Abteilung für
Angewandte Mathematik

SS 2023
Ludwig Striet
Tatjana Stiefken

Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 8 – 19.06.2023

Abgabe: *bis Sonntag, 25.06.2023 24 Uhr per Mail an Ihren Tutor.*

Aufgabe 1 (5 Punkte)**.** Ändern Sie die Klasse BankKonto aus der Vorlesung so ab, dass sie eine zusätzliche private Variable `dispo` vom Typ `double` enthält. Diese soll beim Erstellen auf `0` gesetzt werden. Fügen Sie außerdem eine Methode `setzeDispo(double d)` hinzu, die folgende Funktionalität erfüllt:

- `dispo` darf nicht positiv sein
- `dispo` kann nicht größer als der aktuelle Kontostand gesetzt werden, ist z.B. bei einem Kontostand von -100 kann `dispo` nicht auf -50 gesetzt werden.

Ändern Sie außerdem die Methoden `auszahlen` und die `ueberweisen` so, dass diese überprüfen, dass `dispo` nicht unterschritten wird.

Hinweis: Verwenden Sie die Datei bankkonto.cc von der Vorlesungswebseite. Diese unterscheidet sich etwas von dem Beispiel aus dem Skript.

Aufgabe 2 (5+5 Punkte)**.** Gegeben sei eine sortierte Liste von N Zahlen, d.h. eine Liste von Zahlen

$$e_0, \dots, e_{N-1}$$

s.d. $e_n < e_{n+1}$. Ein Verfahren, um mit logarithmischem Aufwand zu testen, ob ein Element e in der Liste enthalten ist, besteht darin, die Liste wiederholt zu halbieren und nur in der entsprechenden Teilliste zu suchen.

(a) Schreiben Sie das Verfahren als pseudo-code auf und zeigen Sie, dass der Aufwand logarithmisch ist, also $\mathcal{A}(N) = C \cdot \log_2(N)$. Dabei dürfen Sie annehmen, dass N eine Zweierpotenz ist, d.h. $N = 2^n$ für ein $n \in \mathbb{N}$. Konstanten, die nicht von N abhängen, können Sie dabei ignorieren.

(b) Implementieren Sie das Verfahren für sortierte Listen beliebiger Länge N, also so, dass N nicht notwendigerweise eine Zweierpotenz sein muss.

Aufgabe 3 (5 Punke)**.** Ein (nicht besonders effizientes) rekursives Sortierverfahren für eine Liste an Position `int* l` der Länge `unsigned int n` geht so.

- Beginne mit `i = 0`
- Finde die Position `m` des Minimalen Elements in Liste `l[i]...l[n-1]`
- Vertausche die Elemente `l[0]` und `l[m]`
- Rufe die Sortierfunktion für die Liste `l+1` der Länge `n-1` auf

Implementieren Sie eine *rekursive* Funktion

```
void rec_sort(int* l, unsigned int n)
```

die nach dem oben beschriebenen Verfahren eine Liste sortiert.

Hinweis: Schreiben Sie zunächst eine Hilfsfunktion

```
unsigned long min_ind(int* l, unsigned int n)
```

die die Position des minimalen Elements in der Liste beginnend an `l` der Länge `n` findet.

Aufgabe 4 (optional)**.** Schildern Sie in einer Datei `erfahrung.txt` kurz Ihre Erfahrung mit dem aktuellen Übungsblatt. Berichten Sie darin wieder in Stichpunkten bzw. ein bis zwei kurzen Sätzen über Ihre Erfahrungen mit Kursinhalt und Übungsaufgaben. Was fiel Ihnen leicht? Was ist noch unklar? Wie viel Zeit haben Sie für die Bearbeitung der Hausaufgaben benötigt und welche Probleme traten dabei auf?