

Thema, Ziele: Qt: Signals and Slots, Verschachtelung von Layouts, Widgets.

Aufgabe 1: Qt Counter: Eigene Widget-Klasse mit Verschachtelung

Verwenden Sie als Basis Ihre Lösung (evt. die Musterlösung) von Aufgabe 5 der letzten Übung (Qt-Counter als eigene Widget-Klasse).

Ändern Sie die Darstellung mit Hilfe von *verschachtelten Layouts* (ohne "QGridLayout" zu verwenden!) so, dass ein Counter mit folgendem Layout dargestellt wird:

- Erste Zeile: Wert des Zählers, über die ganze Breite ("QLabel").
- Zweite Zeile: Increment- und Decrement-Button, nebeneinander.
- Dritte Zeile: Reset-Button, über die ganze Breite.

Aufgabe 2: Qt Counter: Signal und Slots

Erweitern, modifizieren Sie das Programm von der vorherigen Aufgabe so, dass die ursprünglich "Counter-Klasse", also die ohne Slot "ResetValue()" oder ähnlich zum Einsatz kommt. Der Reset-Button soll aber trotzdem funktionieren.

Tipp: In welcher Klasse könnte man nun einen eigenen Slot definieren, wenn man das nicht mehr in der Klasse Counter machen darf? – Was ist an dieser Lösung unschön?

Aufgabe 3: Temperaturumrechnung

Erstellen Sie als eigene Widget-Klasse (von "QWidget" ableiten) ein Programm mit folgender Darstellung:

Temperatur Umrechnung:

Eingabe: ☒ °C ☐ °F

Ergebnis=

Es kommen hier folgende Widgets vor: QLabel, QPushButton, QLineEdit, QRadioButton. Verwenden Sie um die beiden Radiobutton's zusammenzufassen ein "QGroupBox"-Widget. Zur Darstellung kann "QGridLayout" eingesetzt werden.

Das interaktive Verhalten (Umrechnung) braucht noch nicht implementiert zu werden.

Aufgabe 4: Signal and Slots – Verständnisfragen, ohne Computer lösen

Studieren Sie im Qt-Buch "C++-GUI-Programming-with-Qt-4-1st-ed.pdf" (Skript-Server, Verzeichnis "Qt-Book_Blanchette_Summerfield") das Unterkapitel "Signals and Slots in Depth", ab Seite 20 (Pdf: Seite 38).

Beurteilen Sie anschliessend die folgenden "connect()-Anweisungen. Welche sind zulässig, welche nicht? Wenn nicht, warum nicht?

```
QObject::connect( button, SIGNAL( clicked () ),      // 1
                  x1, SLOT( clearValue() ) );

QObject::connect( button, SIGNAL( clicked () ),      // 2
                  c, SIGNAL( valueChanged() ) );

QObject::connect( a, SIGNAL( valueChanged() ),       // 3
                  b, SLOT( setValue(0) ) );

QObject::connect( a, SLOT( newValue() ),             // 4
                  b, SLOT( clearValue() ) );

QObject::connect( spinBox, SIGNAL( valueChanged(int) ), // 5
                  x, SLOT( setValue(int) ) );
```