

---

Programmieren, Algorithmen und Datenstrukturen II

## Praktikum 1: Dateiverarbeitung

Sommersemester 2019

Prof. Dr. Arnim Malcherek

---

### Allgemeine Hinweise zum Praktikum:

- Bereiten Sie die Aufgaben unbedingt zu Hause oder in einem freien Labor vor. Das beinhaltet:
  - Entwurf der Lösung
  - Codieren der Lösung in einem Qt-Creator-Projekt
- Die Zeit während des Praktikums dient dazu, die Lösung testieren zu lassen sowie eventuelle Korrekturen vorzunehmen.
- Das Praktikum dient auch zur Vorbereitung der praktischen Klausur am Ende des Semesters. Versuchen Sie also in Ihrem eigenen Interesse, die Aufgaben selbständig nur mit Verwendung Ihrer Unterlagen bzw. Ihres bevorzugten C++-Buches und ohne Codefragmente aus dem Netz zu lösen.
- Die Lösung wird nur dann testiert, wenn
  - sie erklärt werden kann bzw. Fragen zur Lösung beantwortet werden können.
  - das Programm ablauffähig und die Lösung nachvollziehbar ist.
  - die Hinweise oder Einschränkungen aus der Aufgabenstellung befolgt wurden.
- Zur Erinnerung hier noch einmal die Regeln des Praktikums, die schon in der Vorlesung besprochen wurden:
  - Sie arbeiten in 2er Gruppen.
  - Ein Testat gibt es nur zum jeweiligen Termin.
  - Abschreiben und Kopieren ist verboten.
  - Es gibt keine Noten. Die Bewertung ist lediglich erfolgreich / nicht erfolgreich.
  - Das Praktikum ist Zulassungsvoraussetzung für die Klausur. Hierfür müssen alle fünf Praktikumsübungen testiert sein.

**Lernziele:** Sie lernen Text- und Binärdateien - auch komplexerer Art - sicher und effizient zu verarbeiten.

In diesem Semester wollen wir ein Verwaltungsprogramm für das Reisebüro ‚Upandaway‘ erstellen. Im ersten Praktikum legen wir das Grundgerüst und schreiben ein Programm, mit dem wir existierende Daten aus einem Fremdsystem übernehmen können.

### Aufgabe 1

Legen Sie Klassen zu dem Klassendiagramm aus Abbildung 1 an.

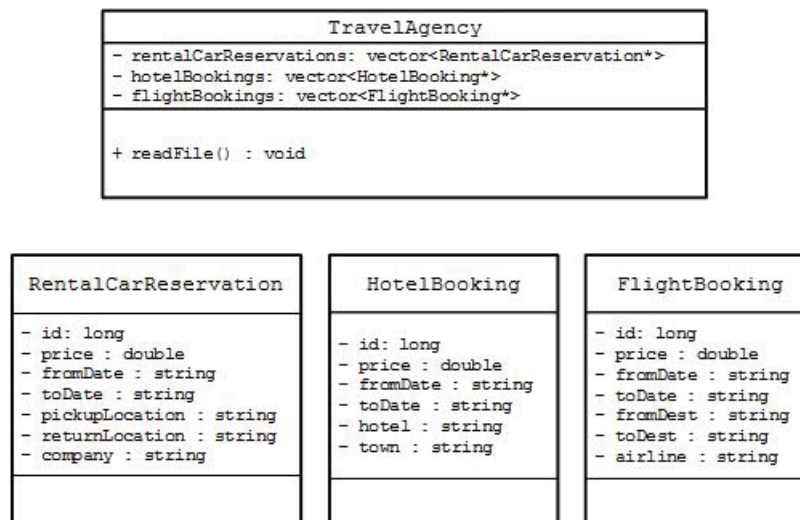


Abbildung 1: Klassendiagramm

### Aufgabe 2

Legen Sie im `main` Programm eine Instanz der Klasse `TravelAgency` an und rufen anschließend die Methode `readFile()` auf. Bei der Datenübernahme aus dem Altsystem wurden aus Bequemlichkeit alle Daten in einer (nicht sortierten) Datei *bookings.txt* zusammengepackt. Die Datei liegt auf Moodle und hat den folgenden Aufbau:

```

F|35|277.01|20150602|20150603|STR|WAW|Lufthansa
R|36|167.01|20150602|20150603|CHOPIN FLUGH.|CHOPIN FLUGH.|Europcar
H|37|256.02|20150602|20150603|Residence 1898|Warschau
F|38|288.93|20150603|20150603|WAW|STR|Lufthansa
    
```

Die Spalten werden jeweils durch das Zeichen `|` voneinander getrennt. Spalte 1 beschreibt den Typ der Buchung. Die Spalten 6-8 hängen vom Typ ab. Im Einzelnen:

- Spalte 1: Typ der Buchung (F für Flight, H für Hotel, R für Rental Car)  
 Spalte 2: ID der Buchung (Attribut `id`)  
 Spalte 3: Preis der Buchung (Attribut `price`)  
 Spalte 4: Start-Datum der Buchung im Format JJJJMMTT (Attribut `fromDate`)  
 Spalte 5: End-Datum der Buchung im Format JJJJMMTT (Attribut `toDate`)  
 Spalte 6: falls Typ = 'F': Startflughafen (`FlightBooking::fromDest`)  
           falls Typ = 'R': Abholstation des Autos (`RentalCarReservation::pickupLocation`)  
           falls Typ = 'H': Hotelname (`HotelBooking::hotel`)  
 Spalte 7: falls Typ = 'F': Zielflughafen (`FlightBooking::toDest`)  
           falls Typ = 'R': Rückgabestation des Autos (`RentalCarReservation::returnLocation`)  
           falls Typ = 'H': Name der Stadt (`HotelBooking::town`)  
 Spalte 8: falls Typ = 'F': Fluglinie (`FlightBooking::airline`)  
           falls Typ = 'R': Autoverleihfirma (`RentalCarReservation::company`)

Legen Sie jeweils die passenden Objekte an, und speichern Sie sie in den Attributen `flightBookings`, `hotelBookings` und `rentalCarReservations` Ihrer Instanz von `TravelAgency`. Geben Sie eine Meldung aus, wie viele Buchungen Sie jeweils von welchem Typ und mit welchem Gesamtwert angelegt haben.

Beispiel: Es wurden `a` Flugbuchungen im Wert von `b €`, `c` Mietwagenbuchungen im Wert von `d €` und `e` Hotelreservierungen im Wert von `f €` angelegt.

### Aufgabe 3

Legen Sie jetzt in `TravelAgency` die Methode `readBinaryFile` an. Fügen Sie in `main` eine Abfrage ein, welche Datei eingelesen werden soll, und rufen Sie dann die entsprechende Methode auf. In `readBinaryFile` lesen Sie die Binärdatei `bookingsBinary.bin` ein. Diese ist so ähnlich aufgebaut wie die Textdatei aus Aufgabe 2 mit folgenden Anpassungen:

- Die Id ist im Format `long` vorhanden
- Der Preis ist im Format `double` vorhanden
- Die Flughafenkürzel sind dreistellige `char`-Arrays
- Namen von Hotels, Städten, Abhol- und Rückgabestationen und Firmen sind 15-stellige `char`-Arrays. Inhalte sind entweder nach 15 Stellen abgeschnitten oder aufgefüllt mit Leerzeichen.
- Der Typ der Buchung ist nach wie vor ein einzelnes `char`
- Das Datum ist ein achtstelliges `char`-Array.

Verfahren Sie jetzt genauso wie in Aufgabe 2: Lesen Sie die Datei ein, und legen Sie die Objekte an. Geben Sie wieder die gleiche Meldung wie in Aufgabe 2 aus.

Um zu prüfen, dass Sie auch die zeichenartigen Werte richtig eingelesen haben, geben Sie zusätzlich jeweils alle Objektwerte der teuersten Buchung einer Kategorie aus.

Beispiel (Werte stimmen nicht):

Teuerste Flugbuchung (Id 53): Vom 20150302 bis 20150303 von FRA nach GRU mit Lufthansa zum Preis von 3456.23 Euro.

Teuerste Mietwagenbuchung (Id 35): Vom 20150102 bis 20150401 von Hamburg Flughafen nach Aarhus Flughafen mit AVIS zum Preis von 2112.31 Euro.

Teuerste Hotelbuchung (Id 63): Vom 20150401 bis 20150410 im Bellagio in Las Vegas zum Preis von 6432.12 Euro.