
Programmieren, Algorithmen und Datenstrukturen II
Praktikum 3: Graphisches User-Interface

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, die Qt-Library einzusetzen, um ein einfaches User-Interface aufzubauen.

In diesem Praktikum erstellen wir ein User Interface für unsere Reisebürosoftware. Die Aufgaben bauen aufeinander auf. Auch wenn zur Abnahme die letzte Aufgabe reicht, empfehle ich Ihnen sehr, die Aufgaben einzeln als Projekt anzulegen, da dies auch die Vorgehensweise in der Klausur sein wird.

Aufgabe 1

Erstellen Sie ein einfaches Desktop-UI für Ihr Programm, mit dem Sie in der Lage sind, das Einlesen der Buchungen anzustoßen sowie Buchungen anzuzeigen. Abbildung 1 zeigt ein mögliches Beispiel (Sie dürfen aber auch andere Realisierungen verwenden (Buttons, Icons, etc.)):

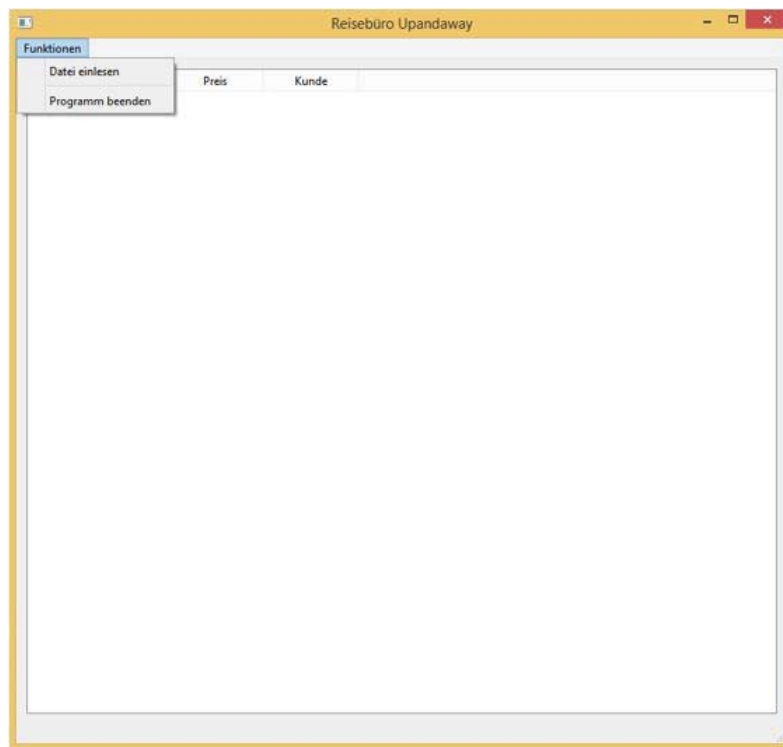


Abbildung 1: Beispiel-Layout 1

Beim Anklicken des Menü-Eintrags ‚Datei einlesen‘ soll die Datei mit den Buchungen aus dem zweiten Praktikum eingelesen werden. Anschließend soll ein Popup wie in Abbildung 2 gezeigt werden, in dem das Ergebnis des Einlesens angezeigt wird:

Bei Klick auf ‚OK‘ soll sich dieses Fenster wieder schließen, und alle Buchungen sollen wie in Abbildung 3 angezeigt werden.

Tipp: Verwenden Sie für das Popup-Fenster ein Objekt vom Typ `QMessageBox::information`. Informationen zur Verwendung finden Sie in der Online Hilfe und im Beispiel *‘Standard Dialogs Example’*.

Zur Anzeige wird ein Tabellen-Container verwendet. Diesen finden Sie als *Table Widget* im UI-Designer. Lesen Sie die Online-Doku zum Objekt `QTableWidget`, um herauszufin-



Abbildung 2: Beispiel Popup-Fenster

| Buchungsnummer | Preis | Kunde | Reisennummer |
|----------------|---------|-------------------|--------------|
| 70 | 1199 | Volker Graul | 14 |
| 71 | 466 | Volker Graul | 14 |
| 72 | 501.03 | Volker Graul | 14 |
| 73 | 1358 | Volker Graul | 14 |
| 74 | 330 | Volker Graul | 15 |
| 75 | 751.03 | Volker Graul | 15 |
| 76 | 290 | Volker Graul | 15 |
| 29 | 258.03 | Roland Peitsch | 5 |
| 30 | 251.03 | Roland Peitsch | 5 |
| 31 | 316.01 | Hans-Werner Moors | 6 |
| 32 | 111.01 | Hans-Werner Moors | 6 |
| 33 | 381.01 | Hans-Werner Moors | 6 |
| 34 | 288.93 | Hans-Werner Moors | 6 |
| 35 | 277.01 | Uli Stein | 7 |
| 36 | 167.91 | Uli Stein | 7 |
| 37 | 256.02 | Uli Stein | 7 |
| 38 | 288.93 | Uli Stein | 7 |
| 54 | 2892.04 | Eduard Angele | 11 |
| 55 | 234.91 | Eduard Angele | 11 |
| 56 | 1345.82 | Eduard Angele | 11 |
| 57 | 2986 | Eduard Angele | 11 |
| 58 | 224.19 | Eduard Angele | 11 |

Abbildung 3: Beispiel Layout Buchungen

den, wie man diesen Container befüllt.

Um den Kundennamen anzuzeigen, müssen Sie ihn erst über den Umweg der Vektoren `allBookings`, `allTravels` und `allCustomers` aus dem jeweiligen Objekt `Customer` heraussuchen.

Achten Sie darauf, dass bei erneutem Anklicken des Menüeintrags ‚Datei einlesen‘ die Tabelle erst wieder gelöscht wird. Alternativ können Sie eine erneute Auswahl des Menüeintrags nach erfolgreichem Lesen auch unterbinden, indem Sie den Menüeintrag anschließend ausgrauen. Wählen Sie selbst eine Möglichkeit. Was nicht passieren darf, ist, dass Sie die gleichen Buchungen nochmal hinzufügen und dann doppelt anzeigen.

Bei Klick auf den Menüeintrag *Programm beenden* soll das Programm ohne weitere Abfragen beendet werden.

Aufgabe 2

Erweitern Sie jetzt Ihr UI, um eine Detailanzeige: Bei Doppelklick in eine Zeile der Tabelle soll in einem zusätzlichen Detaildisplay die Buchungsnummer sowie das Von-Datum und das Bis-Datum wie im Beispiel in Abbildung 4 angezeigt werden.

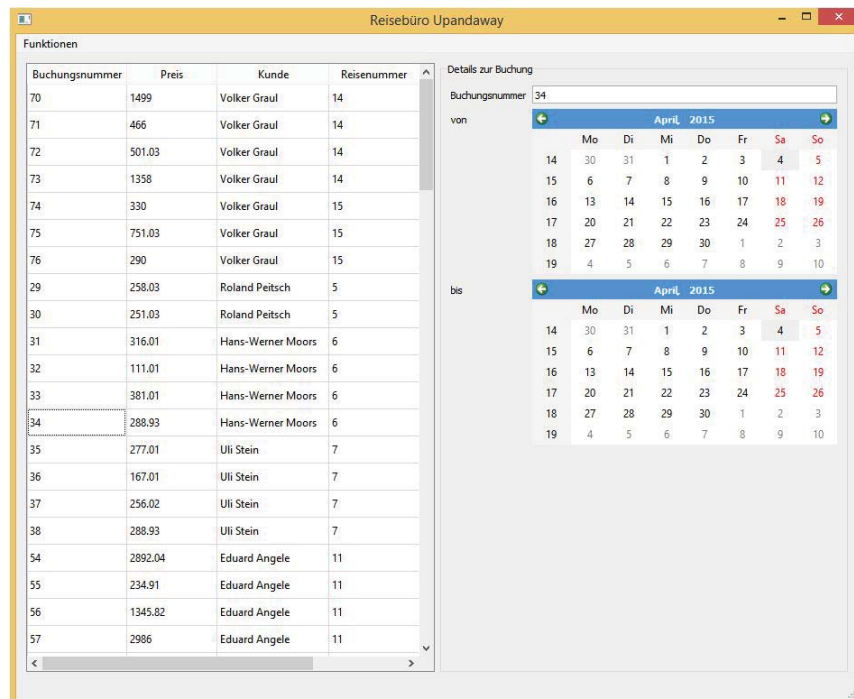


Abbildung 4: Kalendercontrols

Aufgabe 3

Fügen Sie einen Menüeintrag (oder ähnliches, je nach Lösung in Aufgabe 1) hinzu, mit dem Sie eine Buchung anlegen können. Erfassen Sie die Buchungsdetails (hier können Sie wieder selbst entscheiden, wie Sie es realisieren z.B. Popup-Fenster mit allen nötigen Details, eigener Bereich im Hauptfenster, aufeinanderfolgende Dialogabfragen in eigenen Popups etc.). Verwenden Sie zum Anlegen der Buchung die Methode `createBoooking` aus dem letzten Praktikum. Die angelegte Buchung soll anschließend in der Tabelle der Buchungen im Hauptfenster miterscheinen.

Abbildung 5 zeigt eine mögliche Beispielrealisierung.

The image shows a standard Windows-style dialog box titled "Neue Buchung". It contains the following fields and controls:

- Preis:** A text box containing "0,00".
- Reise-ID:** A text box containing "0".
- Startdatum:** A date picker showing "01.01.2000".
- Enddatum:** A date picker showing "01.01.2000".
- Buchungstyp:** A dropdown menu with "Flugbuchung" selected.
- Startflughafen:** An empty text box.
- Zielflughafen:** An empty text box.
- Fluglinie:** An empty text box.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

Abbildung 5: Beispiellösung zum Anlegen neuer Buchung

Tipp für diejenigen, die noch nicht soviel Programmiererfahrung haben: Es ist einfacher, alles in einem Fenster ablaufen zu lassen, auch wenn es nicht so schön aussieht, aber in diesem Praktikum geht es (noch) nicht um Schönheitspreise. Das heißt, Sie können das Anlegen einer neuen Buchung einfach in einem Bereich des Hauptfensters starten, den Sie für diesen Zweck reservieren.

Wenn Sie eine Lösung mit einem eigenen Fenster programmieren, müssen Sie die Kommunikation zwischen den Fenstern sicherstellen (die neu angelegte Buchung soll ja gleich mitangezeigt werden). Das funktioniert am einfachsten, indem Sie dem Popup-Fenster im Konstruktor einen Zeiger auf den Aufrufer mitgeben, den das Popup dann zum Aufruf von Methoden des Hauptfensters nutzen kann (callback). Alternativ könnten Sie auch mit fensterübergreifenden Signals und Slots arbeiten, aber das ist schon etwas fortgeschritten und nur für interessierte Freiwillige.