



第7篇 (基础) Qt定时器和随机数

一、定时器

二、随机数

代码地址：<https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Example07>

一、定时器

Qt中有两种方法来使用定时器，一种是定时器事件，另一种是使用信号和槽。一般使用了多个定时器时最好使用定时器事件来处理。

1. 新建Qt Gui应用，项目名称为 `myTimer`，基类选择 `QWidget`，类名为 `Widget`。
2. 到 `widget.h` 文件中添加函数声明：

```
protected:
    void timerEvent(QTimerEvent *);
```

然后添加私有变量定义：

```
int id1, id2, id3;
```

3. 下面到设计模式，向界面上拖入两个标签部件 `Label`。
4. 下面进入 `widget.cpp` 文件，先在构造函数中添加如下代码：

```
id1 = startTimer(1000); // 开启一个1秒定时器，返回其ID
id2 = startTimer(2000);
id3 = startTimer(10000);
```

这里开启了三个定时器，分别返回了它们的 `id`，这个 `id` 用来区分不同的定时器。定时器的时间单位是毫秒。每当一个定时器溢出时，都会调用定时器事件处理函数，我们可以在该函数中进行相应的处理。

5. 下面添加定时器事件处理函数的定义：

```
void Widget::timerEvent(QTimerEvent *event)
{
    if (event->timerId() == id1) {          // 判断是哪个定时器
        ui->label->setText(tr("%1").arg(qrand()%10));
    }
    else if (event->timerId() == id2) {
        ui->label_2->setText(tr("hello world!"));
    }
    else {
        qApp->quit();
    }
}
```

这里先使用 `timerId()` 函数返回了溢出的定时器的 `id`，然后根据该 `id` 来判断是哪个定时器溢出了，并进行相应的处理。每当第一个定时器溢出时都产生一个小于10的随机数；当第二个定时器溢出时，就更改标签的文本；当第三个定时器溢出时则退出应用程序。现在可以运行程序，查看效果。

6. 如果只是想开启少量的定时器，也可以使用信号和槽来实现。

先在 `widget.h` 中添加一个私有槽声明：

```
private slots:
    void timerUpdate();
```

然后到设计模式向界面上添加一个行编辑器部件 `Line Edit`，再到 `widget.cpp` 中添加头文件包含：

```
#include <QTimer>#include <QDateTime>
```

然后在构造函数中添加如下代码：

```
QTimer *timer = new QTimer(this);
// 关联定时器溢出信号和相应的槽函数
connect(timer, SIGNAL(timeout()), this, SLOT(timerUpdate()));
timer->start(1000);
```

这里创建了一个定时器，并将其溢出信号和更新槽关联起来，最后使用 `start()` 函数来开启定时器。

下面添加 `timerUpdate()` 函数的定义：

```
void Widget::timerUpdate()
{
    //获取系统现在的时间
    QDateTime time = QDateTime::currentDateTime();
    //设置系统时间显示格式
    QString str = time.toString("yyyy-MM-dd hh:mm:ss dddd");
    //在标签上显示时间
    ui->lineEdit->setText(str);
}
```

这里在行编辑器中显示了当前的时间。现在可以运行程序，查看效果。

二、随机数

关于随机数，在Qt中是使用 `qrand()` 和 `qsrand()` 两个函数实现的。在前面的程序中已经看到了 `qrand()` 函数的使用，其可以产生随机数，`qrand()%10` 可以产生0-9之间的随机数。要想产生100以内的随机数就是 `%100`。以此类推。

在使用 `qrand()` 函数产生随机数之前，一般要使用 `qsrand()` 函数为其设置初值，如果不设置初值，那么每次运行程序，`qrand()` 都会产生相同的一组随机数。为了每次运行程序时，都可以产生不同的随机数，我们要使用 `qsrand()` 设置一个不同的初值。这里使用了 `QTime` 类的 `secsTo()` 函数，它表示两个时间点之间所包含的秒数，比如代码中就是指从零点整到当前时间所经过的秒数。

下面先在 `widget.cpp` 的构造函数中添加如下代码：

```
qsrand(QTime(0, 0, 0).secsTo(QTime::currentTime()));
```

然后在 `timerUpdate()` 函数的最后添加如下代码：

```
int rand = qrand() % 300;           // 产生300以内的正整数
ui->lineEdit->move(rand, rand);
```

这样，每过一秒，行编辑器都会移动到一个随机的位置。大家可以运行程序，查看效果。