

# 3. Praktikum

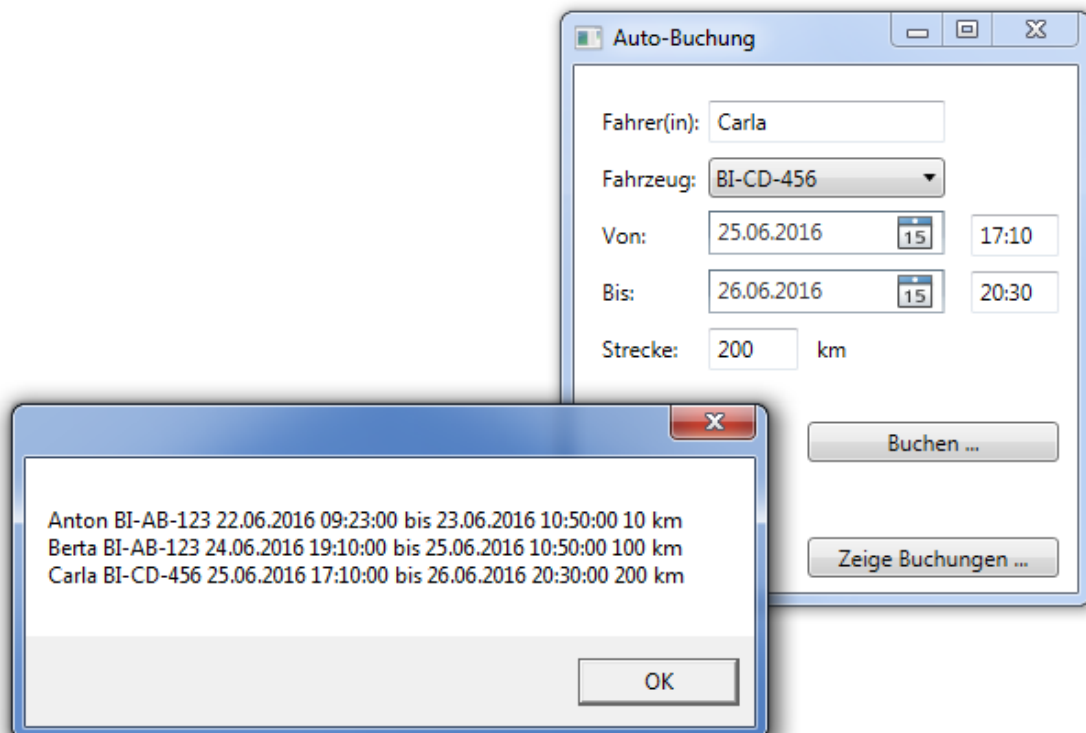
Jörn Loviscach

Versionsstand: 20. Juni 2016, 16:35



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

In diesem Praktikum soll eine einfache Buchungssoftware für Elektroautos entwickelt werden: Man kann seinen Namen eingeben, eines der Autos auswählen, den Zeitraum für die Buchung wählen und angeben, welche Strecke man zurücklegen will. Die Software prüft dann, ob das Auto verfügbar ist und ob es für diese Fahrt und die Fahrten danach genügend Ladezeit in der Garage besitzt. (Schreiben Sie erst den Teil des Software, der prüft, ob das Auto verfügbar ist, erst danach den Teil, der auch den Ladezustand prüft.) Ist die Buchung möglich, wird sie gespeichert. Die Software informiert den Benutzer, ob die Buchung gelungen ist. Auf Knopfdruck lassen sich alle gespeicherten Buchungen in einer MessageBox anzeigen.



Sie können zum Beispiel Klassen dieser Art schreiben:

```
class EAuto {
    kennzeichen : string
    ladezeit : double
    maximaleReichweite : double
}

EAuto {
    EAuto(string kennzeichen, double maximaleReichweite, double ladezeit)
    ToString() : string
}
```

```
class Verwaltung {
    buchungen : List<Buchung>
}

Verwaltung {
    ListeBuchungen() : string
    VersucheBuchung(string wer, EAuto fahrzeug, DateTime von, DateTime bis, double strecke) : bool
}
```

```
class Buchungszeitraum {
    bis : DateTime
    von : DateTime
}

Buchungszeitraum {
    Buchungszeitraum(DateTime von, DateTime bis)
    ToString() : string
}

Buchungszeitraum {
    ÜberschneidetSichMit(Buchungszeitraum z) : bool
    ÜberschneidetSichNichtMit(Buchungszeitraum z) : bool
    ZeitDanachBis(Buchungszeitraum z) : double
}
```

```
class Buchung {
    fahrzeug : EAuto
    strecke : double
    wer : string
}

Buchung {
    Buchung(string wer, EAuto fahrzeug, DateTime von, DateTime bis, double strecke)
}

Buchung {
    Fahrzeug() : EAuto
    Strecke() : double
    Zeitraum() : Buchungszeitraum
}
```

Die Ladezeit der Klasse EAuto bedeutet die Ladezeit (in Stunden) vom minimalen Akkustand bis zur maximalen Akkufüllung. Die Methode VersucheBuchung der Klasse Verwaltung liefert true zurück, wenn die Buchung erfolgreich war, sonst false. In einigen der Klassen ist die virtuelle Methode ToString überschrieben, damit man leichter Ausgaben machen kann beziehungsweise damit die Fahrzeuge in der ComboBox mit ihren Kennzeichen erscheinen.

Im Konstruktor von MainWindow können Sie Fahrzeuge konstruieren und direkt in die ComboBox zur Auswahl eintragen:

```
comboBoxFahrzeug.Items.Add(new EAuto("BI-AB-123", 100.0, 10.0));
comboBoxFahrzeug.Items.Add(new EAuto("BI-CD-456", 200.0, 7.0));
comboBoxFahrzeug.Items.Add(new EAuto("BI-EF-789", 150.0, 9.0));
```

Die Eingaben in die Textfelder können Sie mit den jeweils passenden statischen Parse-Methoden in Werte umwandeln. Fangen Sie dabei auftretende Exceptions. Werfen Sie selbst eine Exception (und fangen diese an dieser Stelle), wenn der Endezeitpunkt der Buchung vor dem Anfangszeitpunkt liegt.