



# 第15章 Qt 5单元测试框架

## 15.1 QTestLib框架

## 15.2 简单的Qt单元测试

## 15.3 数据驱动测试

## 15.4 简单性能测试





## 15.1 QTestLib框架

QTestLib提供了许多特性，见表15.1。

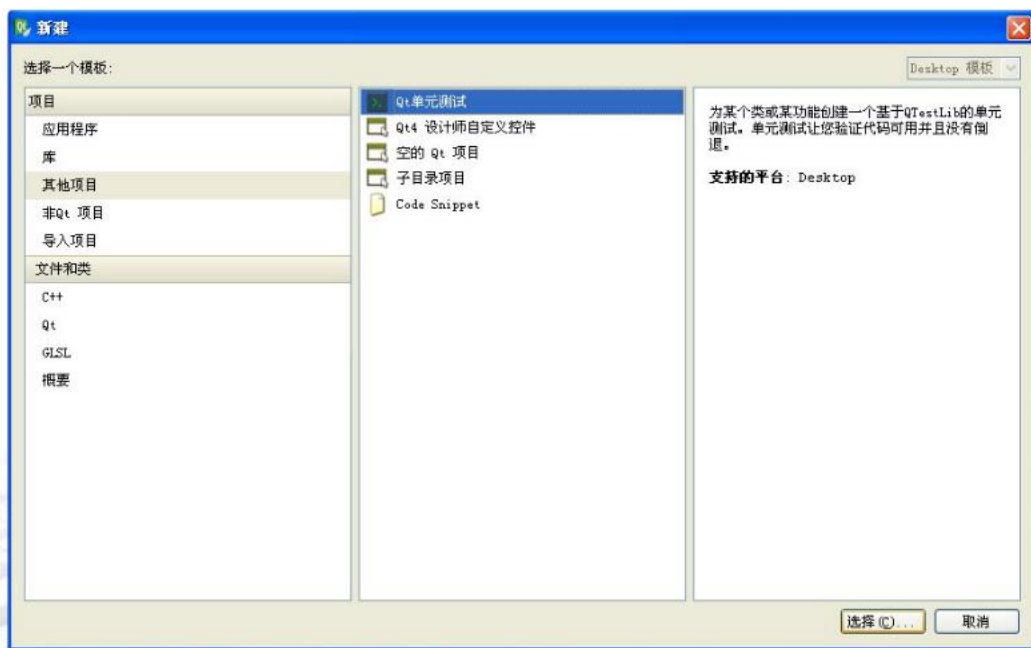
特 性	详 细 描 述
轻量级	QTestlib 只包含 6000 行代码和 60 个导出符号
自包含	对于非 GUI 测试，QTestlib 只需要 Qt 核心库的几个符号
快速测试	QTestlib 不需要特殊的测试执行程序，不需要为测试而进行特殊的注册
数据驱动测试	一个测试程序可以在不同的测试数据集上执行多次
基本的 GUI 测试	QTestlib 提供了模拟鼠标和键盘事件的功能
IDE 友好	QTestlib 的输出信息可以被 Visual Studio 和 KDevelop 解析
线程安全	错误报告是线程安全的、原子性的
类型安全	对模板进行了扩展使用，防止由隐式类型转换引起的错误
易扩展	用户自定义类型可以容易地加入到测试数据和测试输出中



## 15.2 简单的Qt单元测试

(1) 建立单元测试框架，步骤如下。

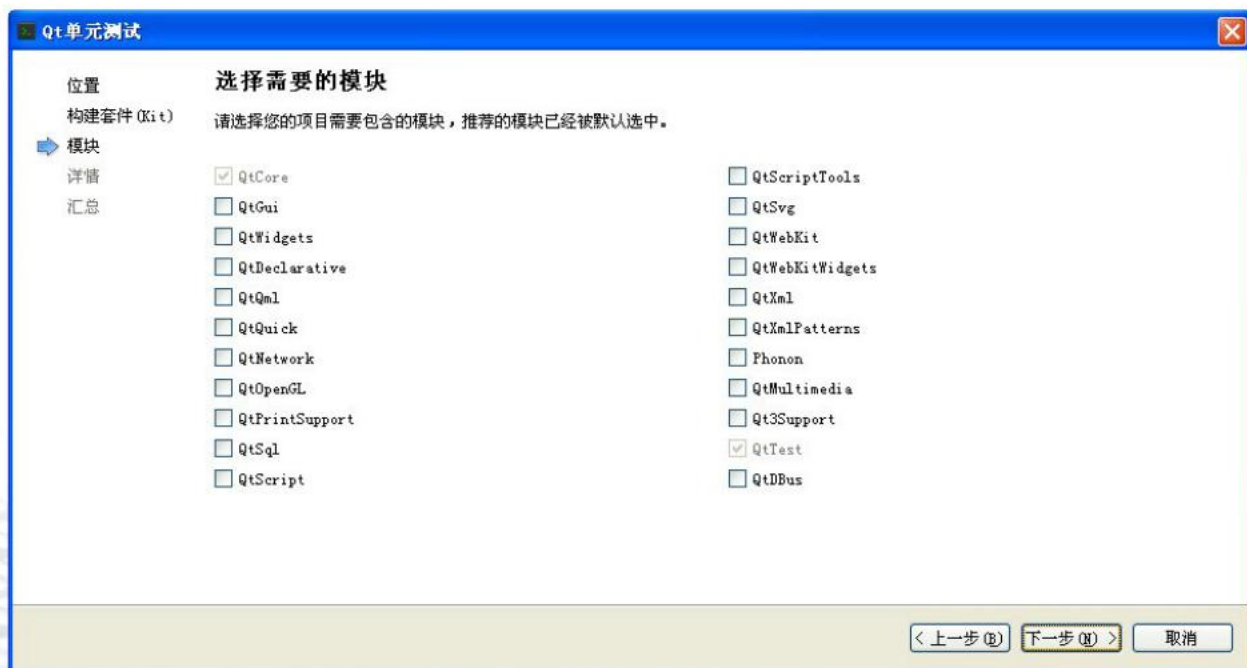
选择“文件”→“新建文件或项目”菜单项，出现如图15.1所示的对话框，选择“其他项目”→“Qt单元测试”菜单项，单击“选择”按钮继续。





## 15.2 简单的Qt单元测试

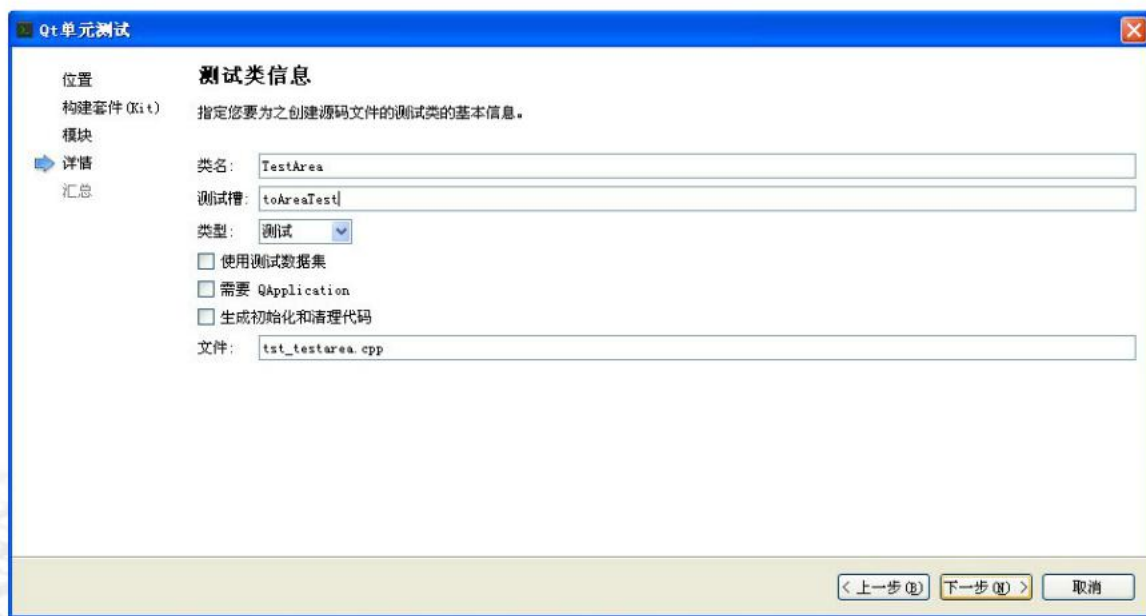
为测试项目命名“名称”为“AreaTest”，单击“下一步”按钮，出现如图15.2所示的对话框，选择项目需要包含的模块。





## 15.2 简单的Qt单元测试

单击“下一步”按钮，在如图15.3所示的对话框中设置将要创建的测试类的基本信息。

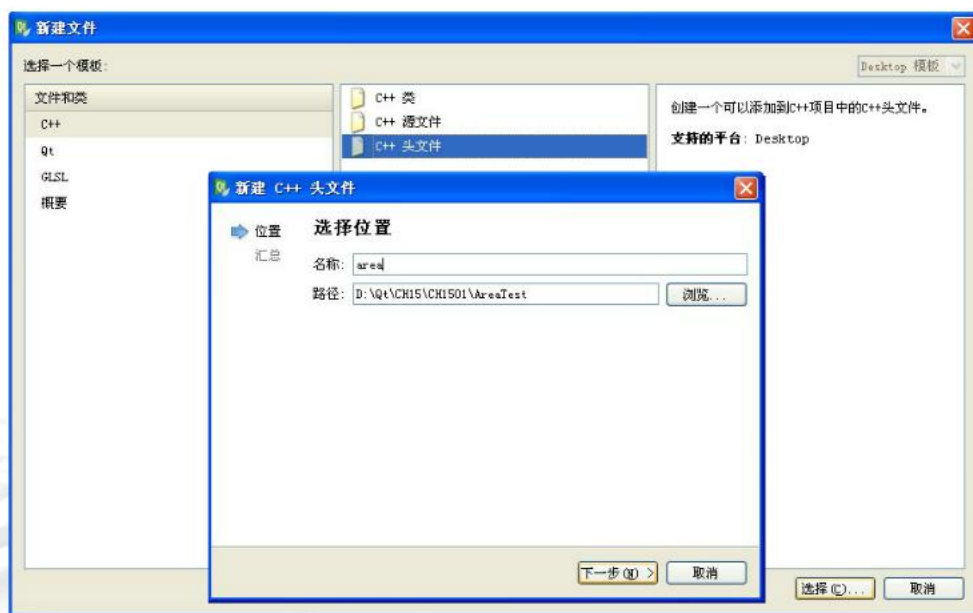




## 15.2 简单的Qt单元测试

(2) 计算圆面积类的具体实现步骤如下。

在项目名上单击鼠标右键，图15.4所示的“新建文件”对话框中，选择新建“C++ 头文件”，单击“选择”按钮，在弹出的“新建C++头文件”对话框中填写文件的“名称”为“area”。





## 15.2 简单的Qt单元测试

头文件“area.h”的具体代码如下：

```
#include <QObject>

class Area:public QObject
{
    Q_OBJECT

public:
    Area(){}
    ~Area(){}
    Area(const Area &area)
    {
        m_r = area.m_r;
    }
    Area(int r)
    {
        m_r=r;
    }
    double CountArea()
    {
        return 3.14*m_r*m_r;
    }
private:
    double m_r;
};
```





## 15.2 简单的Qt单元测试

(3) 测试代码所在的源文件“tst\_testarea.cpp”[的具体内容](#)。  
测试函数运行结果如图15.5所示。

```
***** Start testing of TestArea *****  
Config: Using QTest library 5.0.2, Qt 5.0.2  
PASS : TestArea::initTestCase()  
PASS : TestArea::toAreaTest()  
PASS : TestArea::cleanupTestCase()  
Totals: 3 passed, 0 failed, 0 skipped  
***** Finished testing of TestArea *****
```







## 15.3 数据驱动测试

例1，测试字符串转换为全小写字符的功能。

(1) 建立单元测试框架（操作方法同前），具体设置如下。

项目名称：TestQString

测试类名：TestQString

测试槽：testToLower

生成源文件：tst\_testqstring.cpp

(2) 源文件“tst\_testqstring.cpp”[的具体代码。](#)





## 15.3 数据驱动测试

(3) 测试结果如图15.6所示。

```
***** Start testing of TestQString *****  
Config: Using QTest library 5.0.2, Qt 5.0.2  
PASS   : TestQString::initTestCase()  
PASS   : TestQString::testToLower(lower)  
PASS   : TestQString::testToLower(mixed)  
PASS   : TestQString::testToLower(upper)  
PASS   : TestQString::cleanupTestCase()  
Totals: 5 passed, 0 failed, 0 skipped  
***** Finished testing of TestQString *****
```



## 15.3 数据驱动测试

例2，测试计算圆面积的功能。

(1) 建立单元测试框架（操作方法同前），具体设置如下。

项目名称：AreaTest2

测试类名：TestArea

测试槽：toArea

生成源文件：tst\_testarea.cpp

(2) 新建C++头文件“area.h”，[其具体代码。](#)

(3) 在源文件“tst\_testqstring.cpp”中完成测试工作，[其具体实现代码。](#)



## 15.3 数据驱动测试

(4) 测试结果如图15.7所示。

```
***** Start testing of TestArea *****  
Config: Using QTest library 5.0.2, Qt 5.0.2  
PASS   : TestArea::initTestCase()  
PASS   : TestArea::toArea<1>  
PASS   : TestArea::toArea<2>  
PASS   : TestArea::toArea<3>  
PASS   : TestArea::cleanupTestCase()  
Totals: 5 passed, 0 failed, 0 skipped  
***** Finished testing of TestArea *****
```



## 15.4 简单性能测试

(1) 建立单元测试框架（操作方法同前），具体设置如下。

项目名称：TestQString2

测试类名：TestQString2

测试槽：testBenchmark

生成源文件：tst\_testqstring2.cpp

(2) 源文件“tst\_testqstring2.cpp” [中具体代码。](#)

(3) 测试结果如图15.8所示。

```
***** Start testing of TestQString2 *****
Config: Using QTest library 5.0.2, Qt 5.0.2
PASS   : TestQString2::initTestCase()
PASS   : TestQString2::testBenchmark()
RESULT : TestQString2::testBenchmark():
        0.0010 msecs per iteration (total: 72, iterations: 65536)
PASS   : TestQString2::cleanupTestCase()
Totals: 3 passed, 0 failed, 0 skipped
***** Finished testing of TestQString2 *****
```