

Aufgabe Nr. 4

19.12.2023

Für diese Übung benötigen Sie die Vorgabedateien `stats.h` und `stats.cc` aus Moodle. Die folgenden Aufgaben beziehen sich auf Ergänzungen in der Datei `stats.cc`. An der Datei `stats.h` sollen keine Änderungen vorgenommen werden (geben Sie diese Datei bitte dennoch ab).

Aufgabe 4.1

Füllen Sie den Vektor `v` mit 3902365 Zufallszahlen im Intervall $[1, \dots, 100]$. Zum Füllen soll die Funktion `generate_n`¹ mit dem vorgegebenen Funktor `Data` verwendet werden.

Aufgabe 4.2

Bitte bearbeiten Sie diese Aufgabe nach Vorlesung 10.

Sortieren Sie den Vektor in aufsteigender Reihenfolge und geben den Inhalt des Vektors aus, indem Sie die Werte mittels `copy` über einen `ostream_iterator` auf den Standardausgabekanal kopieren. Zwischen den Zahlenwerten soll ein Komma, gefolgt von einem Leerzeichen, stehen.

Aufgabe 4.3

Legen Sie eine `map` als lokale Variable an, in der die Häufigkeit der erzeugten Zufallszahlen gespeichert wird. Zum Ermitteln der Häufigkeit sollen Sie mit `for_each` über den Vektor iterieren und für jedes Element eine Funktion aufrufen, die den betreffenden Eintrag in der `map` aktualisiert.

Aufgabe 4.4

Geben Sie die Häufigkeitsverteilung in einer Art Balkengrafik wie folgt aus. Dazu soll für jeden Eintrag in der `map` eine Zeile auf dem Standardausgabekanal erzeugt werden. Die Zeile soll mit dem auf drei Zeichen Breite festgelegten Schlüssel beginnen, dann folgen ein Doppelpunkt und ein Leerzeichen. Danach soll der zu diesem Eintrag gespeicherte Wert auf sechs Zeichen Breite ausgerichtet sowie ein weiteres Leerzeichen ausgegeben werden. Im Anschluss wird jeweils ein Doppelkreuz („#“) für jede volle Tausend Vorkommen des jeweiligen Eintrags hintereinander ausgegeben.

Eine Ausgabe könnte dann zum Beispiel folgendermaßen aussehen (gekürzter Ausschnitt):

¹Vergleiche https://en.cppreference.com/w/cpp/algorithm/generate_n.

```

1: 39283 #####
2: 38955 #####
3: 38950 #####
4: 39134 #####
.      .      .
.      .      .
.      .      .
97: 38727 #####
98: 39011 #####
99: 38840 #####
100: 38989 #####

```

Aufgabe 4.5

Schließlich soll mit Hilfe der Funktion `accumulate` der Durchschnitt der erhobenen Häufigkeiten berechnet und auf dem Standardausgabekanal ausgegeben werden.

Aufgabe 4.6

Bewerten Sie die Ergebnisse aus den Aufgaben 4.4 und 4.5 vor dem Hintergrund der Implementierung des Funktors `Data` in `stat.h`. Was fällt auf?