



Vorlesung „Mensch-Computer-Interaktion“

GRAPHICAL USER INTERFACES

Slots und Events

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



Wintersemester 2020/21

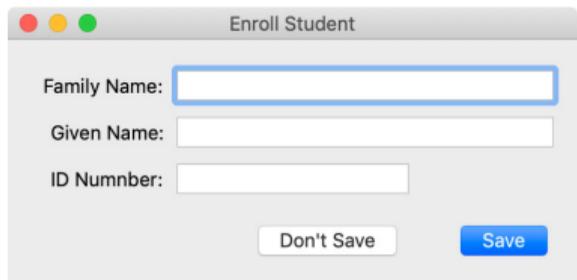


Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Wie kommen die Daten in die Felder?

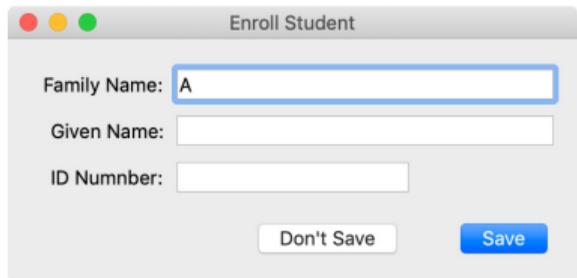
vorher:



The screenshot shows a window titled "Enroll Student" with three input fields: "Family Name:", "Given Name:", and "ID Numnber:". The "Family Name" field is highlighted with a blue border, indicating it is the active field. Below the fields are two buttons: "Don't Save" and "Save".

Ablauf – so oder so ähnlich:

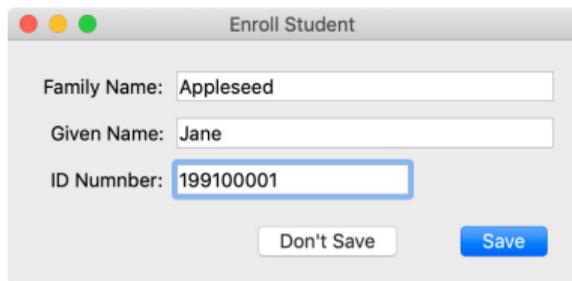
nachher:



The screenshot shows the same "Enroll Student" window. The "Family Name" field now contains the letter "A" and is still highlighted with a blue border. The "Given Name" and "ID Numnber" fields are empty. The "Don't Save" and "Save" buttons remain at the bottom.

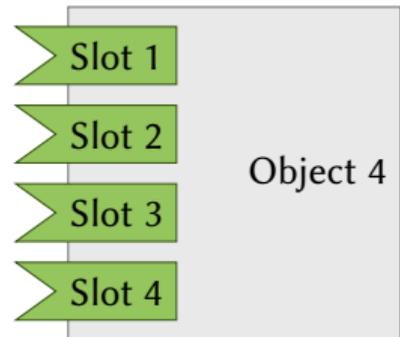
Wie kommen die Daten aus dem Eingabedialog?

Ablauf – so oder so ähnlich:



The image shows a screenshot of a dialog box titled "Enroll Student". It contains three text input fields: "Family Name" with the value "Appleseed", "Given Name" with the value "Jane", and "ID Nummer" with the value "199100001". The "ID Nummer" field is highlighted with a blue border. At the bottom of the dialog, there are two buttons: "Don't Save" and "Save".

Signals und Slots



Object 1:

```
class Object1 : public QObject {  
    Q_OBJECT  
  
signals:  
    void Signal1();  
    void Signal2(int value);  
    void Signal3();  
};
```

Object 2:

```
class Object2 : public QObject {  
    Q_OBJECT  
  
signals:  
    void Signal1();  
    void Signal2(int value);  
};
```

- ▶ Signale müssen den Rückgabetyt void haben.
- ▶ Signale dürfen nicht implementiert werden.
- ▶ Klasse muss direkt/indirekt von QObject ableiten.
- ▶ Q_OBJECT muss im privaten Teil stehen.

Object 3:

```
class Object3 : public QObject {  
    Q_OBJECT  
  
    public slots:  
        void Slot1() {}  
        void Slot2() {}  
};
```

Object 4:

```
class Object4 : public QObject {  
    Q_OBJECT  
  
    public slots:  
        void Slot1() {}  
        void Slot2(int value) {}  
        void Slot3() {}  
        void Slot4() {}  
};
```

- ▶ Slots sind „ganz normale“ Methoden.
- ▶ Eigene Slots müssen implementiert werden.
- ▶ Klasse muss direkt/indirekt von QObject ableiten.
- ▶ Q_OBJECT muss im privaten Teil stehen.

Objekte anlegen, verbinden:

```
Object1 obj1;  
Object3 obj3;  
  
QObject::connect(&obj1, &Object1::Signal1, &obj3, &Object3::Slot1);
```

- ▶ Slot muss identische Parametersignatur wie Signal haben.
- ▶ Slot darf weniger Parameter haben.
- ▶ Compiler prüft Verbindung.
- ▶ Standardmäßig wird Slot bei emittiertem Signal sofort aufgerufen.

Signale:

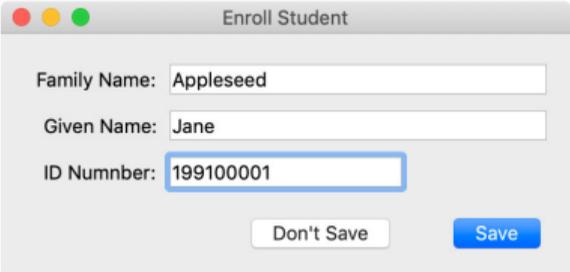
- ▶ Neue Klasse von passender Basisklasse ableiten.
- ▶ Eigene Signale deklarieren.
- ▶ Signale an passenden Stellen emittieren.

Slots:

- ▶ Neue Klasse von passender Basisklasse ableiten.
- ▶ Eigene Slots deklarieren und implementieren.

Object-Klassenhierarchie

Ausschnitt



The image shows a screenshot of a graphical user interface window titled "Enroll Student". The window has a standard macOS-style title bar with red, yellow, and green window control buttons. It contains three text input fields: "Family Name" with the value "Appleseed", "Given Name" with the value "Jane", and "ID Nummer" with the value "199100001". The "ID Nummer" field is highlighted with a blue border. At the bottom of the window, there are two buttons: "Don't Save" and "Save".

Q0bject-Hierarchie

Ausschnitt

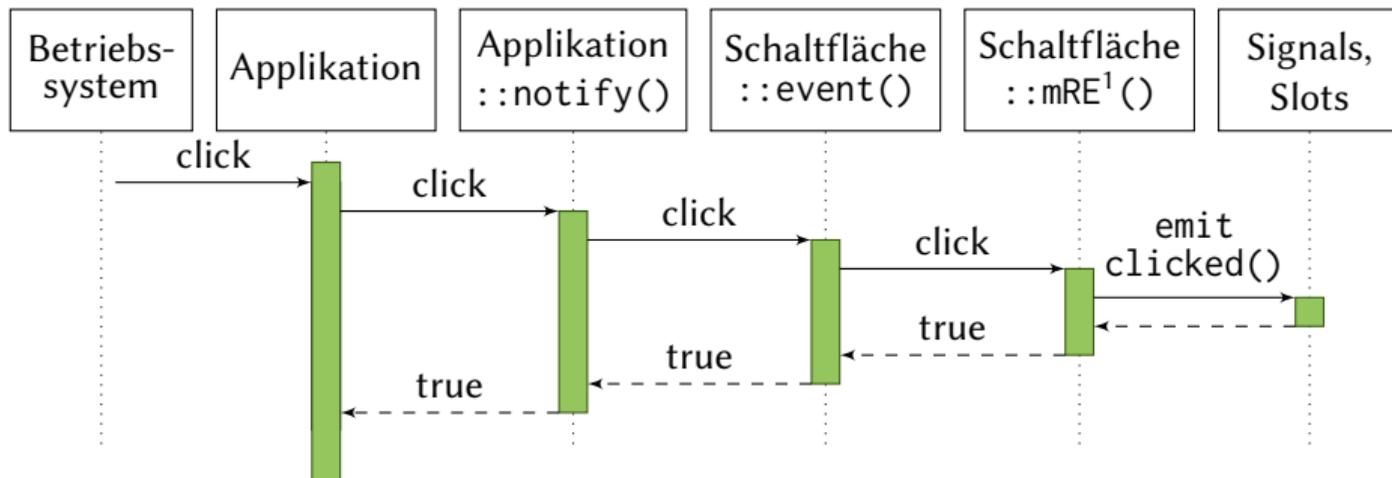


The image shows a screenshot of a macOS-style dialog box titled "Enroll Student". It features three text input fields: "Family Name" with the value "Appleseed", "Given Name" with the value "Jane", and "ID Nummer" with the value "199100001". The "ID Nummer" field is highlighted with a blue border. At the bottom of the dialog, there are two buttons: "Don't Save" and "Save".

Field Label	Value
Family Name:	Appleseed
Given Name:	Jane
ID Nummer:	199100001

Event-Verarbeitung

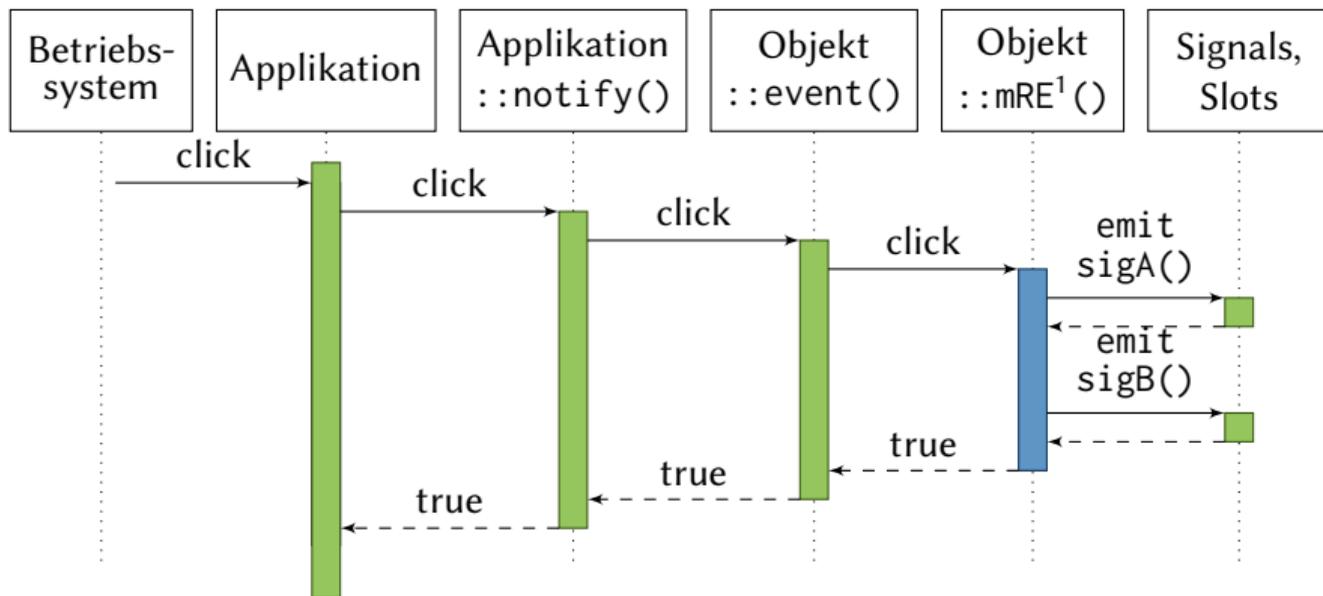
stark vereinfacht



¹ mouseReleaseEvent

Eigene Event-Verarbeitung in eigenem Objekt

stark vereinfacht



¹ mouseReleaseEvent



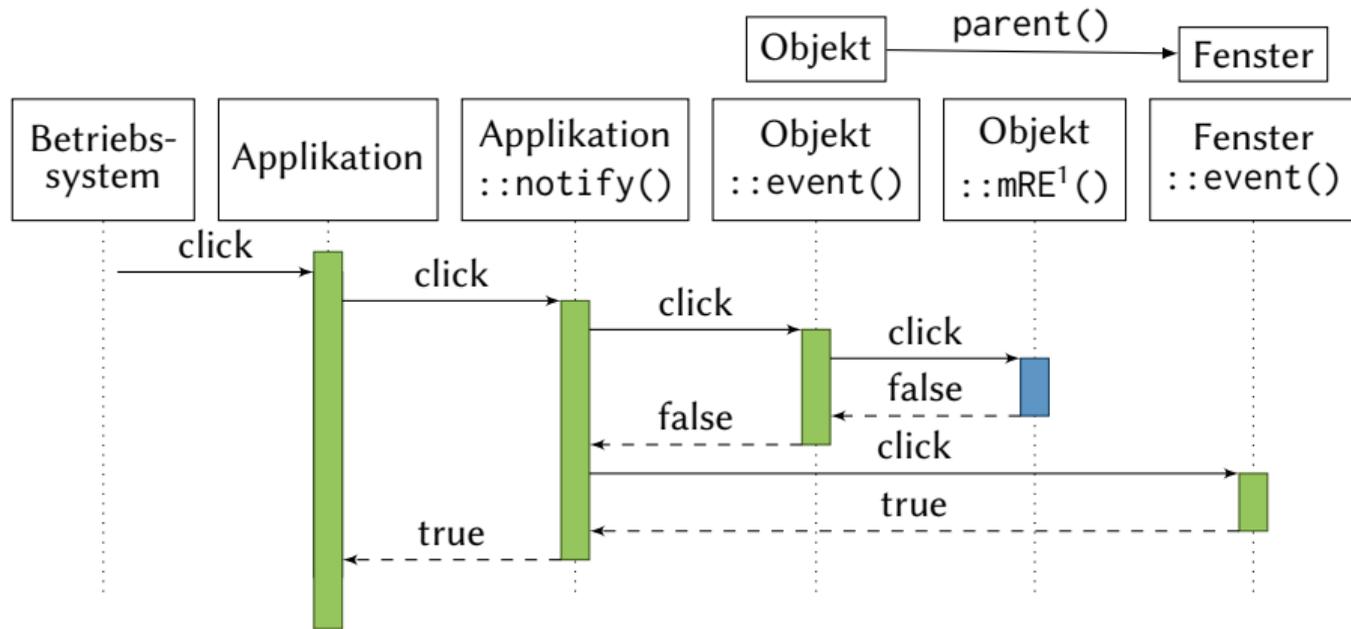
Qt



selbst implementiert

Eigene Event-Verarbeitung in eigenem Objekt

stark vereinfacht



¹ mouseReleaseEvent



Qt



selbst implementiert

- ▶ high-level: beide tun dasselbe

Events:

Signals/Slots:

- ▶ Beispiel: Formulareingabe
- ▶ Signals, Slots
- ▶ Events

Prof. Dr. Tom Vierjahn

▶  tom.vierjahn@w-hs.de

Visual Computing

▶  <https://vc.w-hs.de>

▶  VisualComputingWH

▶  Visual Computing WH

▶  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)