Michael Feischl
Andrea Scaglioni

Wintersemester 2023
15.12.2023

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 10

`std::vector<>`, Referenzen, `const`

Die Pflichtübungen sind mit * gekennzeichnet. Verwenden Sie Kommentare (`/*...*/` oder `//...`), um Ihren Code zu dokumentieren, wie im Skript gezeigt. Spezifizieren Sie in einem Blockkommentar (`/* ... */`) am Ende des Codes, wie Sie Ihre Implementierung getestet haben.

Aufgabe 10.1. * Schreiben Sie eine Klasse `ComplexVector` zur Speicherung von Vektoren $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{C}^n$ für $n \in \mathbb{N}$. Implementieren Sie folgende Methoden:

- Konstruktor `ComplexVector(int n, Complex z)`
- `size()`, welche die Länge n zurückgibt
- Zugriffsmethoden `void setEntry(int i, Complex z)` und `Comlpex getEntry(int i)`

Sie dürfen die Klasse `Complex` (Skriptum, Folie 56) verwenden. Verwenden Sie `const` und Referenzen wo nötig. Implementieren Sie außerdem folgende Funktionen

- `Complex scalarProduct(ComplexVector u, ComplexVector v)`, welche das Skalarprodukt

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} a_i b_i \in \mathbb{C}$$

  berechnet
- `ComplexVector sum(ComplexVector u, ComplexVector, v)`, welche die Summe von zwei Vektoren berechnet.

Verwenden Sie (read-only) Referenzen wo es sinnvoll ist!
Das Produkt zweier komplexer Zahlen $a = a_r + ia_i, b = b_r + ib_i \in \mathbb{C}$ ist durch

$$ab = a_r b_r - a_i b_i + i(a_r b_i + a_i b_r)$$

definitert.

Aufgabe 10.2. * Erweitern Sie die Klasse `Matrix` (Skriptum, Folie 53) um die Methode `Vector MultiplyVectorRight(Vector x)`, welche für eine gegebene Matrix $A \in \mathbb{R}^{m\times n}$ und einen Vektor $x \in \mathbb{R}^n$ das Matrix-Vektor-Produkt $Ax \in \mathbb{R}^m$, wo $(Ax)_i = \sum_{j=1}^{n} A_{i,j} x_j$ für alle $i = 1, \dots, m$, berechnet.
Implementieren Sie die Methode `const Vector& getColumn(int i)`, welche die i-te *Spalte* A zurückgibt.
Implementieren Sie die Methode `Vector MultiplyVectorRight(Vector x)`, welche für einen gegebenen Vektor $y \in \mathbb{R}^m$, das Matrix-Vektor-Produkt $y^T A \in \mathbb{R}^n$, wo $(y^T A)_i = \sum_{j=1}^{m} y_j A_{j,i}$ für alle $i = 1, \dots, n$, berechnet.
Verwenden Sie `const` und Referenzen wo notig. Verwenden Sie (read-only) Referenzen wo es sinnvoll ist!

Aufgabe 10.3. Erweitern Sie die Klasse `ComplexVector` aus Übung 10.1 um:

- eine Methode `Vector realPart()`, die einen `Vector`, der die Realteile in der gleichen Reihenfolge enthält, zurückgibt,
- eine Methode `Vector imaginaryPart()`, die dasselbe mit dem Imaginärteil macht.

Verwendend Sie die Klasse `Vector` aus der Vorlesung (Folie 49).

Aufgabe 10.4. Sei $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{C}^n$, $z \in \mathbb{C}$. Implementieren Sie folgende Funktionen:

- `ComplexVector MultiplyScalar(ComplexVector v, Complex z)`, welche das Produkt mit einem Skalar $(zv_1, \ldots, zv_n)$ berechnet
- `ComplexVector conjugate(ComplexVector v)`, welche $(\overline{v_1}, \ldots, \overline{v_n})$ berechnet
- `Vector modulus(ComplexVector v)`, welche $(|v_1|, \ldots, |v_n|)$ berechnet

Für $z = z_r + iz_i \in \mathbb{C}$, $\overline{z} := z_r - iz_i$ und $|z| := \sqrt{z_r^2 + z_i^2}$. Verwenden Sie read-only Referenzen wo es sinnvoll ist.

Aufgabe 10.5. Erweitern Sie die Klasse `Matrix` aus der Vorlesung (Folie 52) um die folgenden Methoden:

- `Matrix transpose()`, die die transponierte Matrix berechnet. Die Transponierte der Matrix $A \in \mathbb{R}^{m\times n}$ ist durch $A^T \in \mathbb{R}^{n\times m}$ definiert, wo $(A^T)_{j,k} = (A)_{k,j}$.
- `Matrix ScalarMultiply(double a)`, die das Produkt mit $\alpha \in \mathbb{R}$ berechnet. Das Produkt der Matrix $A \in \mathbb{R}^{m\times n}$, mit der Zahl $\alpha \in \mathbb{R}$ ist durch $(\alpha A)_{j,k} = \alpha(A)_{j,k}$ definiert.
- `Matrix RightMultiply(Matrix B)`, die das Produkt von Rechts mit einer anderen Matrix berechnet. Das Produkt zweier Matrizen $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$ ist durch $(AB)_{i,k} = \sum_{j=1}^{n}(A)_{i,j}(B)_{j,k}$ definiert.

Verwenden Sie (read-only) Referenzen wo es sinnvoll ist!

Aufgabe 10.6. Sei $p, q, m, n \geq 1$, $A \in \mathbb{R}^{n\times m}$ gegeben. Die $L^{p,q}$ Norm ist durch

$$\|A\|_{p,q} = \left(\sum_{j=1}^{m}\left(\sum_{i=1}^{n}|a_{i,j}|^p\right)^{q/p}\right)^{1/q}$$

definiert. Erweitern Sie die Klasse `Matrix` um eine Methode, die die $L^{p,q}$ Norm berechnet. Verwenden Sie (read-only) Referenzen wo es sinnvoll ist!

Aufgabe 10.7. Erweitern Sie die Klasse `Matrix` um die Methode `void swapRows(int i, int j)`, welche die i-te und j-te Zeile von A vertauscht. Außerdem, implementieren Sie eine Funktion `Matrix sortByRow(Matrix A)`, welche eine sortierte Version von A herstellt, ohne A zu ändern. Die neue Matrix soll nach dem ersten Eintrag jeder Zeile sortiert sein. Zum Beispiel,

$$A = \begin{pmatrix} 2.1 & 7. \\ 1. & -3. \\ -3.3 & 9. \end{pmatrix} \qquad \texttt{sortByRow}(A) = \begin{pmatrix} -3.3 & 9. \\ 1. & -3. \\ 2.1 & 7. \end{pmatrix}$$

Verwenden Sie read-only Referenzen wo es sinnvoll ist.

Aufgabe 10.8. Laut der Vorlesung ist der Zugriff auf `private` Members einer Klasse nur über `set`- und `get`-Methoden der Klasse möglich. Wie lautet die Ausgabe des folgenden C++ Programms? Warum ist das möglich? Erkären Sie, warum das schlechter Programmierstil ist.

```
#include <iostream>
using std::cout;
using std::endl;

class Test{

private:
  int N;

public:
  void setN(int N_in) { N = N_in; };
```

```
  int getN(){ return N; };
  int* getptrN(){ return &N; };

};

int main(){

  Test A;
  A.setN(5);
  int* ptr = A.getptrN();
  cout << A.getN() << endl;
  *ptr = 10;
  cout << ptr << endl;
  cout << A.getN() << endl;

  return 0;
}
```