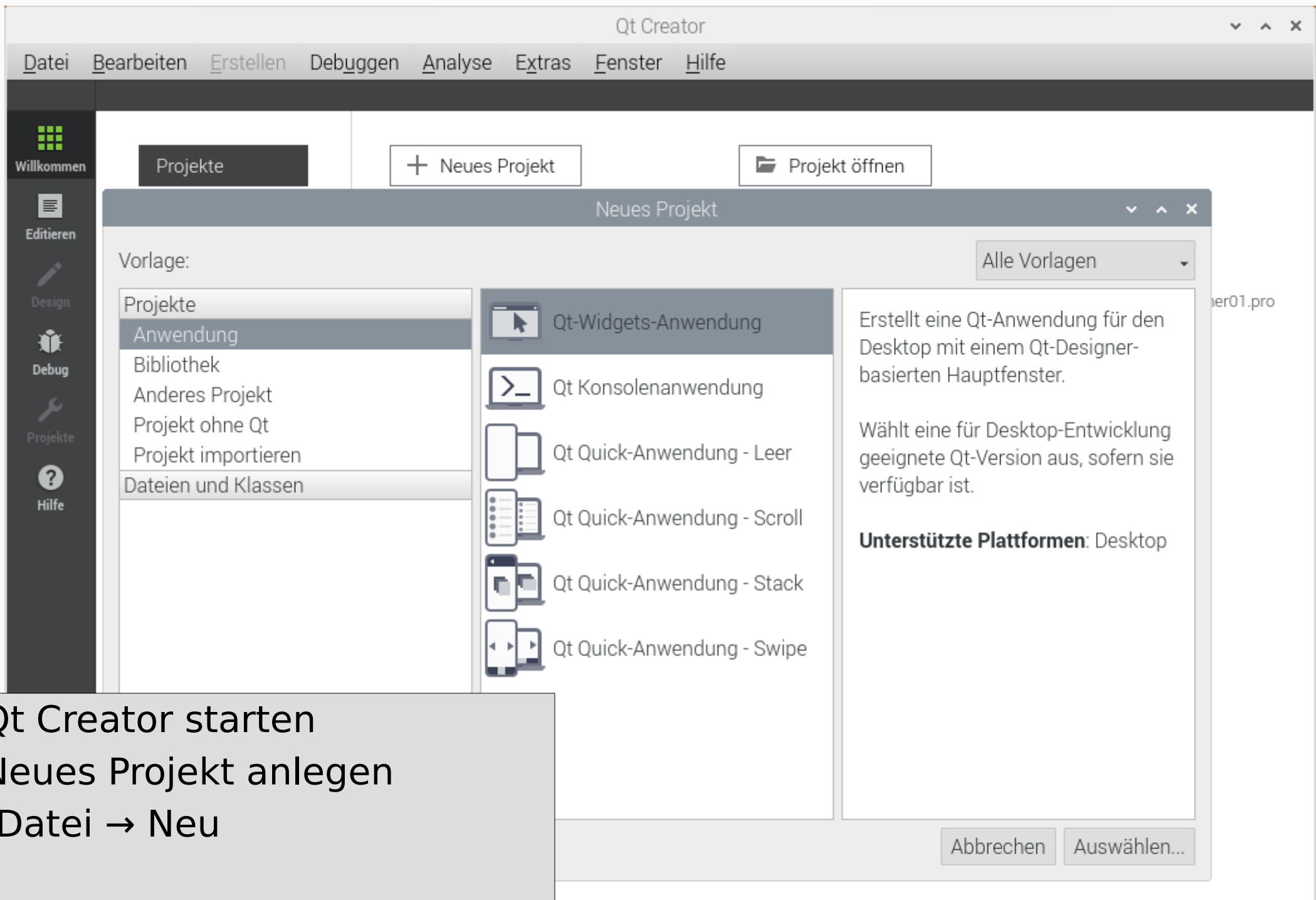
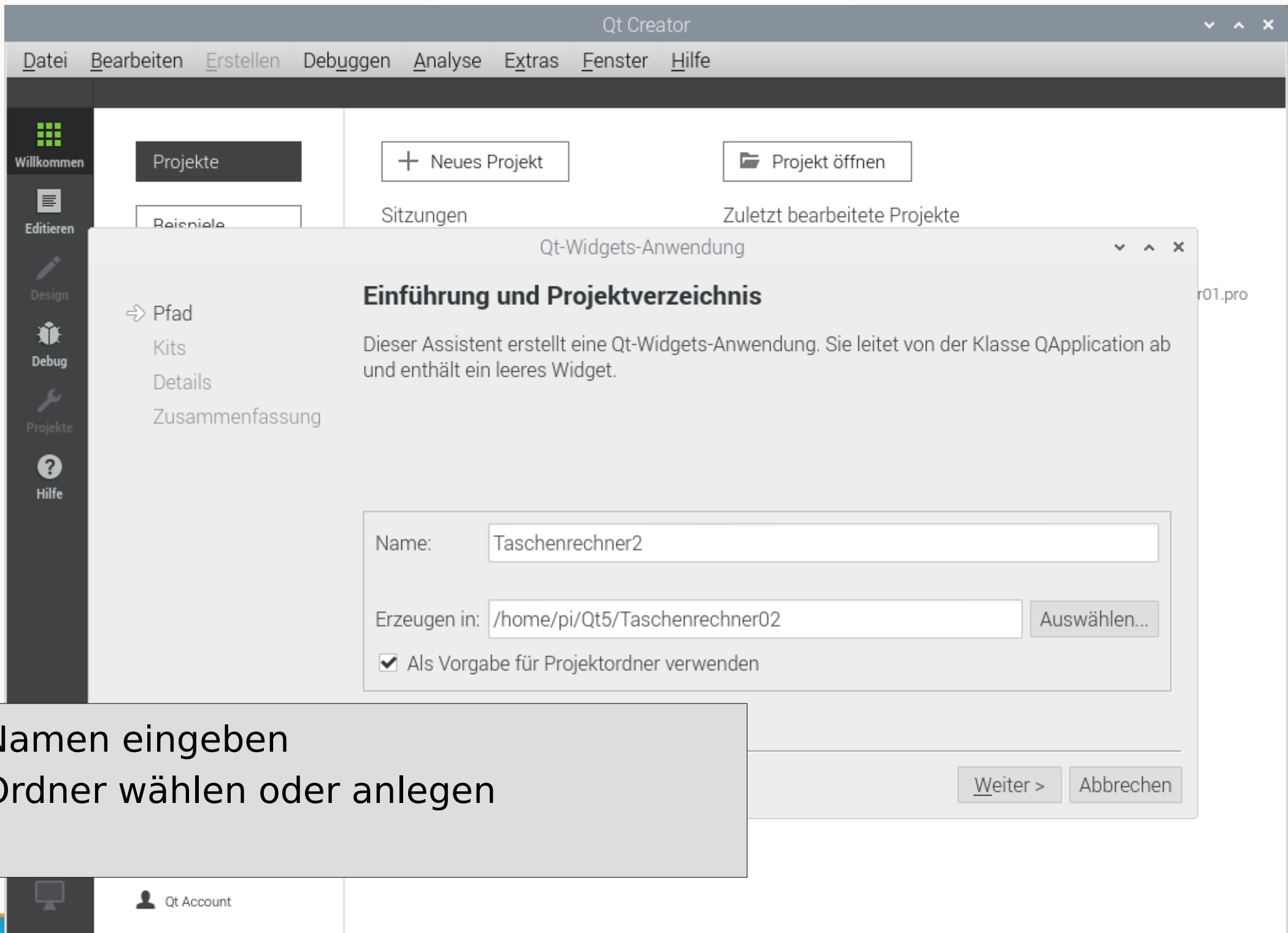


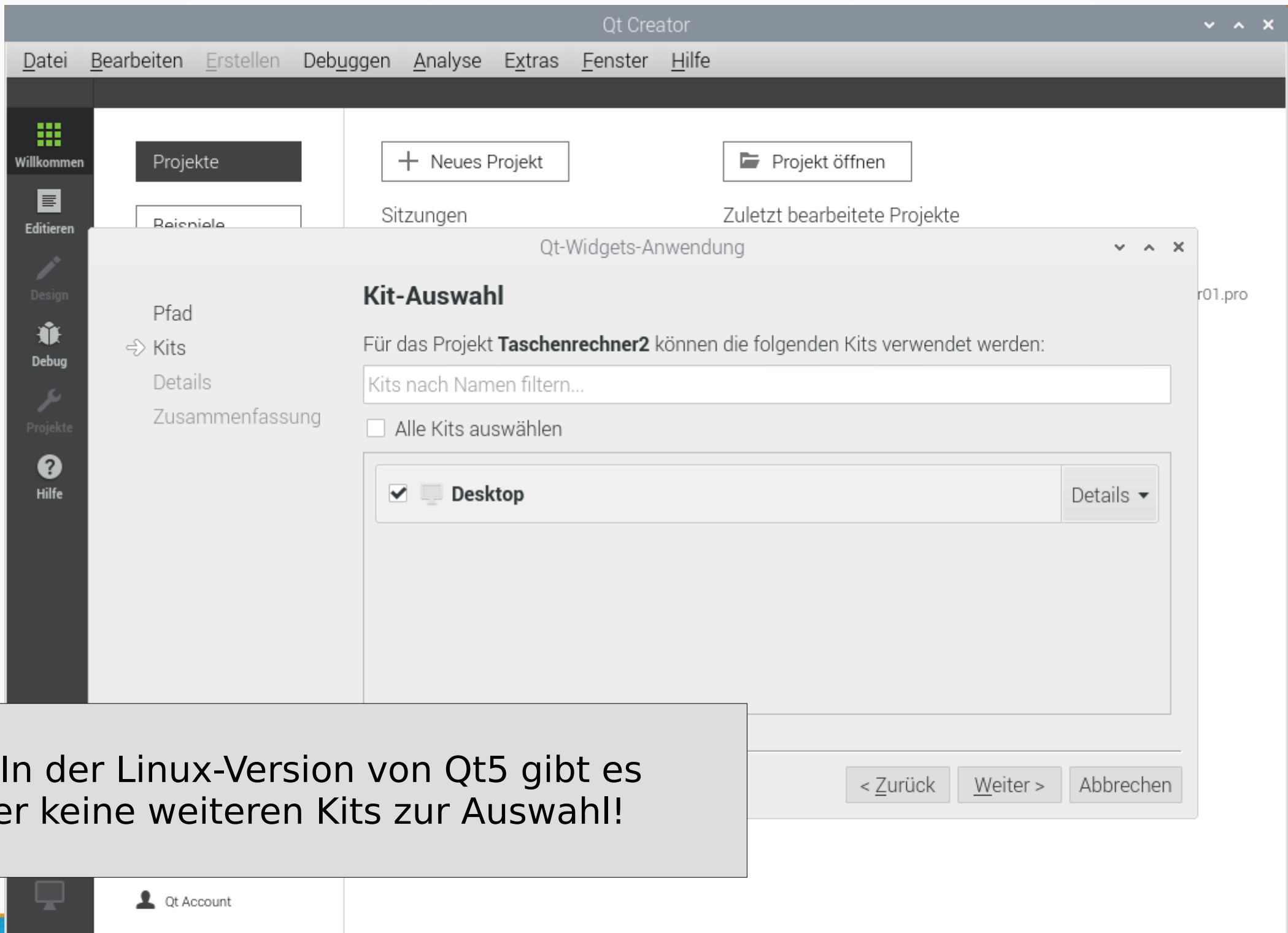
**Ein ganz einfacher Taschenrechner
mit Qt5 auf dem Raspberry Pi 4**



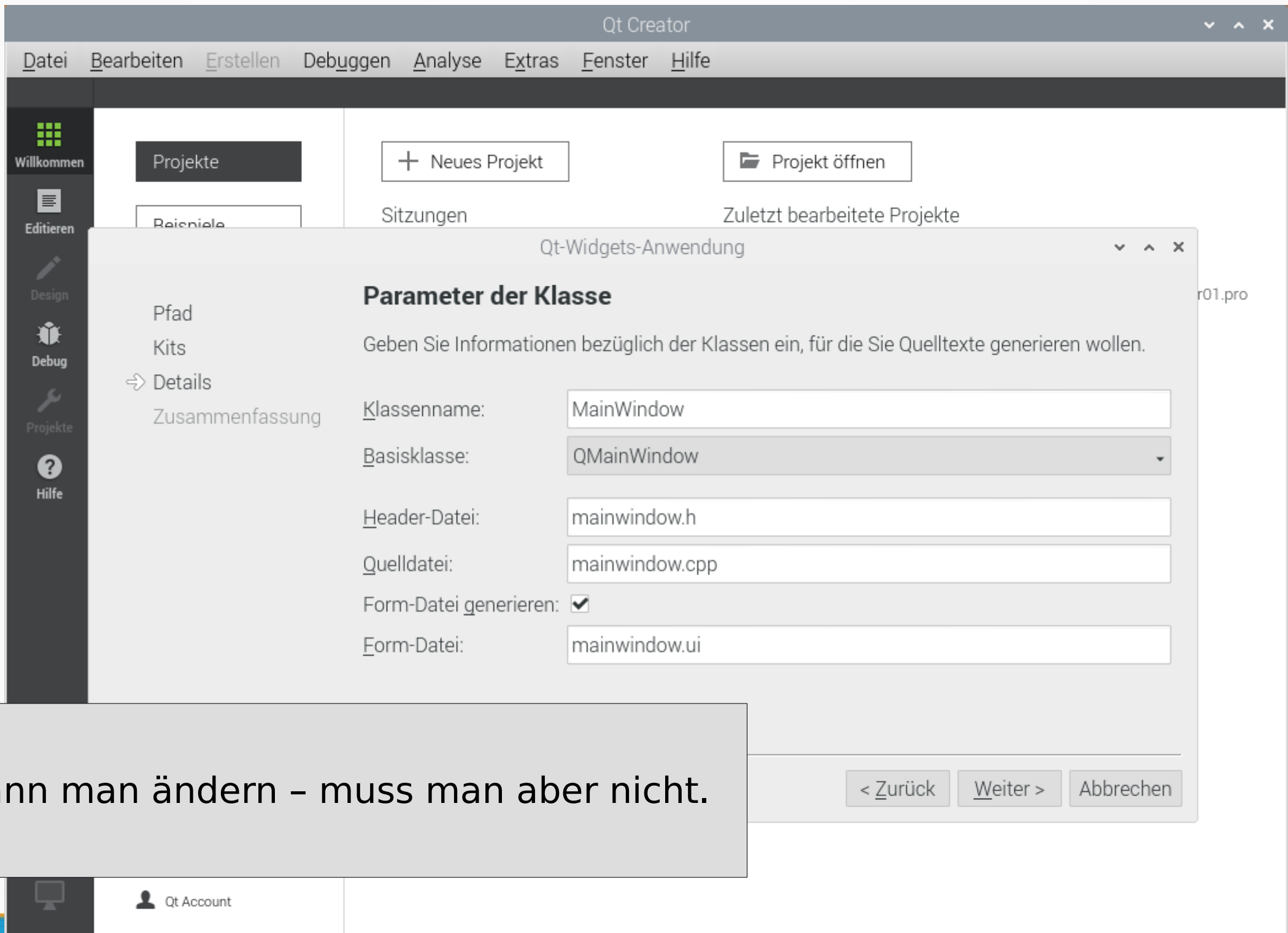
- Qt Creator starten
- Neues Projekt anlegen
→ Datei → Neu



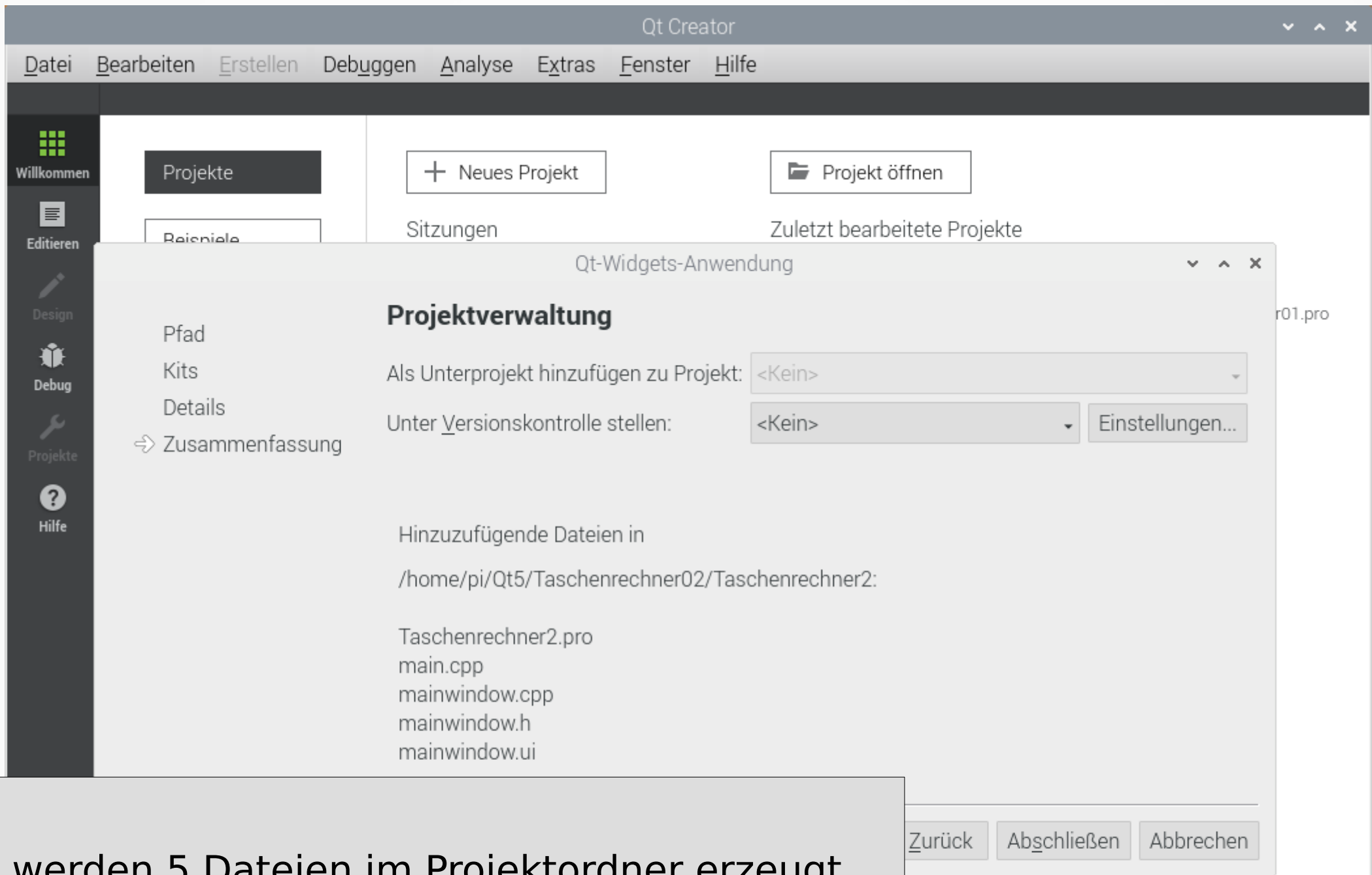
- Namen eingeben
- Ordner wählen oder anlegen



→ In der Linux-Version von Qt5 gibt es hier keine weiteren Kits zur Auswahl!



Kann man ändern - muss man aber nicht.



Es werden 5 Dateien im Projektordner erzeugt.

mainwindow.cpp @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

Projekte **Taschenrechner2**

- Taschenrechner2.pro
- Header-Dateien
 - mainwindow.h
- Quelldateien
 - main.cpp
 - mainwindow.cpp
- Formulardateien
 - mainwindow.ui

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9 }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
```

- mainwindow.ui anklicken
→ Es öffnet sich der GUI-Designer.

mainwindow.ui

Filter

Willkommen

Editieren

Design

Debug

Projekte

Hilfe

Tasch...hner2

Debug

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

- Horizontal Spacer
- Vertical Spacer

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

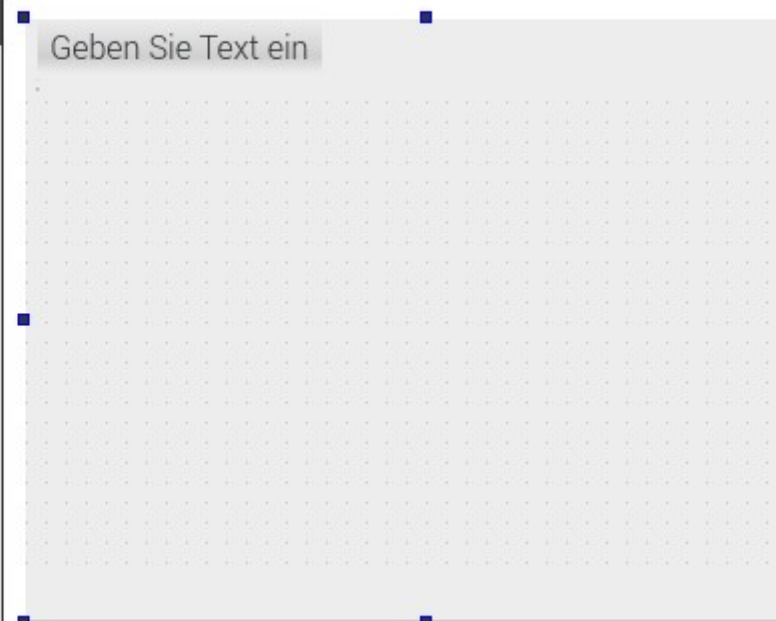
- List View
- Tree View
- Table View
- Column View

Item Widgets (Item-Based)

- List Widget
- Tree Widget
- Table Widget

Containers

- Group Box
- Scroll Area



Filter

Name	Verwendet	Text	Tastenkürzel
------	-----------	------	--------------

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

Filter

MainWindow : QMainWindow

Eigenschaft	Wert
-------------	------

QObject	
objectName	MainWindow
QWidget	
windowModality	NonModal
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 400 x 300]
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0 x 0
maximumSize	16777215 x 16777215

The image shows the Qt Designer interface. On the left is the widget palette with categories: Debug, Projekte, Hilfe, Item Views (Model-Based), Item Widgets (Item-Based), and Containers. The main canvas shows a window titled 'Filter' with a table below it. The Properties Inspector on the right shows the 'MainWindow : QMainWindow' widget with the following properties:

Eigenschaft	Wert
palette	Geerbt
font	A [PibotoLt, 12]
cursor	↔ Pfeil
mouseTracking	<input type="checkbox"/>
tabletTracking	<input type="checkbox"/>
focusPolicy	NoFocus
contextMenuPo...	DefaultContextMenu
acceptDrops	<input type="checkbox"/>
windowTitle	Taschenrechner
windowIcon	
windowOpacity	1.000000
toolTip	

- Im Eigenschafts-Inspektor den **windowTitle** ändern
→ Ich habe für den Fenstertitel „Taschenrechner“ gewählt.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui

Filter

- Layouts
 - Vertical Layout
 - Horizontal Layout
 - Grid Layout
 - Form Layout
- Spacers
 - Horizontal Spacer
 - Vertical Spacer
- Buttons
 - Push Button
 - Tool Button
 - Radio Button
 - Check Box
 - Command Link Button
 - Dialog Button Box
- Item Views (Model-Based)
 - List View
 - Tree View
 - Table View
 - Column View

Geben Sie Text ein

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

QToolBar

Filter

MainWindow : QMainWindow

Eigenschaft	Wert
QObject	
objectName	MainWindow
modal	onModal
geometry	(, 0), 400 x 300]
sizePolicy	preferred, Preferred, 0, 0]
width	x 0
maximumSize	16777215 x 16777215

- Grid Layout wählen und positionieren

mainwindow.ui @ Taschenrechner3 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Willkommen

Editieren

Design

Debug

Projekte

Hilfe

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

- Horizontal Spacer
- Vertical Spacer

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

- List View
- Tree View
- Table View
- Column View

Geben Sie Text ein

Objekt Klasse

- MainWindow QMainWindow
- centralWidget QWidget
- gridLayout QGridLayout**
- menuBar QMenuBar
- mainToolBar QToolBar
- statusBar QStatusBar

Filter

gridLayout : QGridLayout

Eigenschaft Wert

Layout

Eigenschaft	Wert
layoutName	gridLayout
layoutRowStret	0

- Grid Layout wählen und positionieren

mainwindow.ui @ Taschenrechner3 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

- Horizontal Spacer
- Vertical Spacer

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

- List View
- Tree View
- Table View
- Column View

Geben Sie Text ein

Objekt Klasse

- MainWindow QMainWindow
- centralWidget QWidget
- gridLayout QGridLayout**
- menuBar QMenuBar
- mainToolBar QToolBar
- statusBar QStatusBar

Filter

gridLayout : QGridLayout

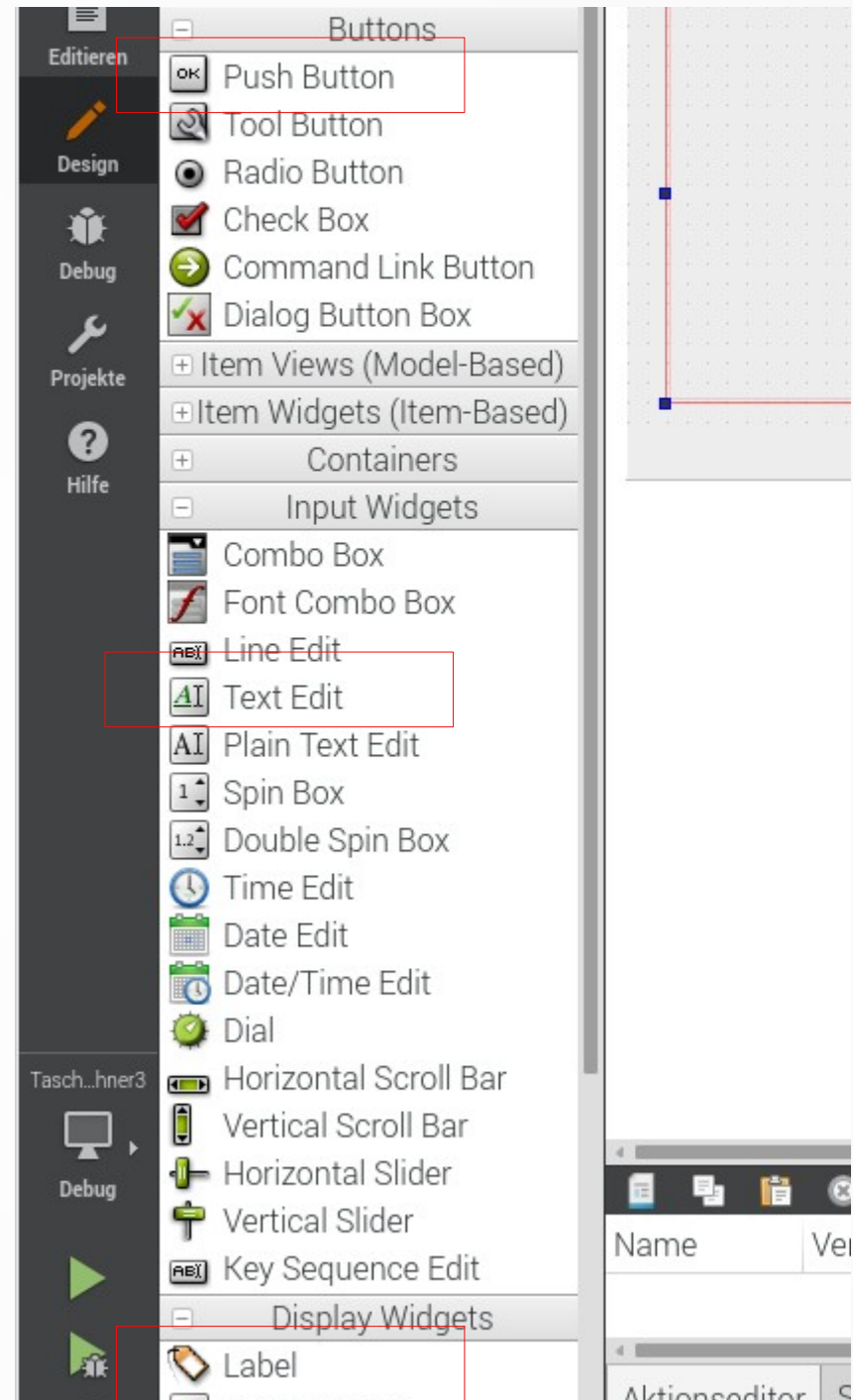
Eigenschaft Wert

Layout

Eigenschaft	Wert
layoutName	gridLayout
layoutRowStret	0

- 3 **Label** und 3 **Text Edit** positionieren
- 2 **Push Button** positionieren

- 3 **Label** und 3 **Text Edit** positionieren
- 2 **Push Button** positionieren



mainwindow.ui @ Taschenrechner3 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Willkommen

Layouts

Spacers

Buttons

Editieren

Push Button

Tool Button

Radio Button

Check Box

Command Link Button

Dialog Button Box

Design

Debug

Projekte

Hilfe

Containers

Input Widgets

Combo Box

Font Combo Box

Line Edit

Text Edit

Plain Text Edit

Spin Box

Double Spin Box

Geben Sie Text ein

TextLabel

TextLabel

TextLabel

PushButton

PushButton

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
gridLayout	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
push...on_2	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit
textEdit_3	QTextEdit
menuBar	QMenuBar
mainToolBar	QToolBar
statusBar	QStatusBar

Filter

pushButton_2 : QPushButton

Eigenschaft	Wert
QObject	
objectName	pushButton_2

- 3 **Label** und 3 **Text Edit** positionieren
- 2 **Push Button** positionieren

- Alles erstellen Strg+Umschalt+B
- Projekt "Taschenrechner2" erstellen Strg+B
- Kompilierungsdatenbank für "Taschenrechner2" erzeugen
- qmake ausführen
- Vollständiges Deployment durchführen
- Deployment des Projekts "Taschenrechner2" durchführen
- Alles neu erstellen
- Projekt "Taschenrechner2" neu erstellen
- Alles bereinigen
- Projekt "Taschenrechner2" bereinigen
- Erstellen abbrechen Alt+Rücktaste
- Ausführen** Strg+R
- Ausführung ohne Deployment
- Kit-Auswahl für Erstellung und Ausführung öffnen...

Klasse	Widget
MainWindow	QMainWindow
CentralWidget	QWidget
grid...ut_2	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit
textEdit_3	QTextEdit
Bar	QMenuBar
oolBar	QToolBar
Bar	QStatusBar

- Item Views (widget-based)
 - List View
 - Tree View
 - Table View
 - Column View
- Item Widgets (Item-Based)
 - List Widget
 - Tree Widget
 - Table Widget

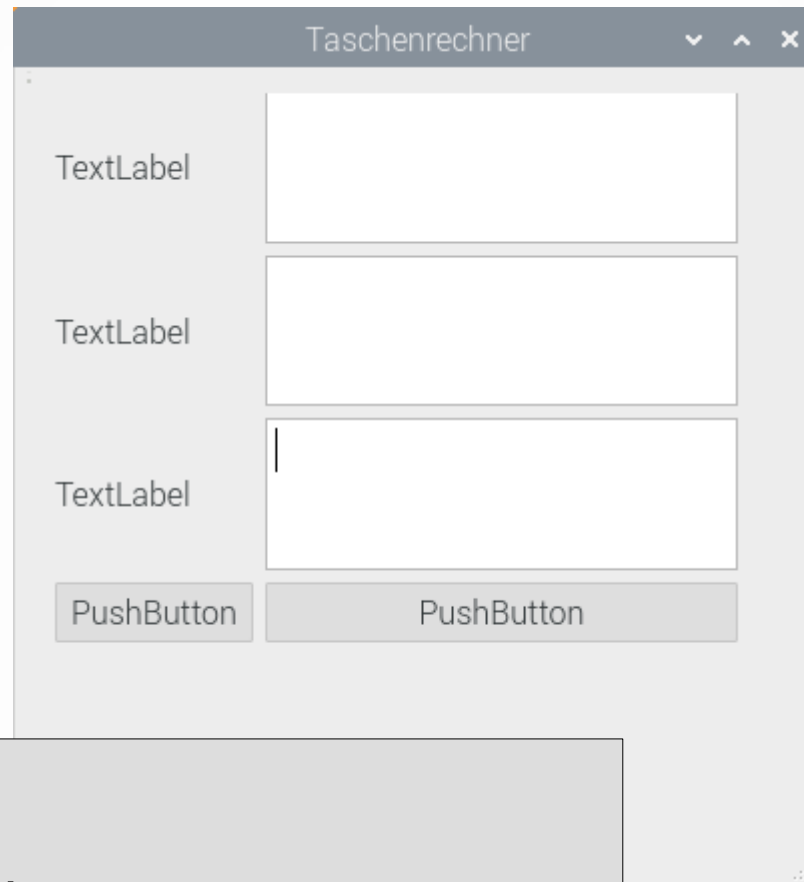
Name	Verwendet	Text	Tastenkürzel

Eigenschaft	Wert
MainWindow : QMainWindow	
= QObject	
objectName	MainWindow
= QWidget	
windowModality	NonModal
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 400 x 273]
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	400 x 273
maximumSize	400 x 273

- Containers
 - Group Box
 - Scroll Area

Name	Verwendet	Text	Tastenkürzel

Eigenschaft	Wert



Das ist schon ganz okay.
Ich möchte aber kleinere Edit-Text-Felder
und die Buttons sollen gleich groß sein.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Objekt

Objekt	Klasse
centralWidget	QWidget
gridLayout	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
push...on_2	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit
textEdit_3	QTextEdit

Filter

textEdit : QTextEdit

Eigenschaft	Wert
geometry	[(106, 6), 200 x 40]
sizePolicy	[Expanding, Expanding, ...]
minimumSize	200 x 40
Breite	200
Höhe	40
maximumSize	200 x 40
Breite	200
Höhe	40
sizeIncrement	0 x 0

Geben Sie Text ein

TextLabel

TextLabel

TextLabel

PushButton

PushButton

Das ist schon ganz okay.
 Ich möchte aber kleinere Edit-Text-Felder
 und die Buttons sollen gleich groß sein.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Objekt Klasse

- centralWidget QWidget
- gridLayout QGridLayout
- label QLabel
- label_2 QLabel
- label_3 QLabel
- push...tton QPushButton
- push...on_2 QPushButton
- textEdit QTextEdit
- textEdit_2 QTextEdit
- textEdit_3 QTextEdit

Filter

textEdit : QTextEdit

Eigenschaft	Wert
geometry	[(106, 6), 200 x 40]
sizePolicy	[Expanding, Expanding, ...]
minimumSize	200 x 40
Breite	200
Höhe	40
maximumSize	200 x 40
Breite	200
Höhe	40
sizeIncrement	0 x 0

Also müssen minimumSize und maximumSize auf feste Größen gesetzt werden.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Objekt

Objekt	Klasse
centralWidget	QWidget
gridLayout	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
push...on_2	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit
textEdit_3	QTextEdit

Filter

textEdit : QTextEdit

Eigenschaft	Wert
geometry	[(106, 6), 200 x 40]
sizePolicy	[Expanding, Expanding, ...]
minimumSize	200 x 40
Breite	200
Höhe	40
maximumSize	200 x 40
Breite	200
Höhe	40
sizeIncrement	0 x 0

Ich stelle für alle 3 Edit-Text-Elemente jeweils 200x40 und für die beiden Push Buttons je 80x30 ein.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

Item Widgets (Item-Based)

Containers

Input Widgets

Geben Sie Text ein

TextLabel

TextLabel

TextLabel

PushButton

PushButton

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
gridLayout	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
push...on_2	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit

Filter

pushButton_2 : QPushButton

Eigenschaft	Wert
enabled	<input checked="" type="checkbox"/>
geometry	[(129, 211), 100 x 30]
sizePolicy	[Minimum, Fixed, 0, 0]
minimumSize	100 x 30
Breite	100
Höhe	30
maximumSize	100 x 30
Breite	100
Höhe	30

Das sieht dann so aus...

Danach lege ich noch auf die gleich Weise die Fenstergröße mit 400x400 fest.

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Willkommen

Editieren

Design

Debug

Projekte

Hilfe

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

Item Widgets (Item-Based)

Containers

Input Widgets

Geben Sie Text ein

TextLabel

TextLabel

TextLabel

PushButton

PushButton

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
gridLayout	QGridLayout
label	QLabel
label_2	QLabel
label_3	QLabel
push...tton	QPushButton
push...on_2	QPushButton
textEdit	QTextEdit
textEdit_2	QTextEdit

Filter

MainWindow : QMainWindow

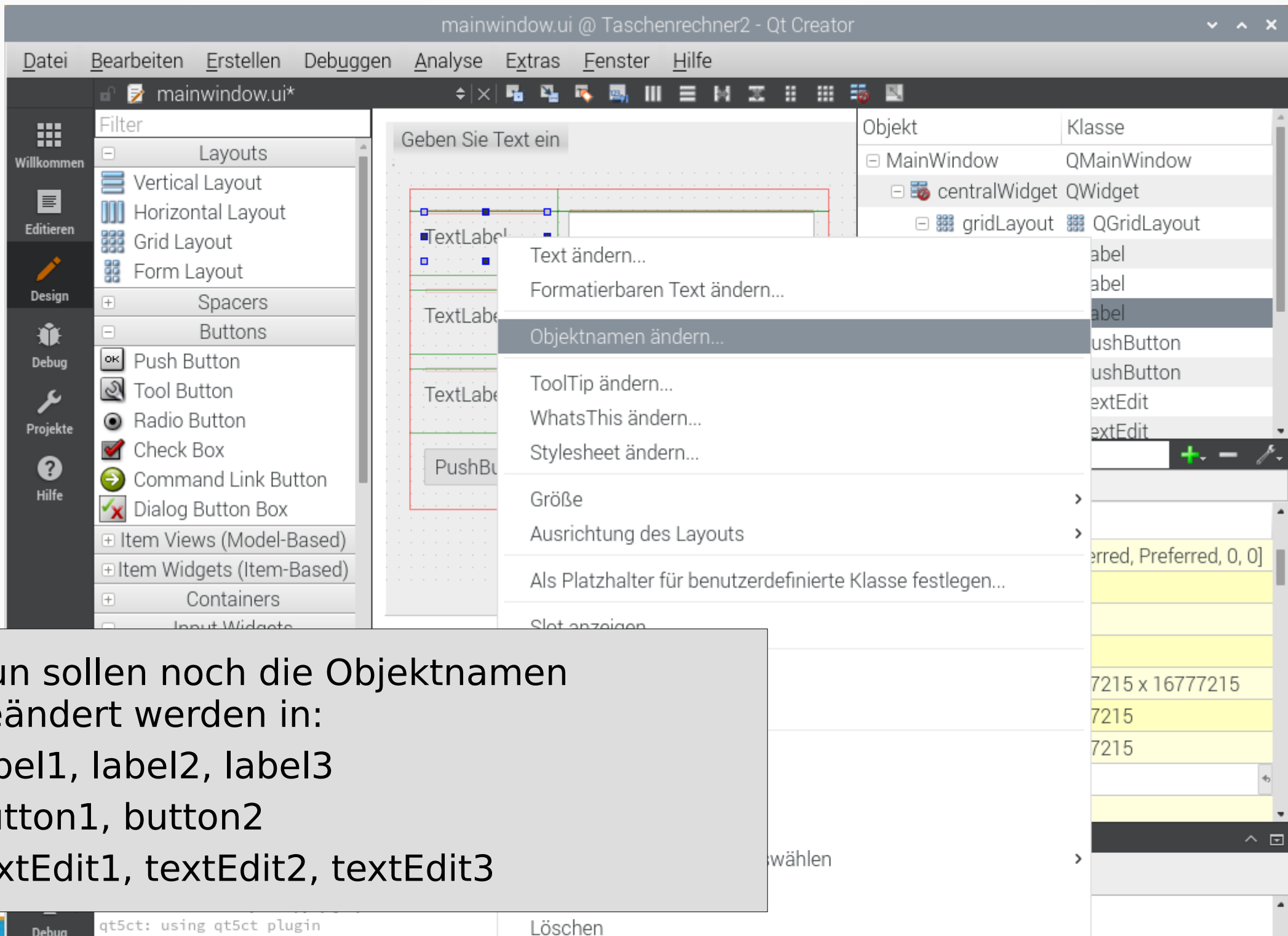
Eigenschaft	Wert
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	400 x 400
Breite	400
Höhe	400
maximumSize	400 x 400
Breite	400
Höhe	400
sizeIncrement	0 x 0
baseSize	0 x 0

Debug/Taschenrechner2...

qt5ct: using qt5ct plugin

Das sieht dann so aus...

Danach lege ich noch auf die gleich Weise die Fenstergröße mit 400x400 fest.



Nun sollen noch die Objektnamen geändert werden in:

- label1, label2, label3
- button1, button2
- textEdit1, textEdit2, textEdit3

mainwindow.ui @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

mainwindow.ui*

Filter

Willkommen

Editieren

Design

Debug

Projekte

Hilfe

Layouts

- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers

Buttons

- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Based)

Item Widgets (Item-Based)

Containers

Input Widgets

Geben Sie Text ein

Summand 1

TextL

TextL

Pus

Text ändern...

Formatierbaren Text ändern...

Objektnamen ändern...

ToolTip ändern...

WhatsThis ändern...

Stylesheet ändern...

Größe

Ausrichtung des Layouts

Objekt	Klasse
gridLayout	QGridLayout
button1	QPushButton
button2	QPushButton
label1	QLabel
label2	QLabel
label3	QLabel
	QTextEdit
	QTextEdit
	QTextEdit
	QMenuBar

wert

preferred, Preferred, 0, 0]

x 0

5777215 x 16777215

5777215

5777215

x 0

x 0

Übergeordnetes Widget auswählen

Die Texte der Label und Buttons werden noch geändert...

The image shows a window titled "Taschenrechner" (Calculator) with a standard title bar containing a dropdown arrow, a maximize icon, and a close icon. The window contains three input fields: "Summand 1", "Summand 2", and "Summe". Below the input fields are two buttons: "Addiere!" (Add!) and "Lösche!" (Delete!).

Nach dem Speichern und **Ausführen** (also neu erstellen bzw. compilieren) sieht es dann so aus...

Nun müssen die **Slots** für die beiden **Push Button** angelegt werden.
In Visual Studio würde man **event handler** sagen.

The screenshot displays the Qt Creator IDE interface. The main window is titled "mainwindow.ui @ Taschenrechner2 - Qt Creator". The menu bar includes "Datei", "Bearbeiten", "Erstellen", "Debuggen", "Analyse", "Extras", "Fenster", and "Hilfe". The toolbar contains various icons for file operations and development tools. The Object Inspector on the right shows a tree view of the UI objects:

Objekt	Klasse
MainWindow	QMainWindow
centralWidget	QWidget
gridLayout	QGridLayout
button1	QPushButton
button2	QPushButton
label1	QLabel
label2	QLabel
label3	QLabel
textEdit1	QTextEdit
textEdit2	QTextEdit

The left sidebar shows a widget palette with the following items:

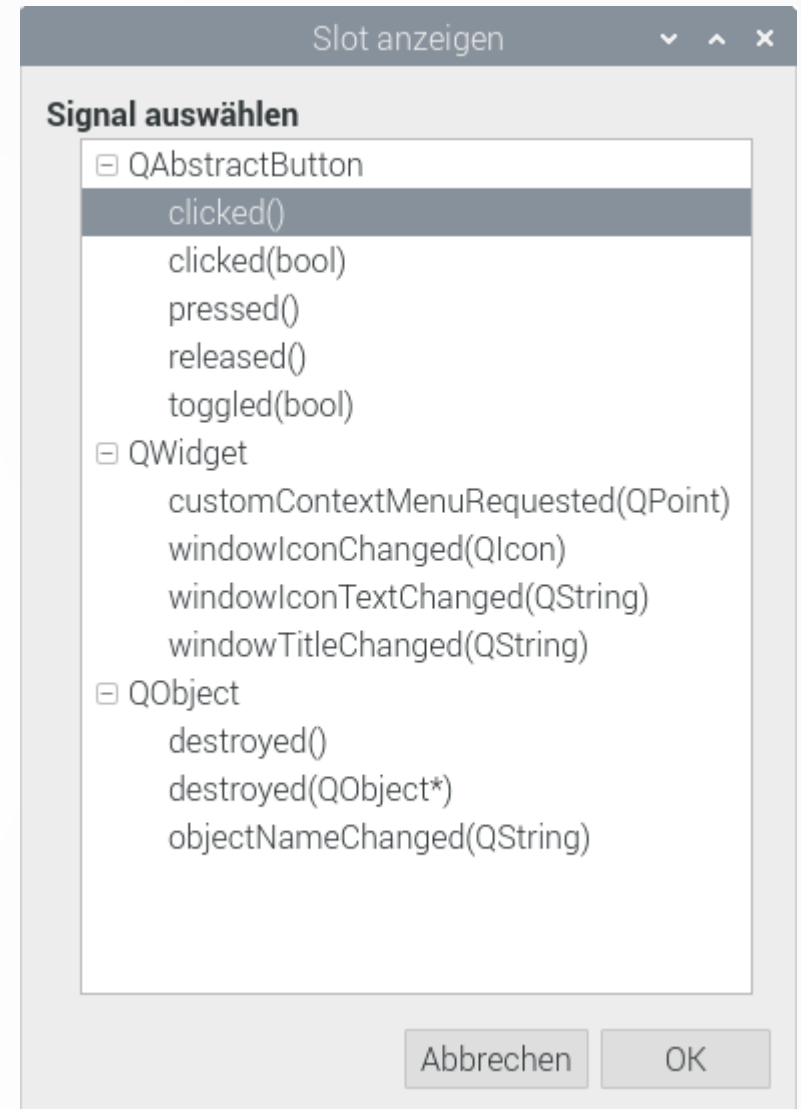
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box
- Item Views (Model-Based)
- Item Widgets (Item-Based)
- Containers
- Input Widgets
- Combo Box
- Font Combo Box
- Line Edit
- Text Edit
- Plain Text Edit
- Spin Box

The context menu is open over a widget, showing the following options:

- Text ändern...
- Objektnamen ändern...
- Widget umwandeln in
- ToolTip ändern...
- WhatsThis ändern...
- Stylesheet ändern...
- Größe
- Ausrichtung des Layouts
- Als Platzhalter für benutzerdefinierte Klasse festlegen...
- Slot anzeigen...
- Nach hinten
- Nach vorn

The bottom status bar shows the application name "Taschenrechner2" and the time "12:27:25". The debug console displays the message "qt5ct: using qt5ct plugin".

Wir wählen **clicked()**.



mainwindow.cpp @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

Projekte mainwindow.cpp* Zeile: 18, Spalte: 5

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9 }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
16 void MainWindow::on_button1_clicked()
17 {
18 }
19
20
```

Hier können wir nun den Code hinein schreiben, der nach dem Button-Click ausgeführt werden soll.

mainwindow.cpp @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

Projekte mainwindow.cpp* Zeile: 23, Spalte: 5

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9 }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
16 void MainWindow::on_button1_clicked()
17 {
18 }
19
20
21 void MainWindow::on_button2_clicked()
22 {
23     |
24 }
25
```

Auch für den zweiten Button wählen wir den Slot + Signal clicked() aus...

mainwindow.cpp @ Taschenrechner2 - Qt Creator

Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe

Projekte mainwindow.cpp* Zeile: 23, Spalte: 5

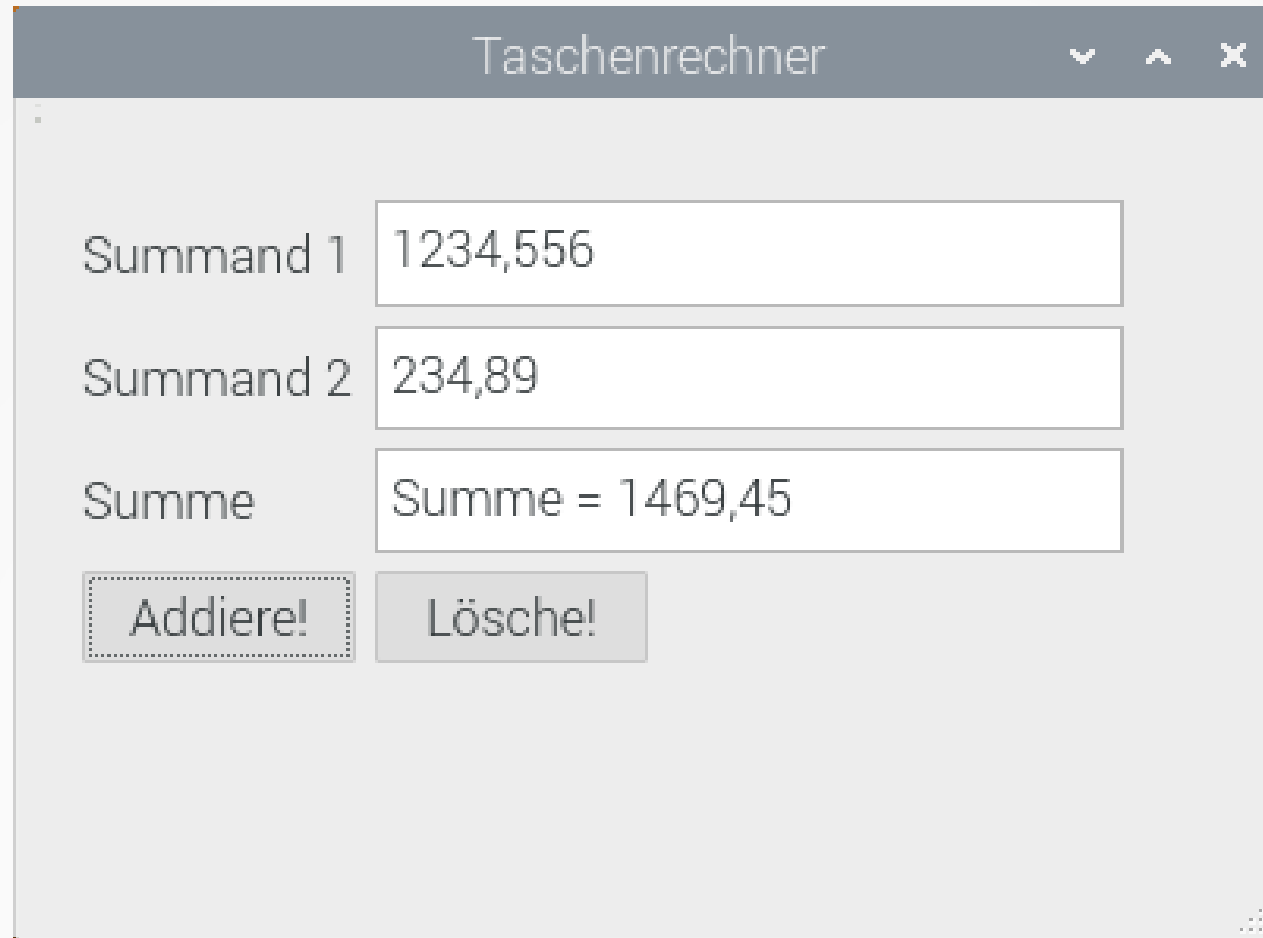
```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9 }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
16 void MainWindow::on_button1_clicked()
17 {
18 }
19
20
21 void MainWindow::on_button2_clicked()
22 {
23     |
24 }
25
```

- Damit ist das GUI-Grundgerüst fertig!
- Jetzt „nur“ noch die Programmierung...

```
mainwindow.cpp @ Taschenrechner2 - Qt Creator
Datei Bearbeiten Erstellen Debuggen Analyse Extras Fenster Hilfe
Projekte mainwindow.cpp Zeile: 27, Spalte: 42
Taschenrechner2
  Taschenrechner2.pro
  Header-Dateien
    mainwindow.h
  Quelldateien
    main.cpp
    mainwindow.cpp
  Formulardateien
    mainwindow.ui

1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9 }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
16 void MainWindow::on_button1_clicked()
17 {
18     // Quelle: https://stackoverflow.com/questions/46254245/get-data-from-qtextedit-line-by-line
19     QString sSummand1 = ui->textEdit1->toPlainText();
20     // Replace Komma / Punkt
21     // Quelle: https://stackoverflow.com/questions/45981060/replace-comma-by-point-in-qlineedit
22     std::replace( sSummand1.begin(), sSummand1.end(), ',', '.');
23     double dSummand1 = (sSummand1.toDouble());
24     QString sSummand2 = ui->textEdit2->toPlainText();
25     std::replace( sSummand2.begin(), sSummand2.end(), ',', '.');
26     double dSummand2 = (sSummand2.toDouble());
27     double dSumme = dSummand1 + dSummand2;
28     // Quelle: https://stackoverflow.com/questions/30055516/converting-and-formatting-double-to-
29     // QString sSumme = QString{ "Summe = %1" }.arg( dSumme, 6, 'g', -1, '0' );
30     QString sSumme = QString{ "Summe = %1" }.arg( dSumme); // vereinfacht
31     std::replace( sSumme.begin(), sSumme.end(), '.', ',');
32     ui->textEdit3 -> setText(sSumme);
33 }
34
35 void MainWindow::on_button2_clicked()
36 {
37     ui->textEdit1 -> setText("");
38     ui->textEdit2 -> setText("");
39     ui->textEdit3 -> setText("");
40 }
41
```

Voila - fertig!



**Ein ganz einfacher Taschenrechner
mit Qt5 auf dem Raspberry Pi 4**

