

Thema, Ziele: Temperatur-Umrechnung, Zeichnen und Malen mit Qt, "Qt-Designer".

Aufgabe 1: Temperatur-Umrechnung mit Interaktion

Das Widget zur Temperatur-Umrechnung von Aufgabe 3 der letzten Übung soll nun um interaktives Verhalten erweitert werden. Die Umrechnung soll jetzt also neu wirklich funktionieren.

Dazu einige Tipps:

- Das untere "QLineEdit"-Objekt dient nur zur Anzeige des Resultates der Umrechnung. Es ist deshalb "Read-Only" zu setzen. Dazu dient die Methode `leResult->setReadOnly(true);`.
- Von den beiden RadioButtons sollte bereits zu Beginn einer aktiviert sein. Dazu dient die Methode `rbCelsius->setChecked(true);`
- Es ist sinnvoll zuerst nur das interaktive Verhalten für den "Rechne"-Button zu implementieren. Dazu ist eine Slot-Funktion `void onRechne()` oder ähnlich zu definieren, welche beim Anklicken des Buttons aufgerufen wird. Diese Funktion liest dann den Text des Eingabefelds vom `leEingabe` und versucht diesen "QString" in einen "double" umzuwandeln. Falls dies möglich ist, kann dann der Wert (abhängig vom aktivierten RadioButton) in Fahrenheit oder Celsius umgerechnet werden und danach im `leResult` angezeigt werden.
- Ein "QString s" kann folgendermassen in einen "double x" umgewandelt werden:

```
bool okay;  
double x = s.toDouble(&okay);  
if (okay) ...
```
- Falls diese Umwandlung *nicht* geklappt hat, kann im `leResult` eine Fehlermeldung angezeigt werden (z.B. "Eingabe ungültig" o.ä).
- Zur Umwandlung eines double Wertes y in einen "QString" kann die Klassenmethode `number(...)` der Klasse "QString" verwendet werden. (Siehe Klassenbeschreibung.)
- Die Abfrage ob ein RadioButton aktiviert ist kann wie folgt durchgeführt werden:

```
if ( rbCelsius->isChecked() ) ...
```
- Nachdem die Umrechnung mit dem "Rechne"-Button funktioniert, kann nun noch dafür gesorgt werden, dass die Umrechnung automatisch vorgenommen wird, sobald die Eingabe im `leEingabe` mit der Return-Taste abgeschlossen wird. – Tipp: Suchen Sie ein passendes Signal des Eingabefeldes!

Aufgabe 2: "Zeichnen und Malen mit Qt" – Analog-Uhr

Im Unterverzeichnis "Vorlage_ueb14_qt4\AnalogWatch" finden Sie ein Qt-Programm für eine Analog-Uhr. Leider sind das Ziffernblatt sowie der Minuten- und Stundenzeiger abhanden gekommen.

Beheben Sie diese Mängel. Ziel ist, dass die resultierende Uhr am Schluss wie die berühmte SBB-Uhr aussieht (siehe Bild). Der Schriftzug "MONDAINE / SBB CFF FFS" kann/soll weggelassen werden.

Tipp: Gehen Sie schrittweise vor! (Das Ziffernblatt am Schluss.)



Aufgabe 3: Temperatur-Umrechnung mit Dialogfenster zur Fehlermeldung

Bei einer Fehleingabe beim Widget zur Temperaturumrechnung von Aufgabe 1 soll neu ein Popup-Window mit einer Fehlermeldung erscheinen – siehe nebenstehendes Bild.

Dazu kann die statische Methode `QMessageBox::warning(...)` wie folgt aufgerufen werden:

```
QMessageBox::warning(this, "Temperaturumrechnung",  
                    "Eingabe ist falsch!");
```



Damit dieses sogenannte Dialog-Fenster nach dem "Wegklicken" nicht automatisch wieder neu erscheint, muss der Fehler beseitigt werden. Dazu kann z.B. der Wert 0 in das `leEingabe` geschrieben werden:
`leEingabe->setText("0");`

Aufgabe 4: "Temperatur-Umrechnung" mit Qt-Designer erstellen – *fakultativ*

Erstellen Sie das Qt-Programm zur Temperatur-Umrechnung gemäss Aufgabe 3 der letzten Übung und bzw. Aufgaben 1 und 3 dieser Übung möglichst weitgehend mit Hilfe des Qt-Designers.

Erstellen Sie vorerst nur das GUI, noch ohne Dialogfenster.

Beachten Sie aber auch: der Weg ist das Ziel. – Experimentieren Sie also ruhig mit dem Qt-Designer "drauf los".

*Ich hoffe, dass es mir gelungen ist Ihnen etwas beizubringen,
es Ihnen auch Spass gemacht hat und
wünsche Ihnen schöne und erholsame Semesterferien!*