Dirk Praetorius
Michele Ruggeri
Bernhard Stiftner

Sommersemester 2016
03.05.2016

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 8

Aufgabe 8.1. Was ist ein Gleitkommazahlsystem? Aus welchen Bestandteilen setzt sich eine Gleitkommazahl zusammen? Wie bestimmt man daraus ihren Wert? Was verbirgt sich hinter den Symbolen `Inf`, `-Inf` und `NaN`? Was ist eine normalisierte Gleitkommazahl? Was ist ein implizites erstes Bit? Welchen Wert haben die größte und die kleinste positive normalisierte Gleitkommazahl im `double`-Gleitkommazahlsystem $\mathbb{F}(2, 53, -1021, 1024)$?

Aufgabe 8.2. Schreiben Sie eine Struktur `Vector` zur Speicherung von `double`-Vektoren der Länge n. Die Struktur enthalte neben der Dimension n den dynamischen Datenvektor. Im Gegensatz zur üblichen Indizierung in `C` (bzw. zur Indizierung in der Vorlesung) soll die Indizierung der Vektorkoeffizienten in Ihrer Struktur `Vector` von 1 bis n laufen (wie in der Mathematik üblich). Ferner schreibe man die zugehörigen Funktionen `newVector`, `delVector`, `getVectorLength`, `setVectorLength`, `getVectorEntry`, `setVectorEntry`, wobei `setVectorLength` den Speichervektor reallokieren soll. Stellen Sie mittels `assert` sicher, dass die Dimension n in `Vector` immer positiv ist. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vector.c` in das Verzeichnis `serie08`.

Aufgabe 8.3. Schreiben Sie eine Funktion `vectorSort`, die einen Vektor $x \in \mathbb{R}^n$ aufsteigend sortiert und durch den sortierten Vektor überschreibt. Sie können jeden Sortieralgorithmus verwenden, den Sie kennen. Verwenden Sie die Struktur `Vector` von Aufgabe 8.2. Welchen Aufwand hat Ihre Funktion? Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vectorsort.c` in das Verzeichnis `serie08`.

Aufgabe 8.4. Schreiben Sie eine Funktion `vectorCut`, die für gegebenes $C > 0$ und gegebenen Vektor $x \in \mathbb{R}^n$ einen Vektor y erzeugt, bei dem alle Einträge x_j mit $|x_j| > C$ aus den Vektor $x \in \mathbb{R}^n$ gestrichen sind, und den gekürzten Vektor y zurückgibt. Als Beispiel liefere die Funktion für $x = (4, 9, -2, 7, 1, -8)$ und $C = 4.2$ den Vektor $y = (4, -2, 1)$. Verwenden Sie die Struktur `Vector` von Aufgabe 8.2. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vectorcut.c` in das Verzeichnis `serie08`.

Aufgabe 8.5. Schreiben Sie eine Funktion `vectorErathostenes`, die für eine natürliche Zahl N einen Vektor erzeugt, der alle Primzahlen $x_j \leq N$ enthält. Bestimmen Sie die Einträge des Vektors mit dem Sieb des Erathostenes (siehe `https://de.wikipedia.org/wiki/Sieb_des_Eratosthenes`). Verwenden Sie die Struktur `Vector` von Aufgabe 8.2. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vectorerathostenes.c` in das Verzeichnis `serie08`.

Aufgabe 8.6. Schreiben Sie eine Funktion `vectorIndex`, die für einen gegebenen Vektor $x \in \mathbb{R}^n$ den kleinsten Index J zurückgibt, sodass

$$\sum_{j=1}^{J} |x_j| \leq \sum_{j=J+1}^{N} |x_j| \leq \sum_{j=1}^{J+1} |x_j|.$$

Verwenden Sie die Struktur `Vector` von Aufgabe 8.2. Welchen Aufwand hat Ihre Funktion? Wenn möglich, finden Sie einen Algorithmus, der Aufwand $\mathcal{O}(n)$ hat! Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `vectorindex.c` in das Verzeichnis `serie08`.

Aufgabe 8.7. Schreiben Sie eine Klasse `University`. Diese soll neben den Feldern `numStudents`, `city` und `name` die Methoden `graduate` und `newStudent` haben. Wird `graduate` aufgerufen, so verringert sich die Anzahl der Studenten um 1, wohingegen `newStudent` die Anzahl um 1 erhöht. Alle Datenfelder sollen als `private` deklariert sein. Sie müssen sich also zusätzlich `get-` und `set`-Methoden schreiben. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `University.cpp` in das Verzeichnis `serie08`.

Aufgabe 8.8. Erstellen Sie eine Klasse `SparKonto` mit den Variablen `kontonummer`, `guthaben` und `zinssatz`. Ferner sollen noch die `get` und `set`-Methoden für die Variablen `zinssatz` und `kontonummer` implementiert werden. Um das Guthaben zu ändern schreiben Sie die Methoden `abheben` und `einzahlen`. Beachten Sie, dass Sie bei einem Sparkonto nicht ins Minus gehen können. Der Zinssatz und die Kontonummer dürfen natürlich auch nicht negativ werden. Schließlich implementiere man noch die Methode `berechneGuthaben`. Testen Sie Ihren Code entsprechend. Speichern Sie den Source-Code unter `sparkonto.cpp` in das Verzeichnis `serie08`.