

Лекция 5. Обработка событий в наборе библиотек Qt

Разработка многоплатформенного ПО

28 сентября 2016 г.

Классы событий



Рис. 1: часть иерархии классов-событий

События

Нестатические методы QEvent

- **void** accept()
- **void** ignore()
- **bool** isAccepted() **const**
- **bool** spontaneous() **const**
- **QEvent::Type** type() **const**

Статические методы QEvent

- **int** registerEventType(
 int hint = -1)

None	FocusIn	MouseButtonPress
ApplicationStateChange	FocusOut	MouseButtonRelease
Clipboard	KeyPress	MouseMove
Close	KeyRelease	Paint

Таблица 1: некоторые типы событий Qt (**enum** QEvent::Type)

Обработчики событий

Обработчики событий QObject

- **bool** event(QEvent *pEvent)
- **void** customEvent(QEvent *pEvent)
- **void** timerEvent(QTimerEvent *pEvent)

Обработчики событий QWidget

- **void** closeEvent(QCloseEvent *pEvent)
- **void** dragEnterEvent(QDragEnterEvent *pEvent)
- **void** keyPressEvent(QKeyEvent *pEvent)
- **void** mouseMoveEvent(QMouseEvent *pEvent)
- **void** paintEvent(QPaintEvent *pEvent)
- **void** resizeEvent(QResizeEvent *pEvent)

Классы устройств для вывода графики

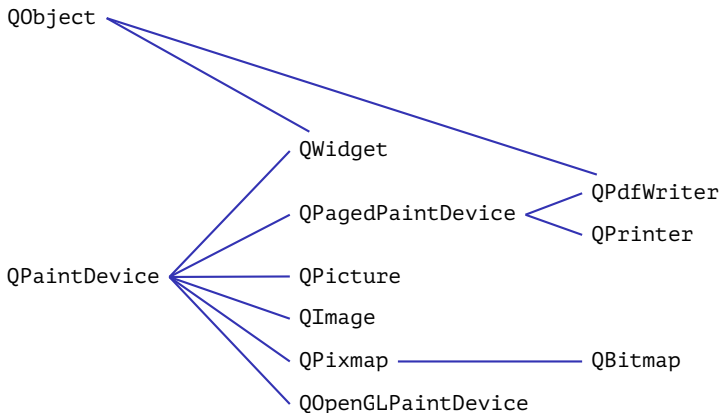


Рис. 2: иерархия классов устройств вывода двумерной графики

Пример

Пример (my-window.cpp)

```
void MyWindow::paintEvent(QPaintEvent *pEvent)
{
    QPainter painter(this);
    // ...
    // painter.draw/* ... */(/* ... */);
    // ...
}
```

Свойства вывода графики

Класс	Свойства
QPen	<ul style="list-style-type: none">• цвет;• толщина;• стиль окончания;• стиль соединения.
QBrush	<ul style="list-style-type: none">• цвет;• стиль;• текстура (опц.);• градиент (опц.)

Класс	Свойства
QFont	<ul style="list-style-type: none">• семейство;• наклон;• жирность;• подчёркивание;• ...

Таблица 2: основные свойства класса QPainter

Методы вывода графики

<code>drawPoint()</code>	<code>drawLine()</code>	<code>drawPolyline()</code>
<code>drawPoints()</code>	<code>drawLines()</code>	<code>drawPolygon()</code>
<code>drawRect()</code>	<code>drawRoundRect()</code>	<code>drawEllipse()</code>
<code>drawArc()</code>	<code>drawChord()</code>	<code>drawPie()</code>
<code>drawText()</code>	<code>drawPixmap()</code>	<code>drawPath()</code>

Таблица 3: основные методы класса QPainter

Пример

Пример (my-window.cpp)

```
void MyWindow::paintEvent(QPaintEvent *pEvent)
{
    QPainter painter(this);
    //
    painter.setRenderHint(QPainter::Antialiasing, true);
    painter.setPen(QPen(Qt::black, 12, Qt::DashDotLine, Qt::RoundCap));
    painter.setBrush(QBrush(Qt::green, Qt::SolidPattern));
    painter.drawEllipse(80, 80, 400, 240);
}
```

Пример

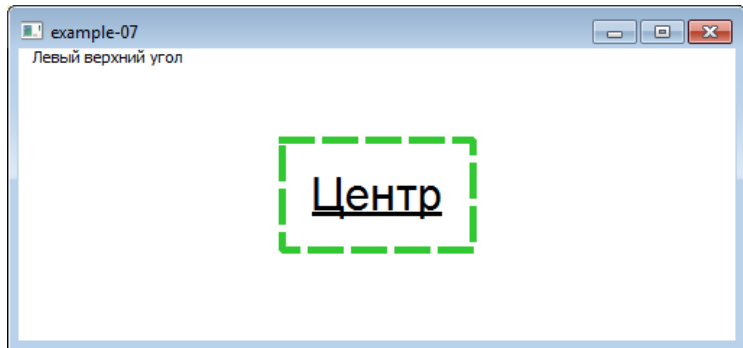


Рис. 3: окно с обработкой событий

Пример

Пример (main-window.h)

```
#ifndef MAIN_WINDOW_H__
#define MAIN_WINDOW_H__

#include <QWidget>

class MainWindow : public QWidget
{
public:
    //
    MainWindow(QWidget * pParent = 0);
    //
};
```

Пример (продолжение)

Пример (main-window.h, окончание)

```
protected:
    //
    void resizeEvent(QResizeEvent *pEvent);
    void paintEvent(QPaintEvent *pEvent);
    //
private:
    //
    QPixmap m_Pixmap;
};    // class MainWindow

#endif    // MAIN_WINDOW_H_
```

Пример (продолжение)

Пример (main-window.cpp)

```
#include "main-window.h"

#include <QtWidgets>

MainWindow::MainWindow(QWidget *pParent)
    : QWidget(pParent)
{
    setAttribute(Qt::WA_NoSystemBackground, true);
}
```

Пример (продолжение)

Пример (main-window.cpp, продолжение)

```
void MainWindow::resizeEvent(QResizeEvent *pEvent)
{
    m_Pixmap = QPixmap(pEvent->size());
    m_Pixmap.fill(Qt::white);
    //
    QPainter painter(&m_Pixmap);
    //
    const QString cstrL = QString::fromLocal8Bit(
        "Левый верхний угол");
    painter.drawText(10, 10, cstrL);
    //
}
```

Пример (продолжение)

Пример (main-window.cpp, продолжение)

```
QFont font("Arial", 24);  
font.setUnderline(true);  
//  
const QString cstrC = QString::fromLocal8Bit("Центр");  
const int cnFlags = Qt::AlignCenter | Qt::TextSingleLine;  
QRect rectText;  
painter.setFont(font);  
painter.drawText(m_Pixmap.rect(), cnFlags, cstrC, &rectText);  
rectText.adjust(-20, -20, 20, 20);  
QPen pen(QColor(50, 200, 50));
```

Пример (продолжение)

Пример (main-window.cpp, окончание)

```
pen.setStyle(Qt::DashLine);
pen.setWidth(5);
painter.setPen(pen);
painter.drawRect(rectText);
} // MainWindow::resizeEvent()

void MainWindow::paintEvent(QPaintEvent *pEvent)
{
    QPainter painter(this);
    painter.drawPixmap(0, 0, m_Pixmap);
}
```


Пример (окончание)

Пример (example-07.cpp)

```
#include "main-window.h"

#include <QApplication>

int main(int nArgC, char *apszArgV[])
{
    QApplication app(nArgC, apszArgV);
    MainWindow *pWindow = new MainWindow;
    pWindow->show();
    //
    return app.exec();
}
```

Пример

Пример (key-window.h)

```
#ifndef KEY_WINDOW_H__
#define KEY_WINDOW_H__

#include <QWidget>

class KeyWindow : public QWidget
{
public:
    //
    KeyWindow(QWidget *pParent = 0);
    //
};
```

Пример (продолжение)

Пример (key-window.h, окончание)

```
protected:
    //
    void keyPressEvent(QKeyEvent *pEvent);
    void keyReleaseEvent(QKeyEvent *pEvent);
    void paintEvent(QPaintEvent *pEvent);
    //
private:
    //
    bool m_bKeyPressed;
};    // class KeyWindow

#endif    // KEY_WINDOW_H_
```

Пример (продолжение)

Пример (key-window.cpp)

```
#include "key-window.h"

#include <QtWidgets>

KeyWindow::KeyWindow(QWidget *pParent)
    : QWidget(pParent),
      //
      m_bKeyPressed(false)
{
    // setAutoFillBackground(false);    // по умолчанию
}
```

Пример (продолжение)

Пример (key-window.cpp, продолжение)

```
void KeyWindow::keyPressEvent(QKeyEvent *pEvent)
{
    m_bKeyPressed = true;
    repaint();
}

void KeyWindow::keyReleaseEvent(QKeyEvent *pEvent)
{
    m_bKeyPressed = false;
    repaint();
}
```

Пример

Пример (key-window.cpp, окончание)

```
void KeyWindow::paintEvent(QPaintEvent *pEvent)
{
    QPainter painter(this);
    painter.fillRect(
        rect(),
        m_bKeyPressed ? Qt::darkGray : Qt::lightGray);
}

// End of File
```

Пример

Пример (timer-window.h)

```
#ifndef TIMER_WINDOW_H__
#define TIMER_WINDOW_H__

#include <QWidget>

class TimerWindow : public QWidget
{
public:
    //
    TimerWindow(QWidget *pParent = 0);
    //
protected:
    //
```

Пример (продолжение)

Пример (timer-window.h, окончание)

```
void showEvent(QShowEvent *pEvent);  
void hideEvent(QHideEvent *pEvent);  
void timerEvent(QTimerEvent *pEvent);  
void paintEvent(QPaintEvent *pEvent);  
//  
private:  
//  
int m_nTimerId;  
int m_nRadius;  
};    // class TimerWindow  
  
#endif    // TIMER_WINDOW_H__
```


Пример (продолжение)

Пример (timer-window.cpp)

```
#include "timer-window.h"

#include <QtWidgets>

TimerWindow::TimerWindow(QWidget *pParent)
    : QWidget(pParent),
      //
      m_nTimerId(0),
      m_nRadius(0)
{
    //
}
```

Пример (продолжение)

Пример (timer-window.cpp, продолжение)

```
void TimerWindow::showEvent(QShowEvent *pEvent)
{
    m_nTimerId = startTimer(100);
}

void TimerWindow::hideEvent(QHideEvent *pEvent)
{
    killTimer(m_nTimerId);
}
```

Пример (продолжение)

Пример (timer-window.cpp, продолжение)

```
void TimerWindow::timerEvent(QTimerEvent *pEvent)
{
    if (pEvent->timerId() == m_nTimerId)
    {
        m_nRadius = (m_nRadius + 1) % 10;
        repaint();
    }
    else
        QWidget::timerEvent(pEvent);
}
```

Пример

Пример (timer-window.cpp, окончание)

```
void TimerWindow::paintEvent(QPaintEvent *pEvent)
{
    const int cnRadius = 10 + 10 * m_nRadius;
    const QColor cColor(
        30 + 22 * m_nRadius, 255 - 20 * m_nRadius, 128 + 10 * m_nRadius);
    //
    QPainter painter(this);
    painter.setPen(cColor);
    painter.fillRect(rect(), Qt::white);
    painter.drawEllipse(rect().center(), cnRadius, cnRadius);
}

// End of File
```

Пример

Пример (длительная обработка событий)

```
void MyWindow::someEvent(QSomeEvent *pEvent)
{
    for (int i = 0; i < m_nMax; ++ i)
    {
        doLengthyTask(i);
        QApplication::instance()->processEvents();
    }
}
```