## C++-Klassenbibliothek Qt

Die ursprünglich von der Fa. Trolltech entwickelte C++-Klassenbibliothek Qt hat das Ziel, einheitliche Schnittstellen für die Oberfächenprogrammierung unter verschiedenen Unix-, Windows- und MacOSvarianten zur Verfügung zu stellen. Sie benutzt dazu Xlib/X11 (Linux,Unix), GDI/Win32 API (Windows) bzw. Carbon/Cocoa (MacOS X). Qt ist auch die Basis der im Linuxbereich weitverbreiten graphischen Benutzeroberfläche KDE. allerdings sind KDE-Programme bis zur Version 3 nicht unter Windows ablauffähig.

Durch die Klassenhierarchie, z.B. sind alle Fensterelementklassen direkt oder indirekt von `QWidget` abgeleitet, stehen viele Komponentenfunktionen in einheitlicher Weise bereit.

Unter dem auf Linux- und Unixsystemen weitverbreiteten Fenstersystem X11 sind die erzeugten Fenster baumartig angeordnet, oberste Fenster (top level widgets) werden üblicherweise mit Titelleiste und Rahmen versehen. Jedes Elternfenster kann mehrere Kindfenster haben, nur die im Elternfenster enthaltenen Teile werden angezeigt. Das Erzeugen eines Fensters bewirkt noch nicht seine Darstellung auf dem Bildschirm, vielmehr ist eine zusätzliche Anweisung notwendig (Realisierung).

Erzeugte und realisierte Bilder sind nur dann sichtbar, wenn sie nicht völlig verdeckt sind. Fensterinhalte verdeckter und wieder sichtbar gewordener Fenster müssen unter X11 vom Fensterprogramm ggf. neu gezeichnet werden. Die Anordnung und Dekoration der Fenster wird von einem Anwendungsprogramm (Window Manager) vorgenommen.

Die Reaktionen, die beispielsweise auf einen Mausklick hin erfolgen sollen, werden durch einen von Qt bereitgestellten Signal/Slot-Mechanismus ermöglicht. Dieser wird durch Spracherweiterungen von C++ realisiert, die vom Präprozessorlauf des "Meta Object Compiler" (moc) behandelt werden.
(Dok.: `http://qt-project.org`)

*Bsp.: Anzeige eines Textes (Qt5-Programm)*

```
#include <QtWidgets>

int main(int argc, char *argv[])
{
  QApplication app(argc,argv);               // Anwendung initialisieren
  app.setFont(QFont("fixed",24));            // Schriftart in Anw. waehlen (24pt)
  QLabel *w = new QLabel("Hello, world!",0); // Textwidget erzeugen
  w->setPalette(QColor("green"));            // Hintergrundfarbe waehlen
  w->setAlignment(Qt::AlignCenter);          // Text zentrieren
  w->resize(320,100);                        // Fenstergroesse festlegen
  w->show();                                 // Widget anzeigen
  return app.exec();                         // Ereigniswarteschlange
}
```

*Profildatei qt5hello.pro:*

```
SOURCES = qt5hello.cpp
QT += widgets
```

*Übersetzen (Qt5 unter Ubuntu Linux 20.04):*

```
# Praeprozessorlauf, Uebersetzen, Starten
qmake qt5hello.pro   # Erzeugt Makefile
make                 # Erzeugt qt5hello (und ggf. qt5hello.moc)
./qt5hello
```

*Bsp.: Uhr als abgeleitete Klasse (Qt5-Programm)*

```
#include <QtWidgets>
#include <ctime>

using namespace std;

class Clock : public QLabel
{
  public:
    Clock(const char* name, QWidget *parent) : QLabel(name, parent) {
      displaytime();
      startTimer(1000);      // ruft alle 1000 Millisek. timerEvent() auf
    }
    virtual ~Clock() {};

  protected:
    void timerEvent(QTimerEvent *event) {
      displaytime();
    }
    virtual void displaytime() {
      time_t t=time(0);
      char s[]="00:00:00";
      strftime(s,strlen(s)+1,"%H:%M:%S",localtime(&t));
      setText(s);
    }
};


int main(int argc, char *argv[])
{
  QApplication app(argc,argv);
  app.setFont(QFont("fixed",24));
  Clock *w = new Clock("Uhr",0);
  w->resize(250,100);
  w->setPalette(QColor("green"));
  w->setAlignment(Qt::AlignCenter);
  w->show();
  return app.exec();
}
```

*Bsp.: Stoppuhr (Qt5-Programm)*

```
#include <QtWidgets>
#include <ctime>
#include <sstream>
#include <string>
#include <iomanip>

using namespace std;

class ClockWidget : public QWidget
{
  Q_OBJECT                                              // notwendig wegen slots
                                                        // (bzw. signals)
public:
  ClockWidget( QWidget *parent=0);                      // Konstruktordekl.
  virtual ~ClockWidget();

public slots:                                           // Spracherweit. (moc)
  void start();                                         // Start-Slot bereitst.
  void stop();                                          // Stop-Slot  bereitst.

protected:
  void timerEvent(QTimerEvent *event);
  virtual void displayruntime();
  QLabel *qlabel;
  QPushButton *button_start,*button_stop;
  QVBoxLayout *qvboxlayout;
  int clockticks;                                       // Hundertstelsekunden
  int timer_id;
};

void ClockWidget::timerEvent(QTimerEvent *event) {
  clockticks++; displayruntime();
}

void ClockWidget::displayruntime() {
  ostringstream os; os.fill('0');
  os << setw(2) << (clockticks/100)/60 << ":"            // Min:Sek:Hdrtsek
     << setw(2) << (clockticks/100)%60 << ":"
     << setw(2) << (clockticks%100) ;
  qlabel->setAlignment(Qt::AlignCenter);
  qlabel->setText(os.str().c_str());
}

ClockWidget::ClockWidget(QWidget *parent) : QWidget(parent)  // Konstruktordef.
{
   clockticks = 0;                                      // Stoppuhr auf 0 stellen
   timer_id   = 0;                                      // Timer nicht gestartet

   qlabel = new QLabel;                                 // Textausgabewidget
   qlabel->setPalette(QColor("white"));                 // Hintergrund weiss
   qlabel->setMinimumHeight(30);                        // Min.hoehe 30 Pixel
```

```
   button_start = new QPushButton("Start");                 // Startknopf
   button_stop =  new QPushButton( "Stop");                 // Stoppknopf
   connect(button_start,SIGNAL(clicked()),this,             // Mausklick auf Startknopf
           SLOT(start()));                                  // mit start-Slot von
                                                            // ClockWidget verknuepfen
   connect(button_stop, SIGNAL(clicked()),this,             // Mausklick auf Stoppknopf
           SLOT(stop()));                                   // mit stop-Slot von
                                                            // ClockWidget verknuepfen

   qvboxlayout = new QVBoxLayout;                           // Layoutcontainer
   qvboxlayout->addWidget(qlabel);                          // Bestandteile zum
   qvboxlayout->addWidget(button_start);                    // Layoutcontainer
   qvboxlayout->addWidget(button_stop);                     // hinzufuegen
   setLayout(qvboxlayout);                                  // Layoutcontainer in
                                                            // Stoppuhrwidget einhaengen
   displayruntime();
};

ClockWidget::~ClockWidget()                                 // Destruktor
{
  delete button_start; delete button_stop;
  delete qlabel; delete qvboxlayout;
};

void ClockWidget::start()                                   // Def. von start
{
  clockticks=0;
  if (timer_id == 0) timer_id = startTimer(10);             // Timer nur starten,
                                                            // falls er nicht laeuft
  displayruntime();
}

void ClockWidget::stop()                                    // Def. von stop
{
  killTimer(timer_id); timer_id = 0;                        // Timer stoppen
  displayruntime();
}

int main( int argc, char **argv )
{
  QApplication app( argc, argv );                           // Initial. der Anw.
  app.setFont(QFont("fixed",24));                           // Font fixed mit 24pt
  ClockWidget *w = new ClockWidget;                         // Widget erzeugen
  w->show();                                                // Widget anzeigen
  return app.exec();                                        // Ereigniswarteschlange
}

#include "qt5stopclock.moc"                                 // Ausgabe von moc anhaengen
```

*Bemerkung:* Bei diesem Programm erzeugt `qmake` durch Aufruf des Metaobjektcompilers `moc` die Includedatei `qt5stopclock.moc`. Diese wird für den Signal/Slot-Mechanismus verwendet. (Die Spracherweiterung `public slots` dient zur Identifikation der Slot-Funktionen im `moc`-Lauf und wird beim Übersetzen per Präprozessoranweisung in `public` umgesetzt.)