Abteilung für
Angewandte Mathematik

SS 2023
Ludwig Striet
Tatjana Stiefken

Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 5 – 22.05.2023

Abgabe: *bis Sonntag, 04.06.2023 24 Uhr per Mail an Ihren Tutor.*

Aufgabe 1 (5 Punkte). Betrachten Sie das folgende C++-Programm

```
1  int main() {
2    int x = 17, y = 8;
3
4    int* a = &x;
5    int& b = y;
6
7    *a = b*8;
8
9    int* z1, z2;
10 }
```

Stellen Sie, wie in der Vorlesung, eine Tabelle mit den Spalten *Variablenname*, *Typ*, *Speicheradresse* und *Wert* auf. Nehmen Sie (vereinfachend) an, dass jeder Datentyp/jede Variable die Größe 1 hat und vergeben Sie fortlaufende Speicheradressen in der Reihenfolge, in der die Variablen angelegt werden. Die Werte sollen denen *unmittelbar vor dem Ende* des Programms entsprechen.

Aufgabe 2 (5 Punkte). Schreiben Sie eine Funktion,

```
1  bool contains(std::vector<std::string> list, std::string s) {
2    ...
3  }
```

die überprüft ob ein gegebener `std::string s` in einem `std::vector<std::string> list` enthalten ist.

Aufgabe 3 (5 Punkte). Benutzen Sie die Klassen `std::string` und `std::vector<std::string>`, um eine Einkaufsliste zu schreiben. Dazu sollen nacheinander Produkte von der Konsole eingelesen werden, und nach jedem Produkt abgefragt werden, ob weitere Produkte hinzugefügt werden sollen. Wenn ein eingegebenes Produkt schon in der Liste steht, soll es nicht erneut hinzugefügt werden, sondern es soll nach dem nächsten Produkt gefragt werden. Wenn keine weiteren Produkte mehr eingegeben werden sollen, sollen alle Produkte auf der Konsole ausgegeben werden.
Hinweis: Sie können einen `std::string` mittels

```
1       std::string s;
2       std::cin >> s;
```

von der Konsole einlesen. Nutzen Sie die `contains` Funktion aus Aufgabe 3 um zu überprüfen, ob ein Gegenstand bereits in der Liste gespeichert ist.

Aufgabe 4 (5 Punkte + 5 Bonuspunkte)**.** ,
(a) Implementieren Sie eine Funktion `forward_diff_quot` die als Argumente Gleitkommazahlen `x`, `h` sowie eine Funktion `f` (als Funktionenpointer) übergeben bekommt und die nach der Formel

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

angenäherte Ableitung von `f` an der Stelle `x` als Gleitkommazahl zurück gibt.
(b) Das Newton-Verfahren zum Finden einer Nullstelle x^* einer Funktion f funktioniert so:
Eingabe: Reelle Zahl x_0 und Toleranz $\delta > 0$.

(1) Setze $x_{alt} = x_0$,
(2) Wenn $|f(x_{alt})| < \delta$: Stoppe
(3) Setze $x_{neu} = x_{alt} - f(x_{alt})/f'(x_{alt})$.
(4) Setze $x_{alt} = x_{neu}$
(5) Gehe zu Schritt (1).

Ausgabe: Approximation x_{neu} von x^*.
Implementieren Sie das Verfahren und testen Sie es beispielsweise an einer quadratischen Funktion!
Hinweis: Sie dürfen das Verfahren mit einem x_0 in der Nähe der tatsächlichen Nullstellen initialisieren. Sie müssen nicht überprüfen, ob überhaupt Nullstellen existieren. Zum Berechnen der Ableitung können Sie die Approximation aus Teil *(a)* verwenden. Den Betrag $|f(\cdot)|$ können Sie ausrechnen oder `cmath` einbinden und die Funktion `std::abs` verwenden.

Aufgabe 5 (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?