die Möglichkeit eine Zahl a zu einem Polynom p mittels `r=a*p` zu multiplizieren.

Aufgabe 13.5. Eine untere Dreiecksmatrix $L \in \mathbb{R}^{n \times n}$ mit

$$L = \begin{pmatrix} \ell_{11} & & & & \mathbf{0} \\ \ell_{21} & \ell_{22} & & & \\ \ell_{31} & \ell_{32} & \ell_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \dots & \ell_{nn} \end{pmatrix}$$

hat höchstens $\frac{n(n+1)}{2} = \sum_{j=1}^{n} j$ nicht-triviale Einträge. Schreiben Sie eine Klasse `matrixL`, in der neben der Dimension $n \in \mathbb{N}$ die Koeffizienten L_{ij} in einem dynamischen Vektor der Länge $\frac{n(n+1)}{2}$ gespeichert werden. Speichern Sie L zeilenweise. Die Klasse soll die folgenden Funktionalitäten enthalten:

- Konstruktor, Copy-Konstruktor, Destruktor,
- Zuweisungssoperator,
- Zugriff auf die Koeffizienten mittels `L(i,j)`,
- die Möglichkeit, zwei Matrizen A und B (passender Dimension) mittels $A+B$ bzw. $A*B$ addieren bzw. multiplizieren zu können.

Implementieren Sie die Konstruktoren und den Destruktor der Klasse, sowie den Zuweisungsoperator und den Koeffizientenzugriff.

Aufgabe 13.6. Überladen Sie den Operator `+` für die Klasse `MatrixL` aus Aufgabe 13.8. Schreiben Sie auch ein Programm, in welchem Sie die Implementierung testen.

Aufgabe 13.7. Beweisen Sie mit der Formel des Matrix-Matrix-Produktes, dass das Produkt zweier unterer Dreiecksmatrizen eine untere Dreiecksmatrix ist. Überladen Sie den Operator `*` für die Klasse `MatrixL` aus Aufgabe 13.8. Schreiben Sie auch ein Programm, in welchem Sie die Implementierung testen.

Aufgabe 13.8. Gegeben sei eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ mit $\ell_{jj} \neq 0$ für alle $j = 1, \dots, n$. Zu gegebenem $b \in \mathbb{R}^n$ existiert dann ein eindeutiges $x \in \mathbb{R}^n$ mit $Lx = b$. Schreiben Sie eine Methode `solveL` für die Klasse aus Aufgabe , die x berechnet. Implementieren Sie die Möglichkeit, für eine untere Dreiecksmatrix $L \in \mathbb{R}^{n\times n}$ und einen Vektor $b \in \mathbb{R}^n$ das System $Lx = b$ mittels `x=L|b` zu lösen, d.h. die Syntaxen `x=solveL(L,b)` und `x=L|b` müssen äquivalent sein.