

Übungsblatt Nr. 11: in der Übungsstunde

- 1) Von <https://www.citypopulation.de/de/austria/tirol/> stammen die statistischen Daten über die Tiroler Gemeinden, die in der Datei `Gemeinden.dat` in der Reihenfolge Gemeindename, Bezirksname, Einwohnerzahlen von 1981, 1991, 2001, 2011, 2021 gespeichert sind.
 - a) Vervollständigen Sie in `v1.cpp` das Programmgerippe `v0.cpp`, sodass Sie damit die Datei `Gemeinden.dat` in den Vektor `v` einlesen können. Es fehlt dazu die Streameingabe.
 - b) Vervollständigen Sie die Ausgaben.
- 2) Verwenden Sie Ranges (aka STL-2.0), wo das schon möglich ist. Ersetzen Sie alle verwendeten Algorithmen aus `algorithm` durch die Ranges-Version, d.h. wechseln Sie vom Namespace `std::` zu `std::ranges`:

```
std::count(...) -> std::ranges::count(...)
count(...)      -> ranges::count(...)
```

Die Algorithmen aus `numeric` haben derzeit noch keine neue Version
- 3) Verwenden Sie die neuen Features der STL-2.0 (Overloads ohne `begin/end` und Projektionen).
- 4) Ich habe in der Datei `accumulate-c++23.hpp` eine Vorabversion geschrieben, die mit `cpp23::accumulate()` aufrufbar ist. Verwenden Sie diese neue Version.

Übungsblatt Nr. 11 Hausübung: Die folgenden Aufgaben sind Pflicht und zählen 1 Punkt!

- 1)
 - a) Geben Sie die Einwohnerzahlen Tirols in den Jahren 1981, 1991, 2001, 2011, 2021 aus.
 - b) Geben Sie die Anzahl der Gemeinden im Bezirk Reutte aus.
 - c) In wievielen davon ist die Einwohnerzahl zwischen 2001 und 2021 gewachsen?

- 2) Verwenden Sie die Headerdatei `Bruch.hpp`. Lesen Sie daraus Brüche aus der Datei `data` in den `vector<Bruch> v`;
a) Wieviele Brüche konnten Sie einlesen? Berechnen Sie die Gesamtsumme.
b) Geben Sie den kleinsten und größten Bruch aus (verwenden Sie `to_double()` in der Vergleichsoperation).
c) Wie oft steht 282 im Nenner?
d) Sortieren Sie den Vektor aufsteigend und geben Sie den Median aus. Der Median ist der mittlere Wert bei einer ungeraden Anzahl und der Mittelwert der beiden mittleren Werte bei einer geraden Anzahl!
- 3) Ein Multiple-Choice Test besteht aus 12 Fragen mit jeweils 4 Antwortmöglichkeiten (Kästchen zum Ankreuzen). Man bekommt für jede Frage 3 Punkte, wenn man alle 4 Antwortmöglichkeiten richtig hat. Man bekommt für jede Frage 1 Punkt, wenn man genau 3 der Kästchen richtig hat. Wie viele Punkte erhält man im Durchschnitt für den ganzen Test, wenn man jedes Kästchen mit der Wahrscheinlichkeit 88% richtig beantwortet? Simulieren Sie den Test 10000 Mal und werten Sie die erzielten Punkte mit einem `SimpleStat` Objekt aus. Sie verwenden am besten die Binomialverteilung `b{int n, double p}` als Verteilung: Summe von `n` 0-1 Spielen, wobei die 1 mit Wahrscheinlichkeit `p` auftritt.
- 4) Starten Sie mit dem Extra-Beispiel der Übung 9 (Polygon)
a) Schreiben Sie für den `Vec2`-Typ 2 Funktionen:
`schiebung(Vec2 v)` und `drehung(double phi)`
Beide Funktionen sollen Lambdas zurückgeben, mit denen man einen `Vec2 a` verschieben oder drehen kann, genauer gesagt sollten sie den verschobenen (`a+v`) oder gedrehten Vektor zurückgeben (also `a` nicht modifizieren). Die Formeln für die Drehung sind:
$$c = \cos(\phi), s = \sin(\phi) : \quad [x, y] \mapsto [c \cdot x - s \cdot y, s \cdot x + c \cdot y]$$

Das erste Lambda muss natürlich den Verschiebungsvektor `capturen`, und das zweite am besten `c, s`.
b) Schreiben Sie nun eine Funktion, die eine beliebige Vektortransformation `f` (wie z.B. die Lambdas aus a)) auf alle Punkte eines Polygons anwendet:
`void transform(Polygon& p, auto f)`
d.h. die Funktion soll das Polygon ändern. Drehen Sie das gegebene Polygon um $\phi = \pi/4$ und verschieben es um `[1, 2]`.

Extra-Aufgabe(n): Diese Aufgaben sind freiwillig und zählen 2 Punkte!

- 5) a) Lösen Sie Hausaufgabe 1 mit den Range-Algorithmen.
b) Lösen Sie Hausaufgabe 2 mit den Range-Algorithmen.