

**Name:****MatNr:****User:****Hinweis:** Denken Sie während der Klausur auch an die **Datensicherung!****Empfehlung:** Arbeiten Sie in kleinen Schritten und sorgen Sie dafür, dass Sie jederzeit ein lauffähiges Anwendungssystem haben.**Allgemeine Vorgaben:**

- **Benutzen Sie ausschließlich MS Visual Studio.**
- Verwenden Sie möglichst keine globalen Variablen oder Objekte und nur solche, die als **const** deklariert sind.
- Programmieren Sie C++-standardkonform (z.B. kein Aufruf von **system**, keine Ein-/Ausgabe im C-Stil)
- Verwenden Sie die vorgegebenen Bezeichner; eigene Bezeichner sind 'sprechende Bezeichner'.
- Beachten Sie das 'Principle of least privilege' (z.B. 'const-correctness').
- Die Lokalisierungsbibliothek (also z.B. **setlocale (LC\_ALL, "")**;) darf benutzt werden.

**Aufgabenstellung:**

Schreiben Sie ein Programm, das die Ergebnisse der Volkszählungen von 2011 und 1984 weiterverarbeitet.

**Anforderungen für die Note ausreichend:**

Ihnen steht die Textdatei 'zensus.txt' zur Verfügung, in der die Ergebnisse der Volkszählungen von 2011 und 1984 in Deutschland gespeichert sind.

- Lesen Sie folgende Daten in ein Objekt einer Klasse **Town** ein: Name des Bundeslandes, Name der Stadt, Gesamtbevölkerung 2011, männliche Bevölkerung 2011, weibliche Bevölkerung 2011, Gesamtbevölkerung 1984, männliche Bevölkerung 1984, weibliche Bevölkerung 1984. Diese Informationen sind fortlaufend für alle Gemeinden in Deutschland in der Datei enthalten, wobei alle Einträge jeweils durch ein newline-Zeichen (\n) getrennt sind.  
Alle Town-Objekte aus der Datei sollen in einen passenden Container (STL oder selbstdefiniert) abgespeichert werden.

Nach Abschluss des Einlesevorgangs soll die Anzahl der in diesem Container enthaltenen **Town**-Objekte ausgegeben werden.

- Alle Städte sollen alphabetisch aufsteigend nach ihren Namen sortiert (ASCII-Code genügt) in eine neue Textdatei 'Towns.txt' abgespeichert werden (s. Menüpunkt [1]). Dabei sollen jeweils der Name der Stadt und ihres Bundeslandes sowie ihre Gesamtbevölkerung 2011 gespeichert werden.

Realisieren Sie zusätzlich eine Menüschieleife, die es dem Benutzer erlaubt, die bisher behandelten Teilaufgaben (s. [1] und [9]) sowie die noch folgenden Teilaufgaben durch die folgenden Menüpunkte bearbeiten zu lassen:

- ```

[1] Alle Städte sortiert nach Namen in Datei speichern
[2] Einwohnerzahlen der Bundesländer anzeigen
[3] Städte sortiert nach größtem relativem Wachstum in Datei speichern
[4] Mehrfach vorkommende Städtenamen anzeigen

[9] Programm beenden
Ihre Wahl:

```

Die Menüpunkte [2] bis [4] können Sie vorläufig leer implementieren. Ihre Funktionalität wird dann beim Abarbeiten der folgenden Teilaufgaben ergänzt.

Anforderungen für die Noten befriedigend / gut / sehr gut (jede zusätzliche Leistung verbessert die Note um eine Stufe):

- Die Einwohnerzahlen aller Bundesländer sollen für das Jahr 2011 bestimmt und zusammen mit dem Namen des Bundeslandes alphabetisch sortiert nach diesen Namen ausgegeben werden (s. Menüpunkt [2]). Die Formatierung der gewünschten Konsolenausgabe ist hier zu sehen:

```
Ihre Wahl: 2
*** Einwohnerzahlen der Bundesländer ***

Baden-Württemberg.....10512441
Bayern.....12443372
...
```

- Das relative Wachstum aller Städte in Prozent von 1984 bis 2011 soll bestimmt und als Liste aller Städte (mit Namen des Bundeslandes und Gesamtbevölkerung 2011) sortiert nach absteigendem relativen Wachstum in einer weiteren Textdatei 'Growth.txt' gespeichert werden (s. Menüpunkt [3]). Hier ein Beispiel zur Formatierung des Anfangs der gewünschten Ausgabe:

|                                 |                    |    |
|---------------------------------|--------------------|----|
| 261.54%: Gemünd                 | Rheinland-Pfalz    | 34 |
| 230.00%: Ammeldingen an der Our | Rheinland-Pfalz    | 23 |
| 220.00%: Gröde                  | Schleswig-Holstein | 11 |
| ...                             |                    |    |
| ...                             |                    |    |

- Es gibt eine Reihe von Städten in Deutschland mit einem nicht eindeutigen Namen, wie beispielsweise 'Münster', welches in Hessen, Bayern und Nordrhein-Westfalen vorkommt (d.h. es gibt mehrere **Town**-Objekte mit identischem Datenelement für den Namen der Stadt). Geben Sie eine Auflistung dieser mehrfach (also mindestens doppelt) vorkommenden Städtenamen aus (s. Menüpunkt [4]). Hierbei soll absteigend nach der Häufigkeit des Vorkommens der Städtenamen und bei gleicher Häufigkeit alphabetisch (ASCII-Code genügt) sortiert ausgegeben werden. Zum Schluss dieser Auflistung soll noch die Anzahl der eindeutigen (also nicht mehrfach vorkommenden) Namen ausgegeben werden.  
Hier ein Ausschnitt der gewünschten Konsolenausgabe:

```
...
Mühlhausen....3
Münster.....3
Rieden.....3
...
Zwingenberg....2

Anzahl eindeutiger Namen: ?????
```

Als Ergebnis erwarte ich in jedem Fall ein **ausführbares** Anwendungssystem.  
Speichern Sie alle Ihre Quellcode dateien (.cpp und .h) und von den zusätzlichen Projektdateien und -ordnern **nur die .vcxproj-Datei** in Ihrem persönlichen Netzverzeichnis.  
Lassen Sie das Aufgabenblatt (mit Namen, Matrikel- und Usernummer) an Ihrem Platz liegen.

Note: