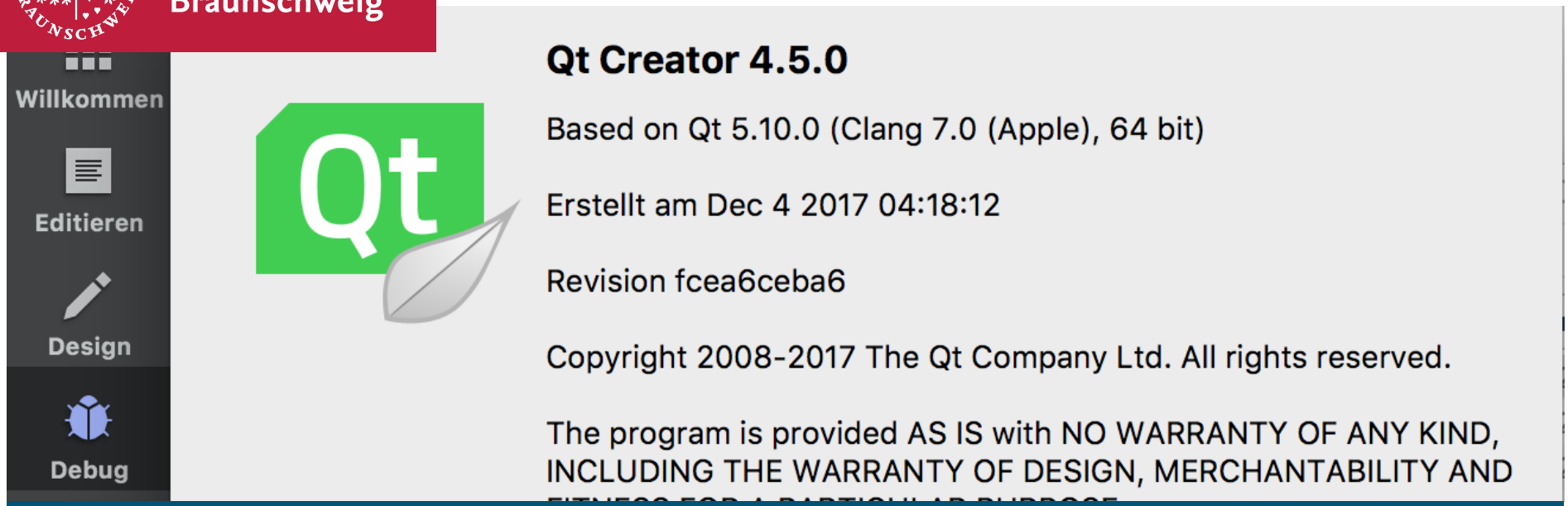
Technische Universität Braunschweig

Institut für Flugführung



GUI-Erstellung mit Qt

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, Andreas Dekiert M. Sc.,
25. Juni 2019

Agenda

09. April	Einführung
16. April	Softwareprojektmanagement
23. April	Entwicklungstools
30. April	GitHub
07. Mai	Software-Dokumentation und Bug-Reporting
14. Mai	Einführung Arduino
21. Mai	Frei
28. Mai	Dateieingabe und -ausgabe
4. & 11. Juni	Tag der Lehre und Exkursionswoche
18. Juni	Einführung von Qt
25. Juni	**GUI-Erstellung mit Qt**
02. Juli	Serielle Kommunikation
09. Juli	API-Anleitung und Projektarbeit
16. Juli	Vorbereitung der Abgabe und Fragen
12. August 10:00	Abgabe

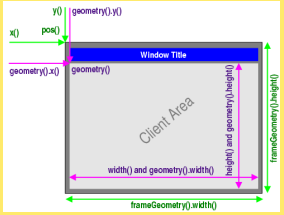
Lehrziele

GUI-Erstellung mit Qt

Als Teilnehmer soll ich am Ende dieser Übung...

- ☐ die QWidget-Klasse kennen
- ☐ GUIs in Qt erstellen können

GUI



GUI mit Qt

RGB-GUI

Qt GUI/Widgets Module

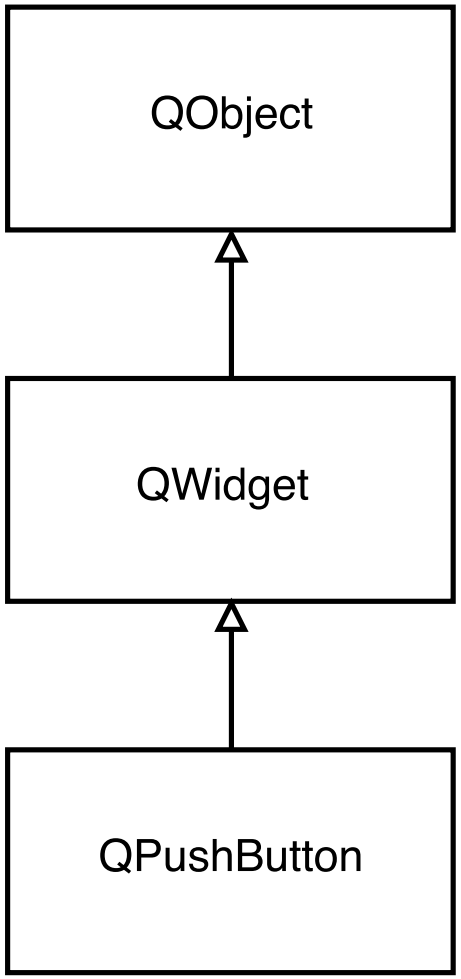
- Qt besitzt eigene Module für GUIs
  - Qt GUI
    Zentrales Modul für graphische Elemente
  - Qt Widgets
    High-Level Objekte, wie z. B. Fenster, Slider und Buttons
- Diese Module enthalten Klassen für eine plattformübergreifende GUI-Programmierung
- Module werden automatisch bei der Erstellung einer Qt-Widgets-Applikation eingebunden

Bei der Konsolenapplikation werden die Module nicht geladen!

Qt-Widgets

- Elementare Bausteine für alle GUI-Elemente
- Jedes darstellbare GUI-Element ist von der Klasse `QWidget` abgeleitet
- `QWidget` Klassen sind wiederum von der `QObjekt` Klasse abgeleitet

$\Rightarrow$ Verwendung von Signals und Slots möglich



Widgets

Geometrie

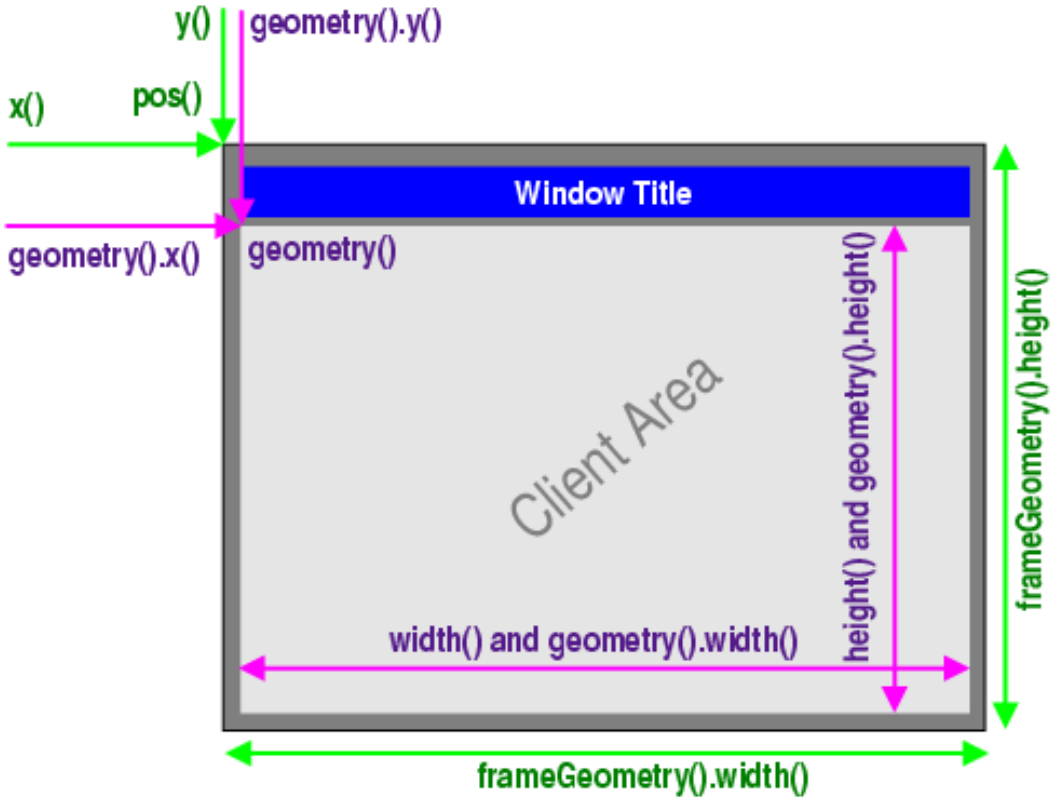


Abbildung 1: Übersicht der Geometriefunktionen für eine `QWidget`-Klasse[1]

[1] http://doc.qt.io/qt-5/application-windows.html

Widgets

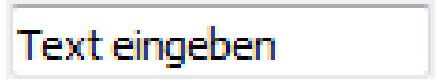
Übersicht

- `QPushButton`

Widgets

Übersicht

- `QPushButton`
- `QLineEdit`



Widgets

Übersicht

- `QPushButton`
- `QLineEdit`
- `QComboBox`

Widgets Übersicht

- `QPushButton`
- `QLineEdit`
- `QComboBox`
- `QSpinBox`

Widgets

Übersicht

- QPushButton
- QLineEdit
- QComboBox
- QSpinBox
- QCheckBox

Widgets

Übersicht

- QPushButton
- QLineEdit
- QComboBox
- QSpinBox
- QCheckBox
- QRadioButton

Widgets

Übersicht

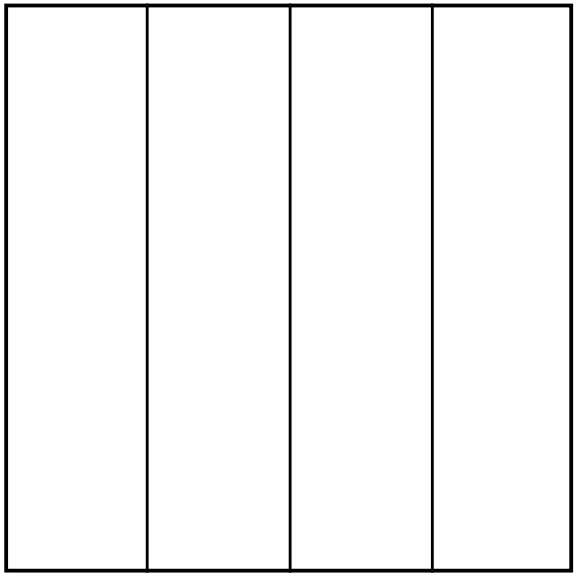
- QPushButton
- QLineEdit
- QComboBox
- QSpinBox
- QCheckBox
- QRadioButton
- QSlider

Layout

Übersicht

- Layouts positionieren die enthaltenen Widgets
- Neben Widgets können auch Layouts hinzugefügt werden
  $\rightarrow$ Verschachtelung möglich

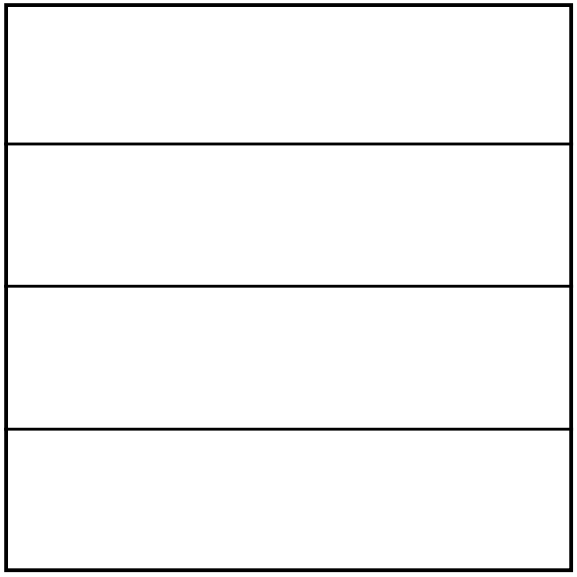
- `QHBoxLayout`



Layout Übersicht

- Layouts positionieren die enthaltenen Widgets
- Neben Widgets können auch Layouts hinzugefügt werden
  $\rightarrow$ Verschachtelung möglich

- `QHBoxLayout`
- `QVBoxLayout`

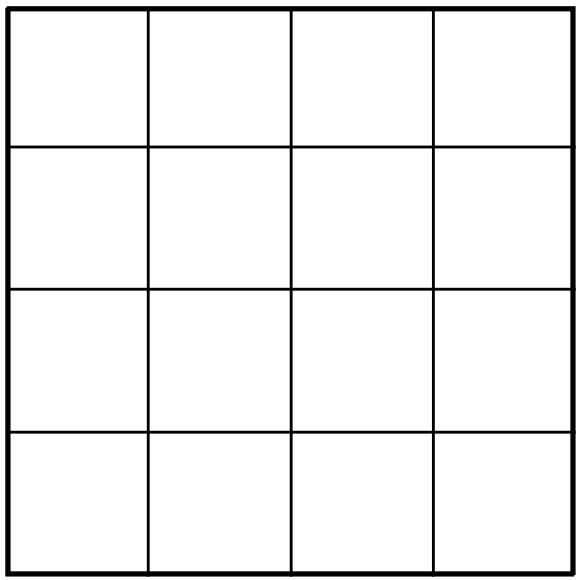


Layout

Übersicht

- Layouts positionieren die enthaltenen Widgets
- Neben Widgets können auch Layouts hinzugefügt werden
  $\rightarrow$ Verschachtelung möglich

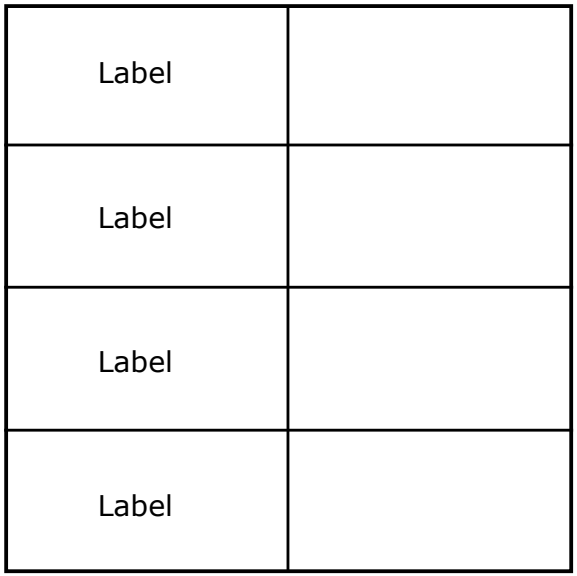
- `QHBoxLayout`
- `QVBoxLayout`
- `QGridLayout`



Layout Übersicht

- Layouts positionieren die enthaltenen Widgets
- Neben Widgets können auch Layouts hinzugefügt werden
  $\rightarrow$ Verschachtelung möglich

- `QHBoxLayout`
- `QVBoxLayout`
- `QGridLayout`
- `QFormLayout`



Gibt es Fragen oder Anmerkungen zu dem Unterthema
GUI mit Qt - Einführung und Elemente?

Abgehakt

GUI-Erstellung mit Qt

Als Teilnehmer soll ich am Ende dieser Übung...

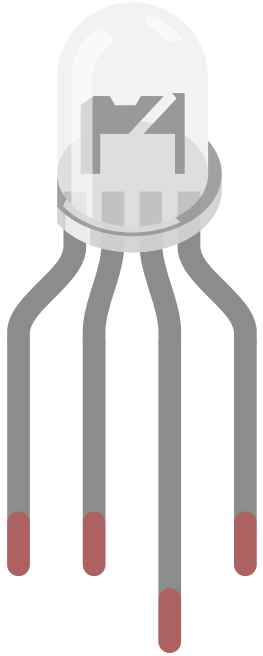
- [x] die QWidget-Klasse kennen
- [ ] GUIs in Qt erstellen können

Übung: GUI zur LED-Farbwahl

Es soll eine GUI programmiert werden, um eine an einen Arduino angeschlossene RGB-LED zu steuern.

RGB-LED

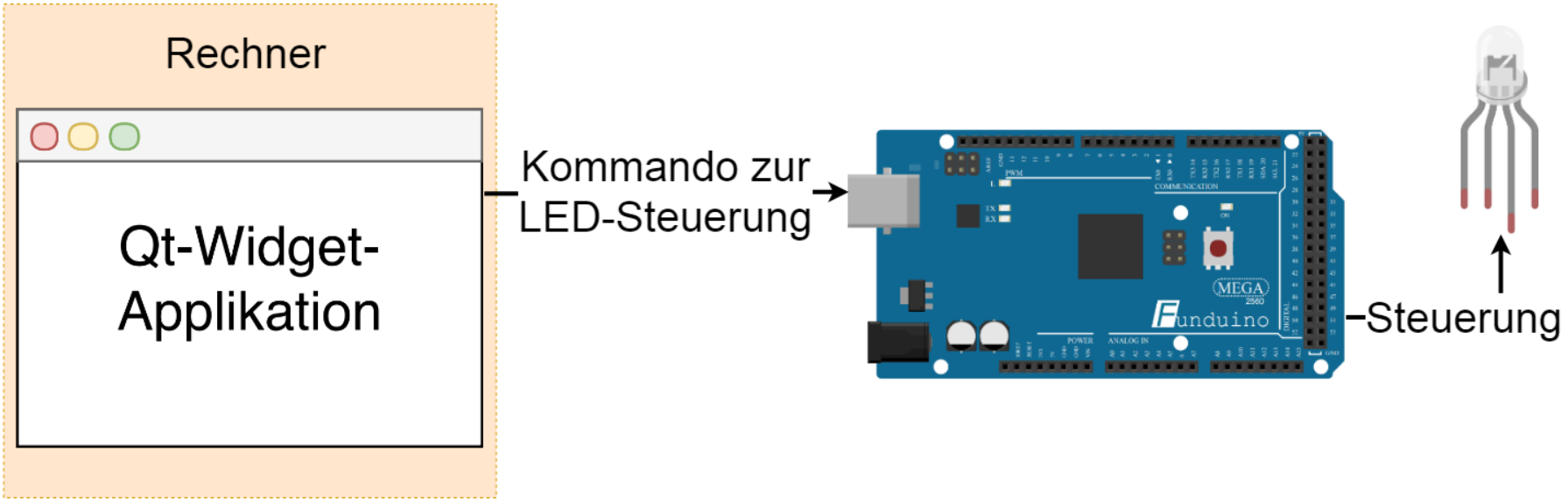
- Ein Bauteil, welches jeweils eine rote, grüne und blaue LED vereint
- Helligkeitssteuerung mittels PWM nach Farbe getrennt möglich
- Beliebige Farben sind durch Farbaddition darstellbar



Übung: GUI zur LED-Farbwahl

Es soll eine GUI programmiert werden, um eine an einen Arduino angeschlossene RGB-LED zu steuern.

Schema



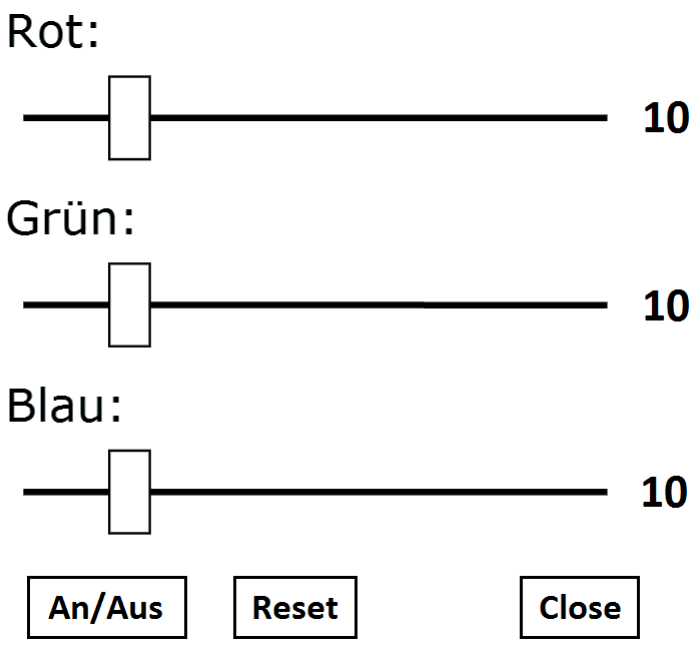
Übung: GUI zur LED-Farbwahl

Es soll eine GUI programmiert werden, um eine an einen Arduino angeschlossene RGB-LED zu steuern.

Anforderungen an die GUI

- Steuerung der drei LED-Farben mittels je eines Schiebereglers
- Wertebereich: 0-100
- Knöpfe, um die LED ein- oder auszuschalten, den Anfangszustand wiederherzustellen und das Programm zu beenden

Skizze



Übung: GUI zur LED-Farbwahl

Es soll eine GUI programmiert werden, um eine an einen Arduino angeschlossene RGB-LED zu steuern.

Vereinbarungen

- Kommunikation zwischen GUI und Arduino erfolgt seriell
- GUI und serielle Kommunikation werden getrennt entwickelt
- Vereinbartes Signal als Schnittstelle:

```
void sendColor(char colorIdentifier, char colorValue);
```

`colorIdentifier` gibt die Farbe an (101: Rot, 102: Grün, 103: Blau)
`colorValue` gibt den Farbwert an (0 - 100)

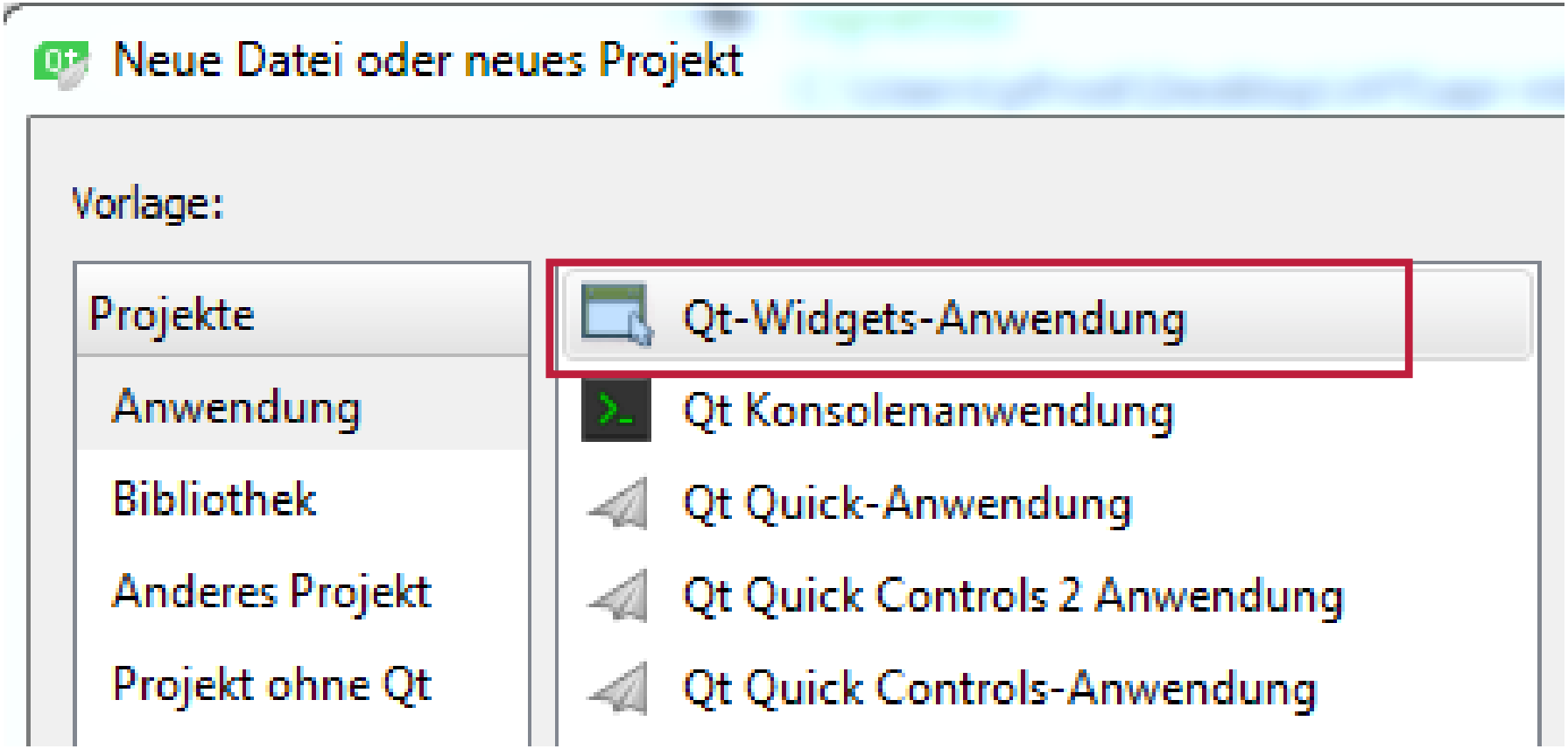
Praxisdemonstration: Qt-Designer
Der Quellcode wird nach der Veranstaltung hochgeladen.

Graphische Benutzeroberfläche erstellen

1. Qt-Projekt erstellen (Qt-Widgets-Anwendung)
2. GUI-Klasse auswählen
3. Layout im Designer erstellen
4. GUI in Quelltext einbinden

Übung

Projekt erstellen

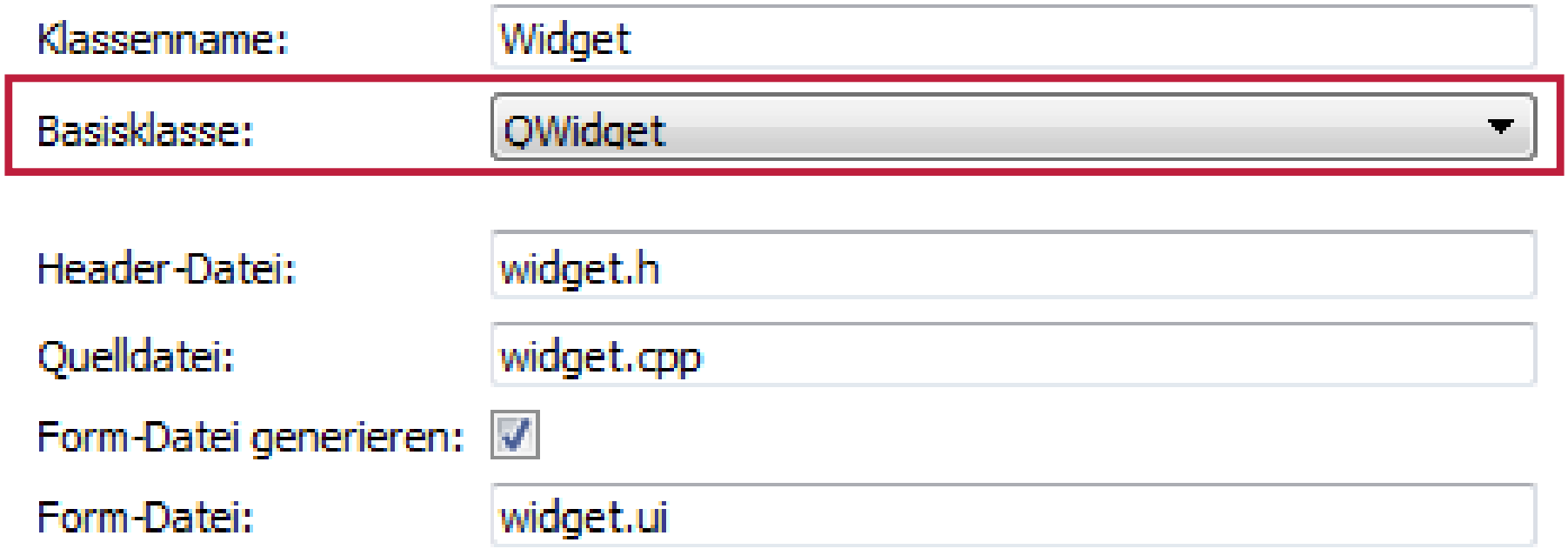


Übung

Klasse erstellen

Parameter der Klasse

Geben Sie Informationen bezüglich der Klassen ein, für die Sie Quelltexte generieren wollen.



Übung

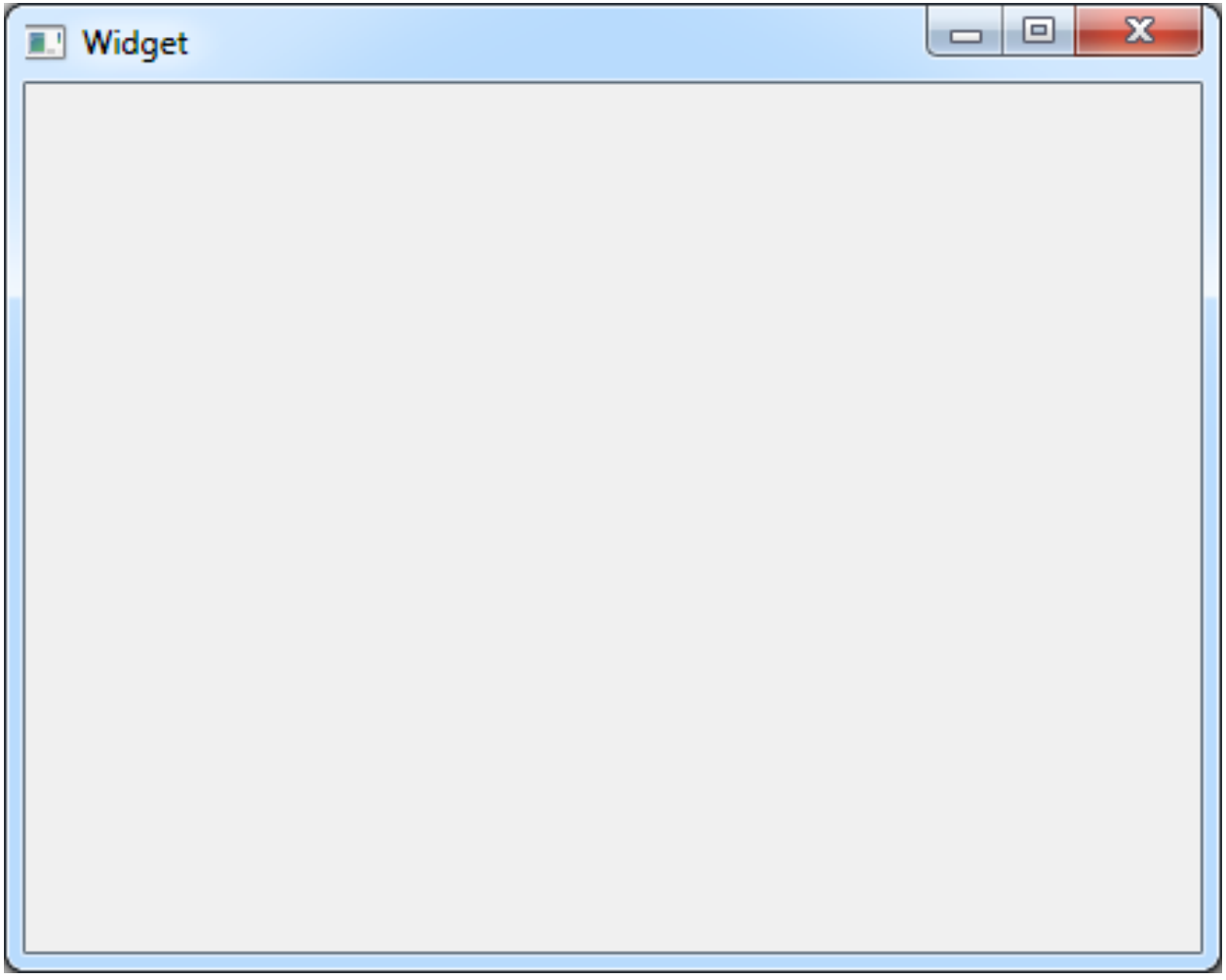
Quelltext

- Quelltext wird automatisch angelegt
- Zugriff auf GUI-Elemente über das Objekt `ui` möglich

```
#include <QWidget>

namespace Ui {
class Widget;
}

class Widget : public QWidget
{
   Q_OBJECT

private:
   Ui::Widget *ui;
```

Übung

Programm starten

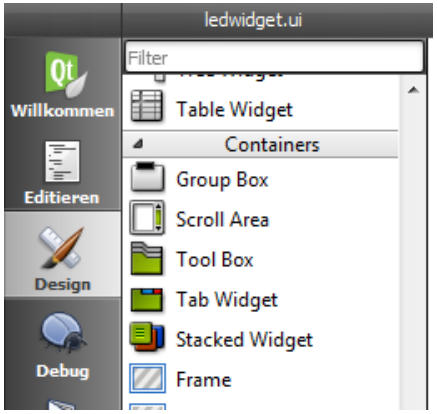


Übung

Benötigte Elemente

Qt Designer

Elemente können über den Qt Designer hinzugefügt werden

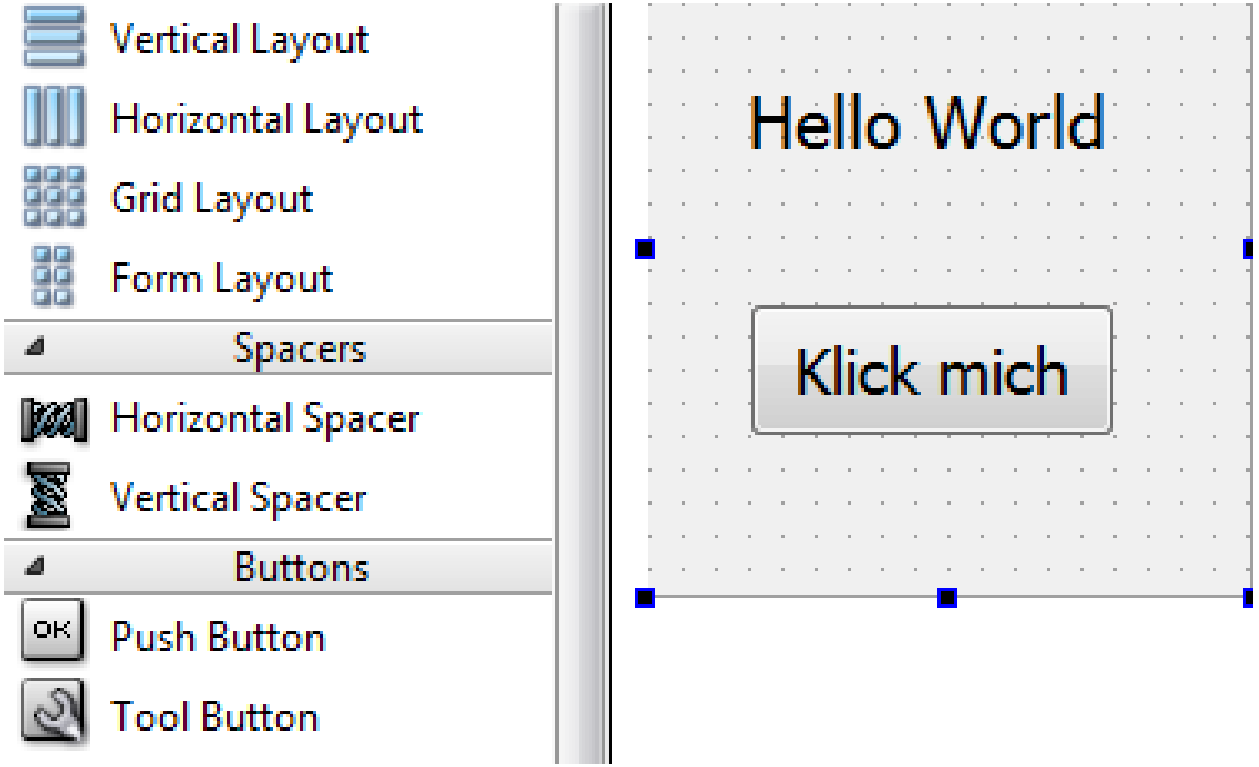


Elemente

`QWidget`	Fensterelement
`QPushButton`	Schaltfläche in Qt
`QLabel`	Element für Beschriftungen
`QSlider`	Schieberegler
`QLineEdit`	Einzeilige Textbox
`QGridLayout`	Rasterlayout
`QHBoxLayout`	Horizontales Layout
`Spacer`	Puffer zum „Auffüllen“ eines Layouts

Übung Layout im Designer erstellen

- Das Zusammenstellen der GUI erfolgt per *Drag and Drop*
- Widgets können auch nächträglich in Layouts einsortiert werden



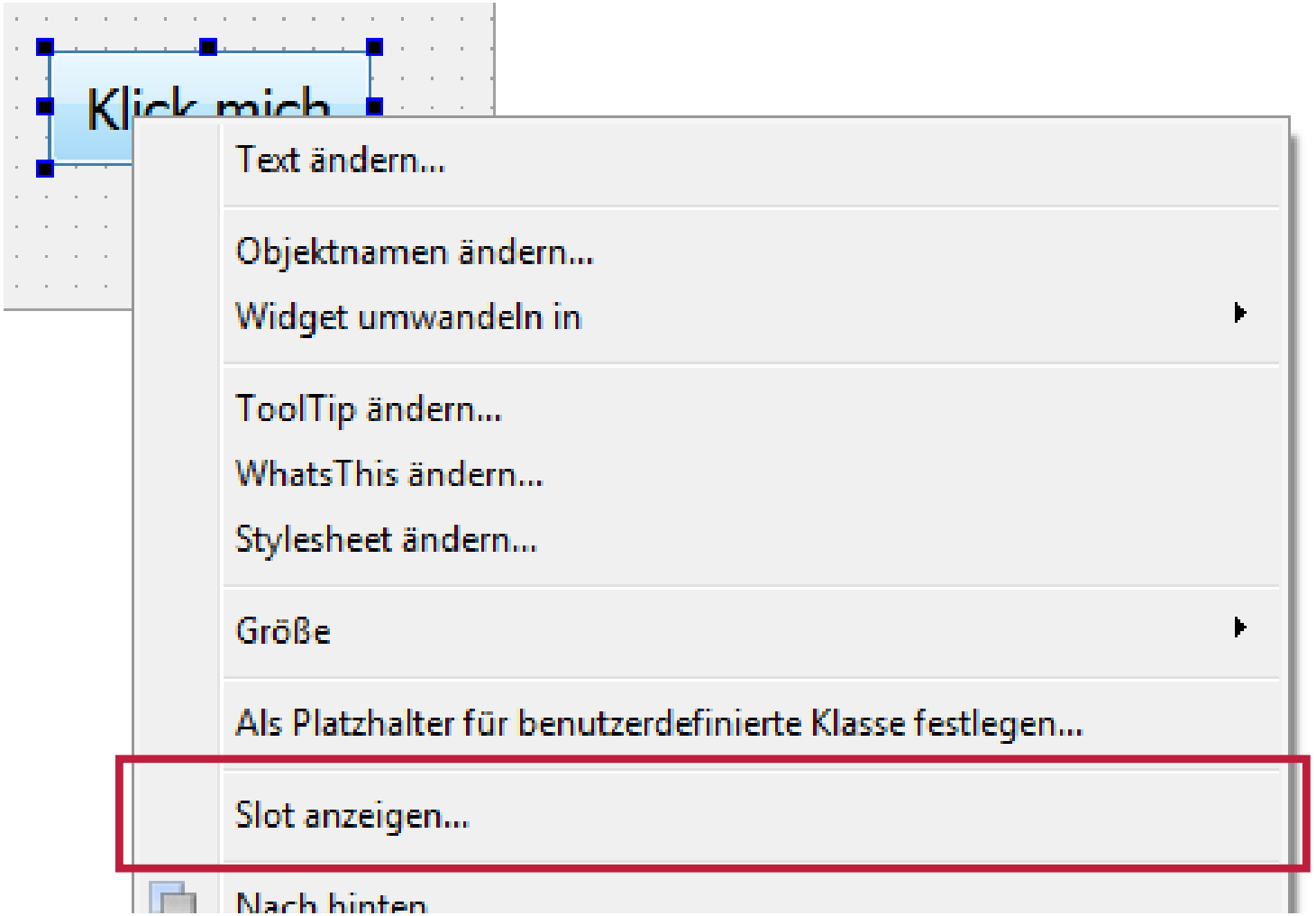
Übung Objektnamen anpassen

- Anordnung der Eigenschaften nach Klassenhierarchie

Eigenschaft	Wert
QObject	
objectName	hwButton
QWidget	
QAbstractButton	
text	Klick mich
icon	
iconSize	16 x 16
shortcut	
checkable	☐

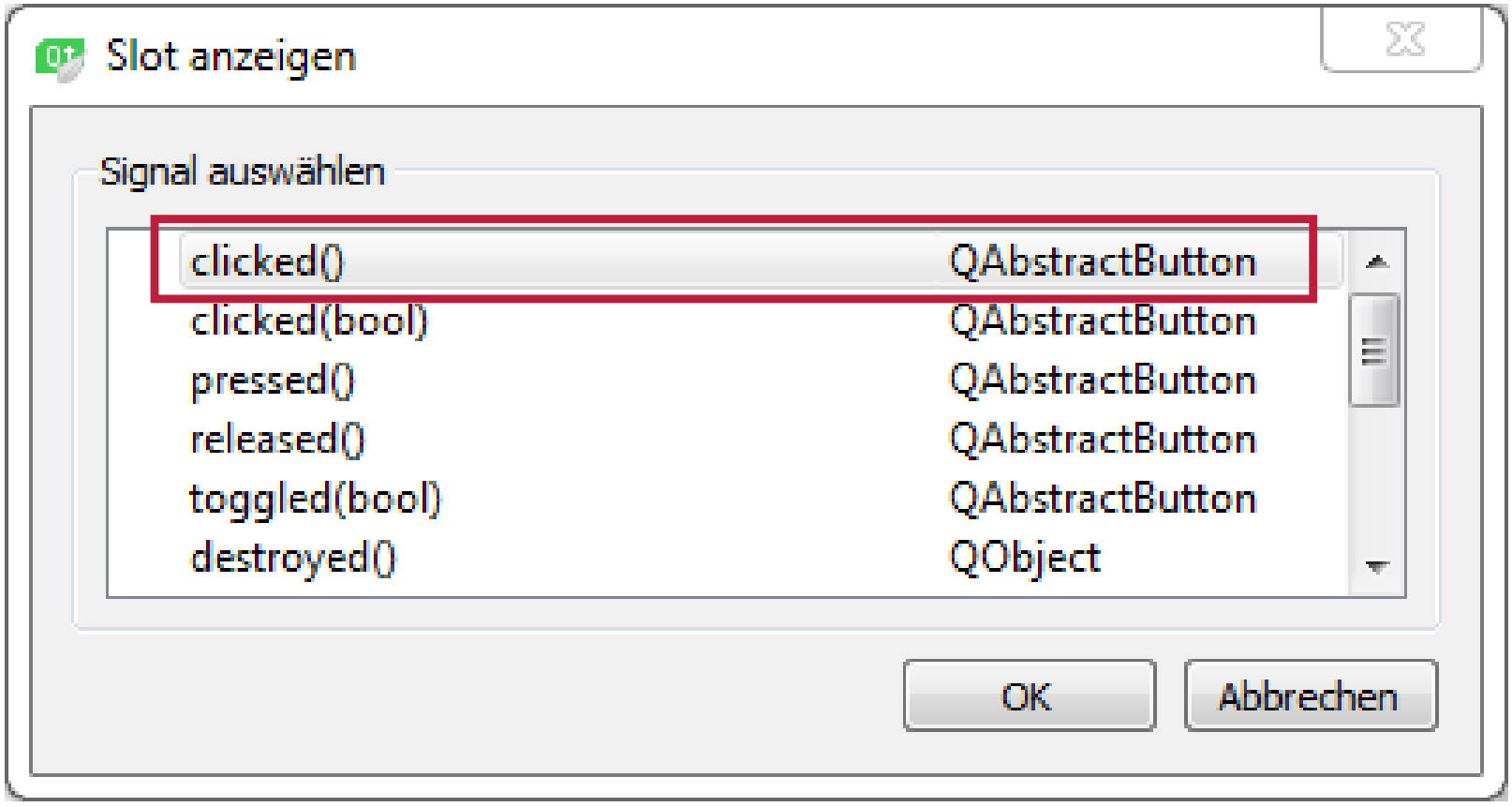
Übung

Slot erstellen



Übung

Slot erstellen



Übung

Quelltext

- Slot wird automatisch über den Dialog „Slot anzeigen“ angelegt
- Über den richtigen Namen wird der Slot automatisch mit dem Signal verbunden

  `on_<ObjektnameDesElements>_clicked()`

```
void Widget::on_hwButton_clicked()
{
   ui->hwLabel->setText("Hallo World");
}
```

Gibt es Fragen oder Anmerkungen zu dem Thema **GUI**?

Abgehakt

GUI-Erstellung mit Qt

Als Teilnehmer soll ich am Ende dieser Übung...

- ☑ die QWidget-Klasse kennen
- ☑ GUIs in Qt erstellen können

Ende

Vielen Dank für eure Aufmerksamkeit!