Dirk Praetorius
Michele Ruggeri

Wintersemester 2014/15
07.01.2015

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 13

Aufgabe 13.1. Write a class `Polynomial` to save polynomials of degree $n \in \mathbb{N}$, which are represented with respect to the monomial basis, i.e.,

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

The class contains the dynamical vector $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ of the coefficients `(double*)` as well as the degree $n \in \mathbb{N}$. Moreover, the class should contain the following features:

- destructor, constructor to allocate the zero-polynomial of degree n, copy-constructor,
- assignment operator,
- access to the coefficients of the polynomials via `[ ]`, i.e., for $0 \le j \le n$ `p[j]` returns the value a_j,
- the possibility to add two polynomials p and q using the expression `r=p+q`,
- the possibility to multiply two polynomials p and q using the expression `r=p*q`,
- the possibility to evaluate a polynomial p and its k-th derivative $p^{(k)}$ at $x \in \mathbb{R}$ with expressions `p(x)` and `p(k,x)`, respectively.

Implement the constructors, the destructor of the class, as well as the assignment operator and the access to the coefficients via `[ ]`.

Aufgabe 13.2. For $k \ge 0$ the k-th derivative $p^{(k)}$ of a polynomial p is still a polynomial. Implement the feature to evaluate the k-th derivative of a polynomial p via `p(k,x)`, where $x \in \mathbb{R}$ and $k \ge 0$. For $k = 0$ the call `p(x)` must be also possible.

Aufgabe 13.3. The sum of two polynomials is still a polynomial. Implement for the class `Polynomial` from Exercise 13.1 the feature of adding two polynomials p and q via `r=p+q`. A number of type `double` is also a polynomial. Implement the opportunity to add a number a to a polynomial p via `r=a+p`.

Aufgabe 13.4. The product of two polynomials is still a polynomial. Implement for the class `Polynomial` from Exercise 13.1 the feature of multiplying two polynomials p and q via `r=p*q`. A number of type `double` is also a polynomial. Implement the opportunity to multiply a number a to a polynomial p via `r=a*p`.

Aufgabe 13.5. A lower triangular matrix $L \in \mathbb{R}^{n \times n}$ with

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

has at most $\frac{n(n+1)}{2} = \sum_{j=1}^{n} j$ nontrivial coefficients. Write a class `matrixL` to save the coefficients L_{ij} in a dynamical vector with length $\frac{n(n+1)}{2}$ together with the dimension $n \in \mathbb{N}$. Save the matrix L row-wise. The class should contain the following features:

- constructor, copy-constructor, destructor,
- assignment operator,
- access to the coefficients via `L(i,j)`,
- the possibility to add and multiply two matrices A and B (of appropriate dimension) via $A + B$ and $A * B$, respectively.

Implement the constructors, the destructor, as well as the assignment operator and the access to the coefficients.

Aufgabe 13.6. Overload the operator `+` for the class `MatrixL` from Exercise 13.8. Then, write a programm to test your implementation.

Aufgabe 13.7. Use the formula for the matrix-matrix product to show that the product of two lower triangular matrices is a lower triangular matrix. Then, overload the operator `*` for the class `MatrixL` from Exercise 13.8. Moreover, write a program to test your implementation.

Aufgabe 13.8. Let $L \in \mathbb{R}^{n\times n}$ be a lower triangular matrix such that $\ell_{jj} \neq 0$ for all $1 \leq j \leq n$. Given $b \in \mathbb{R}^n$, there exists a unique $x \in \mathbb{R}^n$ such that $Lx = b$. Write a method `solveL` for the class from Exercise to compute x. Implement also the feature to solve the system $Lx = b$ (for a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$) by using the command `x=L|b`, i.e., the expressions `x=solveL(L,b)` and `x=L|b` must be equivalent.