

G. Einsendeaufgabe

Objektorientierte Programmierung mit C++ und Qt

Code:

CPBS12E-XX1-N01

Name: Rein	Vorname: Wjatscheslaw
Postleitzahl und Ort: 14193 Berlin	Straße: Cunostr. 54A
Studien- bzw. Vertrags-Nr.: 800463563	Lehrgangs-Nr.: 246

Fernlehrer/in:

Datum:

Note:

Unterschrift Fernlehrer/in:

Bitte reichen Sie Ihre Lösungen über die Online-Lernplattform ein oder schicken Sie uns diese per Post. Geben Sie bitte immer den Code zum Studienheft an (siehe oben rechts).

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Aufgabe:	3
1.1 Aufgabe:	3
1.2 Lösung:	3
2 Aufgabe:	6
2.1 Aufgabe:	6
2.2 Lösung:	6
3 Aufgabe:	8
3.1 Aufgabe:	8
3.2 Lösung:	8
Listings	10

1 Aufgabe:

Aufgabe:

Erstellen Sie für den Multimedia-Player eine Funktion, die die beiden Symbolleisten und die Wiedergabeliste gleichzeitig aus- beziehungsweise einblendet. Der Aufruf der Funktion soll über einen Eintrag **Vollbild** im Menü **Ansicht** erfolgen. Sorgen Sie dafür, dass der Menüeintrag bei aktivierter Vollbilddarstellung mit einem Häkchen markiert ist. Achten Sie bitte auch darauf, dass die Markierungen bei den anderen Funktionen im Menü **Ansicht** beim Aufruf der Funktion **Vollbild** entsprechend geändert werden. Denken Sie außerdem daran, dass die Markierung für den Eintrag **Vollbild** geändert werden muss, wenn bei aktivierter Vollbilddarstellung zum Beispiel eine Symbolleiste wieder eingeblendet wird.

Ein kleiner Tipp zur Lösung:

Es gibt eine Methode der Klasse `QAction`, mit der Sie eine Aktion selbst auslösen können. Der Name dieser Aktion lautet fast genauso wie das Signal, das wir beim Anklicken eines Menüeintrags oder eines Symbols genutzt haben.

35 Pkt.

Lösung:

Es wird davon ausgegangen, dass ein Tipp zur Lösung nur einen Vorschlag und nicht eine Forderung darstellt, wird in der Aufgabe eine eigene Variante implementiert. Die Funktion, die beide Symbolleisten und die Wiedergabeliste ein- bzw. ausblendet, ist in der Listing (1.4) zu sehen. Dabei wird in der Abhängigkeit von der Methode `isChecked()` eine Auswahl getroffen ob die entsprechende Elemente ein- bzw. ausgeschaltet, die Markierungen geändert werden und die Vollbilddarstellung aktiviert bzw. deaktiviert wird.

Um den sinnvollen Zusammenspiel von den betroffenen Elementen zu gewährleisten, sind in den zugehörigen Anzeige-Funktionen von der Symbolleisten und der Wiedergabeliste die Änderungen (siehe Listings 1.1, 1.2 und 1.3) vorzunehmen. Dafür sind die Methoden `setChecked(0)` um die Markierung von dem Vollbildmodus zu löschen und `showNormal()` für die Umschaltung in normale Bildmodus an den entsprechenden Stellen eingefügt. Die zusätzliche Umschaltung in normale Darstellung ist zwar nicht eindeutig in der Aufgabe gefordert, erscheint aber als eine sinnvolle Funktion.

Außerdem sind die Änderungen im Quellcode sowie in der Listings ausführlich durch Kommentare beschrieben.

```

1      /*****
2      * void MainWindow::on_action_Lautst_rke_triggered()
3      *
4      * Aufgabe 1
5      * Funktion: Die Symbolleiste fuer die Lautstaerke wird ein- bzw.
6      *             ausgeblendet. Zusaetzlich wird bei dem Einschalten die
7      *             Markierung von "Vollbild"-Eintrag geloescht und Vollbildmodus
8      *             ausgeschaltet.
9      *****/
10     void MainWindow::on_action_Lautst_rke_triggered()
11     {
12         //die Symbolleiste Lautstaerke ein- bzw. ausblenden
13         if (ui->action_Lautst_rke->isChecked())
14         {
15             ui->toolBarLautstaerke->show();
16
17             //Aufgabe 1: Markierung fuer den Eintrag Vollbild aendern
18             //und Vollbilddarstellung deaktivieren
19             ui->action_Vollbild->setChecked(0);
20             showNormal();
21         }
22         else
23         {
24             ui->toolBarLautstaerke->hide();
25         }
26     }
27

```

Listing 1.1: Die Symbolleiste Lautstaerke ein- bzw. ausblenden

```

1      /*****
2      * void MainWindow::on_actionW_iedergabe_triggered()
3      *
4      * Aufgabe 1
5      * Funktion: Die Symbolleiste fuer die Wiedergabe wird ein- bzw.
6      *             ausgeblendet. Zusaetzlich wird bei dem Einschalten die
7      *             Markierung von "Vollbild"-Eintrag geloescht und Vollbildmodus
8      *             ausgeschaltet.
9      *****/
10     void MainWindow::on_actionW_iedergabe_triggered()
11     {
12         //die Symbolleiste Wiedergabe ein- bzw. ausblenden
13         if (ui->actionW_iedergabe->isChecked())
14         {
15             ui->toolBarWiedergabe->show();
16
17             //Aufgabe 1: Markierung fuer den Eintrag Vollbild aendern
18             //und Vollbilddarstellung deaktivieren
19             ui->action_Vollbild->setChecked(0);
20             showNormal();
21         }
22         else
23         {
24             ui->toolBarWiedergabe->hide();
25         }
26     }
27

```

Listing 1.2: Die Symbolleiste Wiedergabe ein- bzw. ausblenden

```

1  /*****
2  * void MainWindow::on_action_Wiedergabeliste_triggered()
3  * Aufgabe 1
4  * Funktion: Die Wiedergabeliste wird ein- bzw. ausgeblendet.
5  *           Zusätzlich wird bei dem Einschalten die Markierung von
6  *           "Vollbild"-Eintrag gelöscht und Vollbildmodus ausgeschaltet.
7  *****/
8  void MainWindow::on_action_Wiedergabeliste_triggered()
9  {
10     //die Wiedergabeliste ein- bzw. ausblenden
11     if (ui->action_Wiedergabeliste->isChecked())
12     {
13         ui->dockWidget->show();
14         //Aufgabe 1: Markierung fuer den Eintrag Vollbild aendern
15         //und Vollbilddarstellung deaktivieren
16         ui->action_Vollbild->setChecked(0);
17         showNormal();
18     }
19     else
20     {
21         ui->dockWidget->hide();
22     }
23 }
24

```

Listing 1.3: Die Wiedergabeliste ein- bzw. ausblenden

```

1  /*****
2  * void MainWindow::on_action_Vollbild_triggered()
3  * Aufgabe 1
4  * Funktion: Vollbildmodus ein- bzw. ausschalten. Ereignisse beim
5  *           Aktivieren einer Vollbilddarstellung. Es werden die beiden
6  *           Symbolleisten und die Wiedergabeliste ein- bzw. ausgeblendet.
7  *           Und die entsprechenden Markierungen im Menue angepasst.
8  *****/
9  void MainWindow::on_action_Vollbild_triggered()
10 {
11     if (ui->action_Vollbild->isChecked())
12     {
13         ui->toolBarLautstaerke->hide();
14         ui->dockWidget->hide();
15         ui->toolBarWiedergabe->hide();
16         ui->actionW_iedergabe->setChecked(0);
17         ui->action_Lautst_rke->setChecked(0);
18         ui->action_Wiedergabeliste->setChecked(0);
19         showMaximized();
20     }
21     else
22     {
23         ui->toolBarLautstaerke->show();
24         ui->dockWidget->show();
25         ui->toolBarWiedergabe->show();
26         ui->actionW_iedergabe->setChecked(1);
27         ui->action_Lautst_rke->setChecked(1);
28         ui->action_Wiedergabeliste->setChecked(1);
29         showNormal();
30     }
31 }
32

```

Listing 1.4: Vollbildmodus ein- bzw. ausschalten

2 Aufgabe:

Aufgabe:

Sorgen Sie dafür, dass beim Abschalten des Tons über das Symbol die Anzeige der Lautstärke auf dem Schieberegler und dem Drehfeld auf 0 gesetzt wird. Beim Einschalten setzen Sie wieder den vorherigen Wert ein.

Stellen Sie außerdem sicher, dass das Symbol für den abgeschalteten Ton auch dann angezeigt wird, wenn die Lautstärke über den Schieberegler oder das Drehfeld auf 0 gesetzt wird. Denken Sie auch hier bitte daran, wieder das andere Symbol anzuzeigen, wenn die Lautstärke auf einen Wert ungleich 0 geändert wird.

40 Pkt.

Lösung:

Eine mögliche Variante für die Lösung ist aus dem Listing 2.1 zu entnehmen.

```
1  /*****
2  * void MainWindow::on_actionTon_ein_aus_triggered()
3  * Aufgabe 2
4  * Funktion: Lautstaerke ueber Symbol ein- bzw. ausschalten dabei wird
5  *           die aktuelle Value in iLautTmp gespeichert und bei der
6  *           Wiedereinschalten die Lautstaerke auf gesicherte Wert
7  *           zurueckgesetzt.
8  *****/
9  void MainWindow::on_actionTon_ein_aus_triggered()
10 {
11     //ist die Lautstaerke ausgeschaltet?
12     if (meinPlayer->isMuted() == true)
13     {
14         //dann einschalten
15         meinPlayer->setMuted(false);
16         ui->actionTon_ein_aus->setIcon(QIcon(":/images/mute.png"));
17
18         //Aufgabe 2: Lautstaerke wieder auf gespeicherte Wert setzen
19         lautstaerkeSlider->setValue(iLautTmp);
20     }
21     //sonst ausschalten
22     else
23     {
24         meinPlayer->setMuted(true);
25         ui->actionTon_ein_aus->setIcon(QIcon(":/images/mute_off.png"));
26
27         //Aufgabe 2: Vor dem Ausschalten die aktuelle Lautstaerke speichern
28         iLautTmp = lautstaerkeSlider->value();
29         lautstaerkeSlider->setValue(0);
30     }
31 }
32
```

Listing 2.1: Die Lautstärke über Symbol ein- bzw. ausschalten

Um die Lautstärke über einen Symbol ein- bzw. auszuschalten wird aktuelle Value von dem Schieberegler und somit auch dem Drehfeld (da diese miteinander verknüpft sind) in der Variable iLautTemp (siehe Listing 2.1 Zeile 28) zwischengespeichert und bei dem erneuten Einschalten wird die Lautstärke auf gesicherte Wert zurückgesetzt (2.1 Zeile 19). Dabei wird sichergestellt, dass Schieberegler und Drehfeld sich bei dem Ein- bzw. Ausschalten durch Symbol immer in der entsprechenden Stellung befinden.

Das zweite Teil der Aufgabe lässt sich einfach durch die Auswertung der einzustellenden Lautstärke (siehe Listing 2.2 Zeile 11) realisieren. Dabei dient die Variable neueLautstarke als Auswahlkriterium für die Anzeige des Ton-Symbols.

```

1  /*****
2  * void MainWindow::lautstaerkeAendern(int neueLautstarke)
3  * par[in]: int neueLautstarke
4  * Aufgabe 2
5  * Funktion: der Slot fuer das Veraendern der Lautstaerke
6  *           und entsprechende Anpassung der Symbole
7  *****/
8  void MainWindow::lautstaerkeAendern(int neueLautstarke)
9  {
10     meinPlayer->setVolume(neueLautstarke);
11     if (neueLautstarke == 0)
12     {
13         ui->actionTon_ein_aus->setIcon(QIcon(":/images/mute_off.png"));
14     }
15     else
16     {
17         ui->actionTon_ein_aus->setIcon(QIcon(":/images/mute.png"));
18     }
19 }
20

```

Listing 2.2: Der Slot für das Verändern der Lautstärke

3 Aufgabe:

Aufgabe:

Überarbeiten Sie die Methode zum Einfügen der Einträge in die Navigationsliste so, dass nur noch Dateien eingefügt werden, die nicht bereits in der Liste stehen.

Ein kleiner Tipp zur Lösung:

Die Klasse `QListWidget` kennt eine Methode, um die Einträge in der Liste zu durchsuchen. Sie finden diese Methode in der Qt-Dokumentation der Klasse. Die Methode liefert eine Liste mit den Treffern zurück. Diese Liste können Sie zum Beispiel so erzeugen:

```
QList<QListWidgetItem*> meineListe
```

Die Länge der Liste ermitteln Sie über die Methode `count()` der Liste.

25 Pkt.

Lösung:

Für die Lösung wird wie vorgeschlagen eine Methode der Klasse `QListWidget` (siehe Listing 3.1 Zeilen 11, 18) angewendet, die eine Liste nach bestimmten Kriterien durchsucht und bei der Übereinstimmung die Liste von Treffern zurück liefert. Und wenn danach eine andere Methode Namens `count()` (Listing 3.1 Zeile 23) einen Zählerwert Null zurückgibt, kann man davon ausgehen, dass der Eintrag in der Liste nicht existiert und kann eingefügt werden.


```

1      /*****
2      * void MainWindow::listeUndTitelAktualisieren()
3      *
4      * Aufgabe 3
5      * Funktion: Der Slot fuer das Laden der Datei in die Wiedergabeliste.
6      *           Dabei wird sichergestellt, dass nur die Dateien geladen
7      *           werden die nicht in der Liste vorhanden sind.
8      *****/
9      void MainWindow::listeUndTitelAktualisieren()
10     {
11         QList<QListWidgetItem*> meineListe;
12         //den Namen beschaffen
13         QString titel = meinPlayer->media().request().url().toLocalFile();
14
15         //Ueberpruefen ob zulaedende Datei bereits in der Wiedergabeliste
16         //vorkommt. Eine Such-Methode vergleicht die Eintraege auf
17         //Uebereinstimmung, in unserem Fall volle (MatchExactly).
18         meineListe = ui->listWidget->findItems(titel, Qt::MatchExactly);
19
20         //Wenn der Zaehler der Liste gleich Null ist, wurde keinen Eintrag
21         //gefunden, der den Suchkriterien entspricht. Und die Datei kann
22         //geladen werden.
23         if (meineListe.count() == 0)
24         {
25             //den Namen in das Listefeld schreiben
26             ui->listWidget->addItem(titel);
27             //den Titel setzen
28             this->setWindowTitle("Multimedia-Player " + titel);
29         }
30         else
31         {
32             //Die Datei bereits geladen
33         }
34     }
35

```

Listing 3.1: Der Slot für das Laden der Datei in die Wiedergabeliste

Listings

1.1	Die Symbolleiste Lautstaerke ein- bzw. ausblenden	4
1.2	Die Symbolleiste Wiedergabe ein- bzw. ausblenden	4
1.3	Die Wiedergabeliste ein- bzw. ausblenden	5
1.4	Vollbildmodus ein- bzw. ausschalten	5
2.1	Die Lautstärke über Symbol ein- bzw. ausschalten	6
2.2	Der Slot für das Verändern der Lautstärke	7
3.1	Der Slot für das Laden der Datei in die Wiedergabeliste	9