

Aufgabe Flugplaner in C++, QT und SQLite

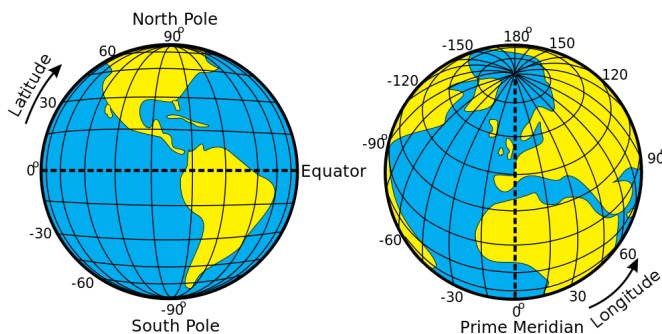
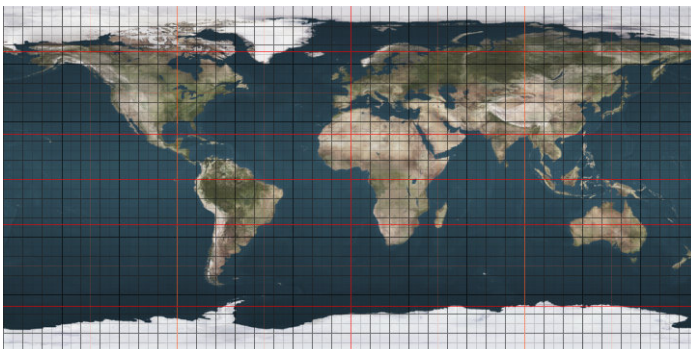
Aufgabe

Das Programm soll auf Daten zu Flügen, Flughäfen,... in einer Datenbank zugreifen und dem Benutzer die Möglichkeit bieten, entweder über ComboBoxen oder über eine Suche nach Namen oder IATA Kurzbezeichnung (z.B. VIE für Wien) den Start- und Endflughafen festzulegen. Zusätzlich soll optional eine Fluglinie ausgewählt werden können.

Danach soll eine Suche aller möglichen Verbindungen erfolgen und die Verbindung mit den wenigsten Zwischenstopps auf der Weltkarte mit Linien und den IATA Kurzbezeichnungen der Flughäfen angezeigt werden. Wenn mehrere Routen mit gleich vielen Zwischenstopps gefunden wurden, sollen die von den der gewählten Fluglinie oder der Allianz der gewählten Fluglinie geflogenen Verbindungen bevorzugt werden. Flüge der gewählten Fluglinie sollen durch rote Linien dargestellt werden, Flüge der Allianz durch blaue Linien und die restlichen Flüge durch graue Linien. Die IATA Kurzbezeichnungen der Flughäfen sollen in schwarzer Schrift angezeigt werden.

In einer Tabelle sollen außerdem noch die Details des gesamten Fluges (Einzelflüge mit Linie,...) angezeigt werden.

Das Bild der Weltkarte hat den Äquator und den Nullmeridian als Mittelpunkt. Die Latitude (Geographische Breite) ist in den Daten mit $\pm 0^\circ$ bis $\pm 90^\circ$ angegeben und entspricht der Höhe des Bildes und die Longitude (Geographische Länge) ist in den Daten mit $\pm 0^\circ$ bis $\pm 180^\circ$ angegeben und entspricht der Breite des Bildes immer ausgehend vom Mittelpunkt. Die linke obere Ecke des Bildes entspricht also Latitude $+90$ und Longitude -180 während die rechte untere Ecke Latitude -90 und Longitude 180 entspricht.



Daten

- [Flugdaten](#)

Hinweise

- [Zeichnen mit QT](#)
- Beachten Sie, dass das Widget zum Zeichnen nicht die anderen Steuerelemente überdeckt, sonst werden diese nicht mehr funktionieren.
- Die Flugdaten befinden sich in einer [SQLite](#) Datenbank.

SQLite über QT

- [QSqlDatabase](#)
- [SQL Database Drivers](#)
- [QT + SQL Tutorial](#)

SQLite ohne QT

- [SQLite C++ Libs Tutorial](#)
- [Umgang mit der Callback Funktion](#)
- [Alternative zur callback-Funktion](#)

Datenbankstruktur

In der **Airline**-Tabelle der Datenbank sind folgende Daten enthalten:

- id --> ID der Airline --> int, z.B. 236
- name --> Name der Airline --> String, z.B. Austrian Airlines
- alliance --> Allianzzugehörigkeit --> int, z.B. 2

In der **Airport**-Tabelle der Datenbank sind folgende Daten enthalten:

- id --> ID des Airports --> int, z.B. 2091
- latitude --> Latitude der Koordinaten --> float, z.B. 48.1103
- longitude --> Longitude der Koordinaten --> float, z.B. 16.5697
- name --> Name des Airports --> String, z.B. Vienna
- iata --> IATA-Code des Airports --> String, z.B. VIE

In der **Alliance**-Tabelle der Datenbank sind folgende Daten enthalten:

- id --> ID der Allianz --> int, z.B. 2
- name --> Name der Allianz --> String, z.B. SkyTeam

In der **Route**-Tabelle der Datenbank sind folgende Daten enthalten:

- airline --> ID der Airline --> int, z.B. 236
- airport1 --> 1. Flughafen der Route --> int, z.B. 2091
- airport2 --> 2. Flughafen der Route --> int, z.B. 1005

Abgabe

gepackt, sodass Erstellen mit Auspacken, make (cmake, qmake, make) möglich ist.