

原 如何使用GTest对Qt工程代码做unit test（在ubuntu环境下）

2012年02月29日 11:32:04

doritoswb

阅读数：3632

标签：

qt

ubuntu

单元测试

makefile

testing

application

一 . 前言

最近，笔者参与开发了一个项目，在ubuntu环境下使用qt作为UI framework开发手机上的应用程序，工程也是用qtcreator创建的。

作为程序开发者来说，开发出的功能不仅要符合需求，而且需要保证代码的健壮性，这就需要对所开发的代码做单元测试。

在本文中，笔者将会介绍一下如何使用gtest（即google test）来对已有的Qt工程代码做单元测试。

关于gtest的知识，笔者推荐《玩转Google开源C++单元测试框架Google Test系列(gtest)》系列，链结如下：

<http://www.cnblogs.com/coderzh/archive/2009/04/06/1426755.html>

下面是笔者的目录结构图，读者可以根据自己的目录结构对本文中出现的配置文件进行修改。

Note：我的目录结构如下

```
build_gtest.sh
gtest-1.6.0/
QtTestProject/
> qt-test-app/
> > addTest.cpp
> > addTest.h
> > main.cpp
> > qt_test_app.pro
- testCases/
  main_test.cpp
> TestCases.pro
```

二. 下载

1. 下载安装QtSDK，里面包含qtcreator工具。下载路径：

<http://qt.nokia.com/downloads>

笔者下载的是linux下的4.8版本。文件名为：

Qt_SDK_Lin32_offline_v1_2_en.run

2. 下载gtest包。下载路径：

<http://code.google.com/p/googletest/downloads/detail?name=gtest-1.6.0.zip&can=2&q=>

笔者下载的是1.6.0版本，包名为：gtest-1.6.0

三 . 安装QtSDK

1. 修改安装文件的属性，使之可执行

```
chmod +x Qt_SDK_Lin32_offline_v1_2_en.run
```

2. 安装QtSDK

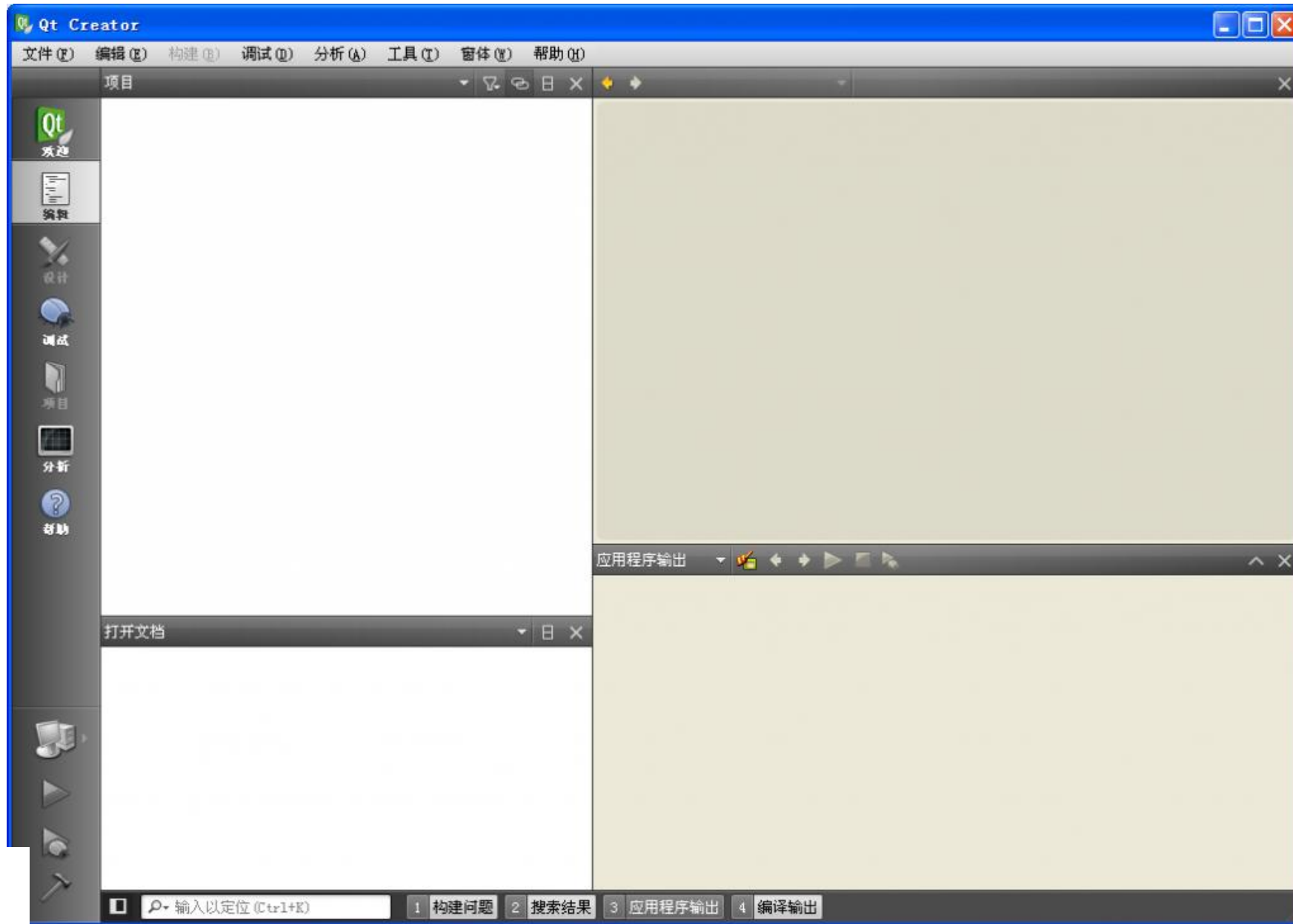
```
./ Qt_SDK_Lin32_offline_v1_2_en.run
```

3. 按提示步骤安装完成QtSDK

qtcreator 将会被安装在 ~/QtSDK/QtCreator/bin/qtcreator目录下，进入该目录

执行qtcreator即可打开qtcreator，利用该IDE可以创建qt的工程。

```
qtcreator
```

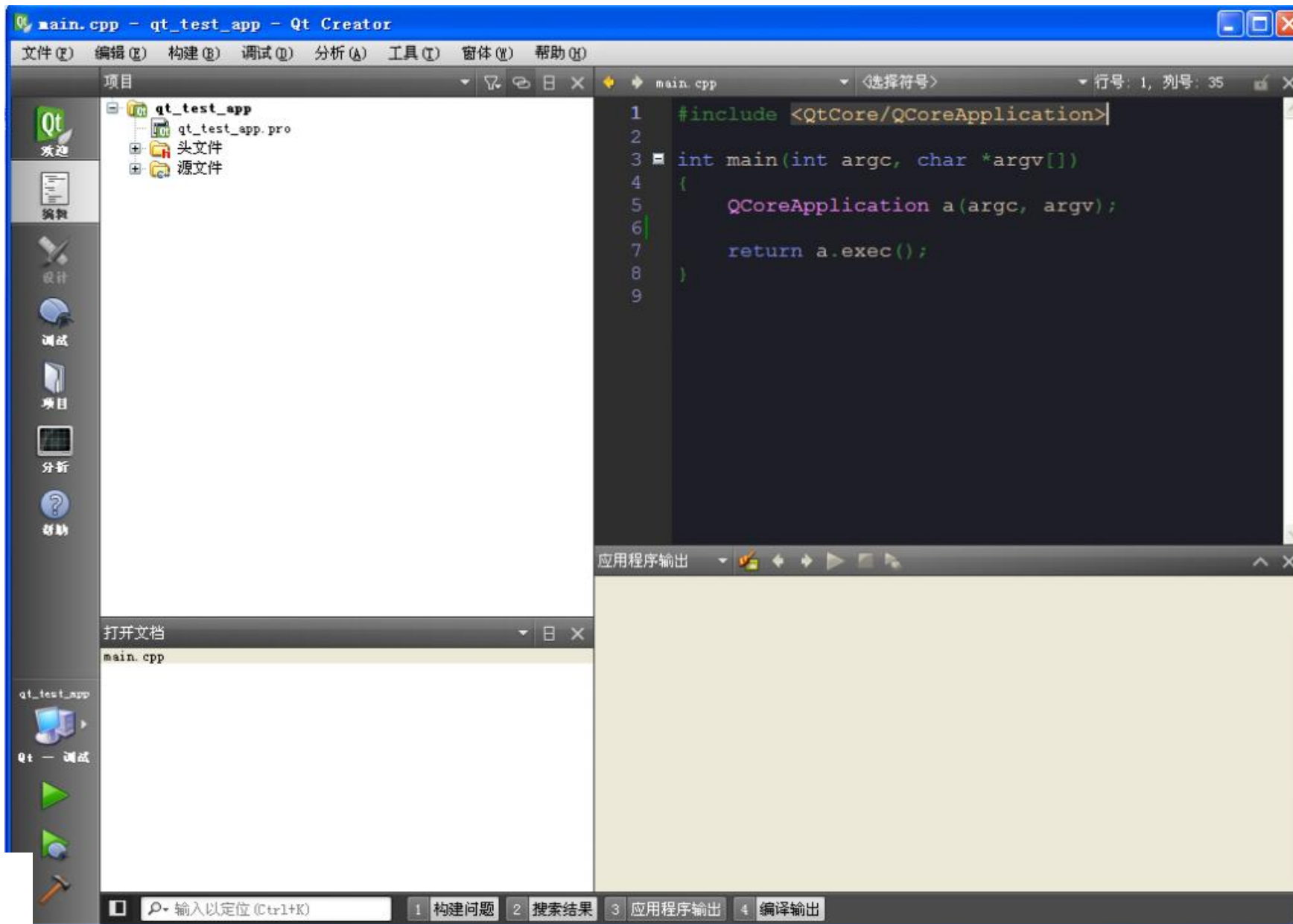


四．创建一个qt示例工程

(这里，我创建了一个比较简单的不带UI界面的Qt 控制台应用当做例子)

1. 打开qtcreator，选择“文件”->“新建工程或文件”，在模板里选择“其他项目”，然后在右边的选择框里选择“Qt 控制台应用”，然后确定工程名称(我的工程名为 qt_test_app)和工程路径，后面的选项一路默认就行了。

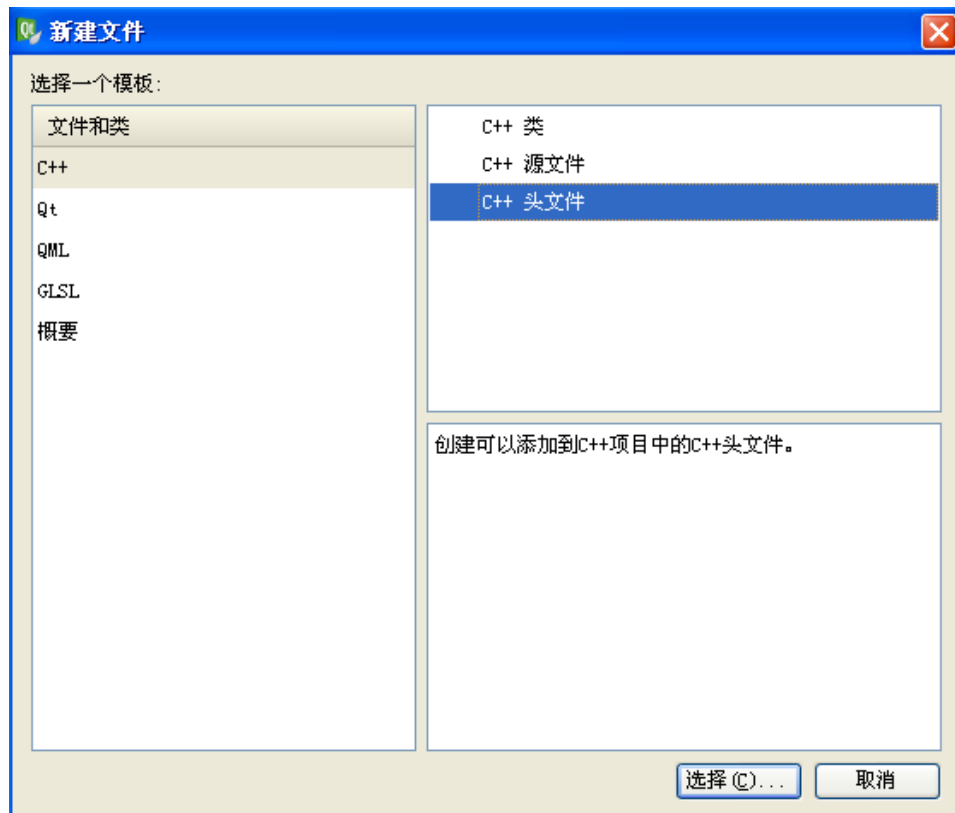
执行完成，最后会生成一个Qt工程qt_test_app



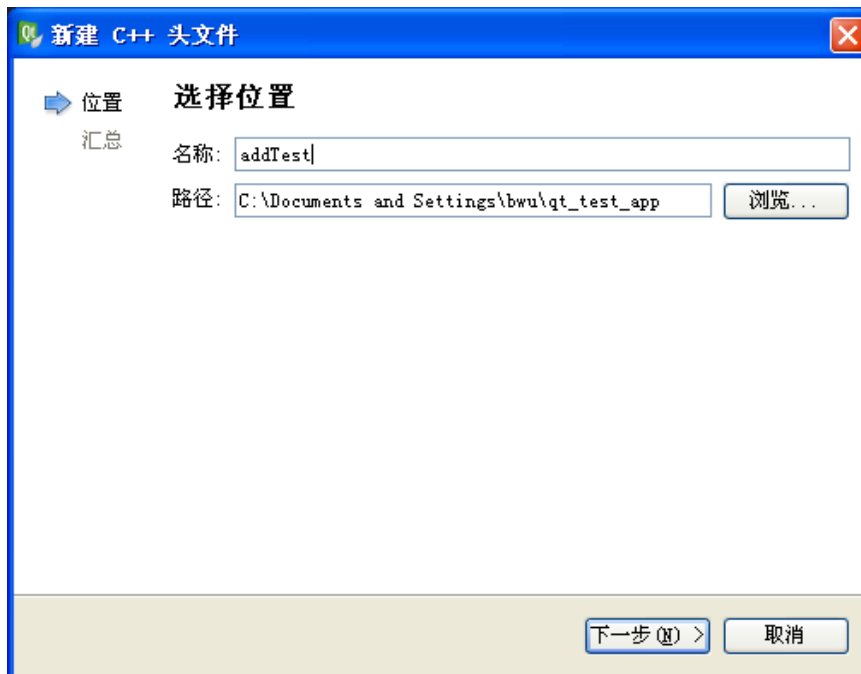
该工程里只有一个文件main.cpp，qtcreator已经产生了部分代码。

2. 新建并添加addTest.h和 addTest.cpp 到工程中

在工程名qt_test_app上点击右键，选择“添加新文件”，选择“c++ 头文件”



点击“选择”，



输入文件名，点击下一步



选择将文件添加到qt_test_app.pro项目里。点击“完成”，文件添加完毕。

使用同样方法创建addTest.cpp文件（新建文件时，需选择c++源文件），也将其添加至qt_test_app.pro工程里。

3. 添加代码

(1) addTest.h 代码如下：

```
#ifndef ADDTEST_H
#define ADDTEST_H

class addTest
{
public:
    addTest();
```



```
~addTest();

int add(int a, int b);

};

#endif // ADDTEST_H
```

(2) addTest.cpp代码如下：

```
#include "addTest.h"

addTest::addTest()

{}

addTest::~addTest()

{}

int addTest::add(int a, int b)

{

    return (a+b);

}
```

(3) main.cpp代码如下：

```
#include <QtCore/QCoreApplication>
```

```
#include "addTest.h"

#include <QtCore/QDebug>

int main(int argc, char *argv[])

{

    QCoreApplication a(argc, argv);

    addTest test;

    int result = test.add(5, 10);

    qDebug()<<"result = "<<result;

    return a.exec();

}
```

4. 在pro工程文件目录下执行qmake，生成makefile

5. 编译运行工程

可以在Application Output窗口里看到输出

result = 15

ok，qt 示例工程创建完毕。接下来就要演示如何使用gtest来对它的代码进行单元测试了。

· 编译libgtest.a包

· 解压gtest包

unzip gtest-1.6.0.zip

2. 配置gtest

```
cd gtest-1.6.0
```

```
./configure
```

```
make
```

gtest-1.6.0不需要make install，我们只是使用gtest的库。

3. 编译libgtest.a包（这个包包含了gtest的核心，将来会被用于qt工程的单元测试中）

将脚本build_gtest.sh放到与gtest-1.6.0同一级的目录下。

执行该脚本

```
./build_gtest.sh
```

执行成功后，可在gtest-1.6.0/lib目录下看到libgtest.a包已经生成。

脚本内容如下：

```
#!/bin/sh
```

```
cd gtest-1.6.0
```

```
g++ -I./include -I./ -c ./src/gtest-all.cc
```

```
ar -rv libgtest.a gtest-all.o    #这一步就是生成libgtest包
```

```
mv -f libgtest.a lib/
```

```
rm gtest-all.o
```

```
cd ..
```

... 创建qt Test工程，并且在其中使用gtest来对qt_test_app工程做单元测试

1. 创建TestCases工程，工程TestCases.pro文件内容如下：

```
include (../QtTestProject/qt_test_app/qt_test_app.pro)

DESTDIR = .

OBJECTS_DIR = .obj

QtTestAppDir = ../QtTestProject/qt_test_app

TEMPLATE = app

TARGET =

DEPENDPATH += . $$QtTestAppDir

INCLUDEPATH += . ../gtest-1.6.0/include ../gtest-1.6.0/include/gtest ../gtest-1.6.0/include/gtest/internal \

    $$QtTestAppDir

LIBS += -L../gtest-1.6.0/lib -lgtest    #这一句就是把gtest的包包含进来，使用libgtest.a包来做单元测试的

#Test cases

SOURCES += \

    main_test.cpp \

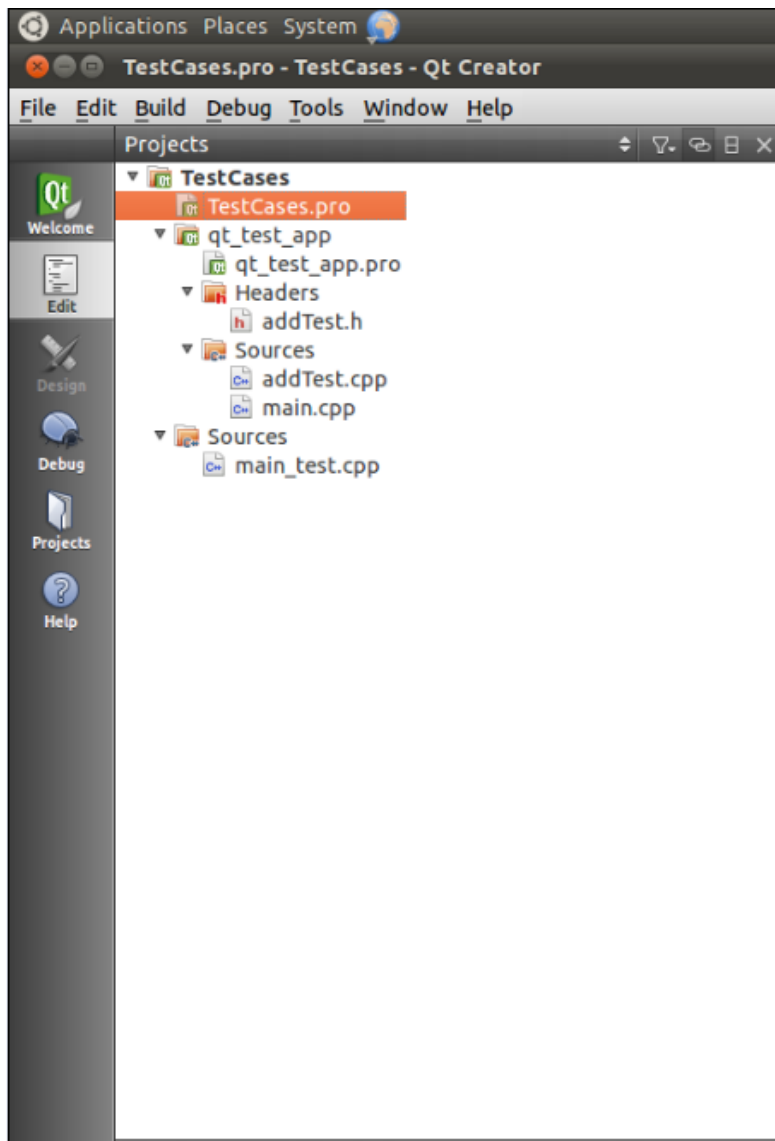
    HEADERS +=
```

```
SOURCES -= main.cpp      #去除qt_test_app工程里面的main.cpp文件，以免出现TestCase工程里有两个main()的错误。
```

此project文件内容完全根据我自己的目录结构来写的，读者可以根据自己的目录结构进行修改，如果不太清楚pro工程文件里面的各项内容的意思，可以参考以下链接：

<http://www.cnblogs.com/Cmpl/archive/2011/08/14/2138516.html>

2. 在pro文件目录下，执行qmake，生成makefile
3. 用qtcreator 打开该工程，就会看到qt-test-app工程已经包含在了TestCases工程里。
4. 新建并添加新文件main_test.cpp到工程TestCases里面（参考第四步）



在main_test.cpp里添加测试代码，用来测试qt_test_app工程里的addTest类的成员函数add()

main_test.cpp的代码如下：

```
#include <gtest/gtest.h>

#include <QtCore/QCoreApplication>

#include "addTest.h"

class QtAppEnvironment: public testing::Environment
{
public:

    QtAppEnvironment(int argc, char ** argv)

        :m_argc(argc),

          m_argv(argv)

    {

    }

    virtual void SetUp()

    {

        app = new QCoreApplication(m_argc,m_argv);

    }

    virtual void TearDown()

    {

        app->exit();

        delete app;

        app = 0;
```

```
    }

    QCoreApplication *app;

    int m_argc;

    char **m_argv;

};

int main(int argc, char ** argv)

{

    testing::AddGlobalTestEnvironment(new QtAppEnvironment(argc, argv));

    testing::InitGoogleTest(&argc,argv);

    return RUN_ALL_TESTS();

}

TEST(addTestCase, Handleadd1)

{

    addTest test;

    EXPECT_EQ(2, test.add(1,1));    //test1

}

TEST(addTestCase, Handleadd2)

{

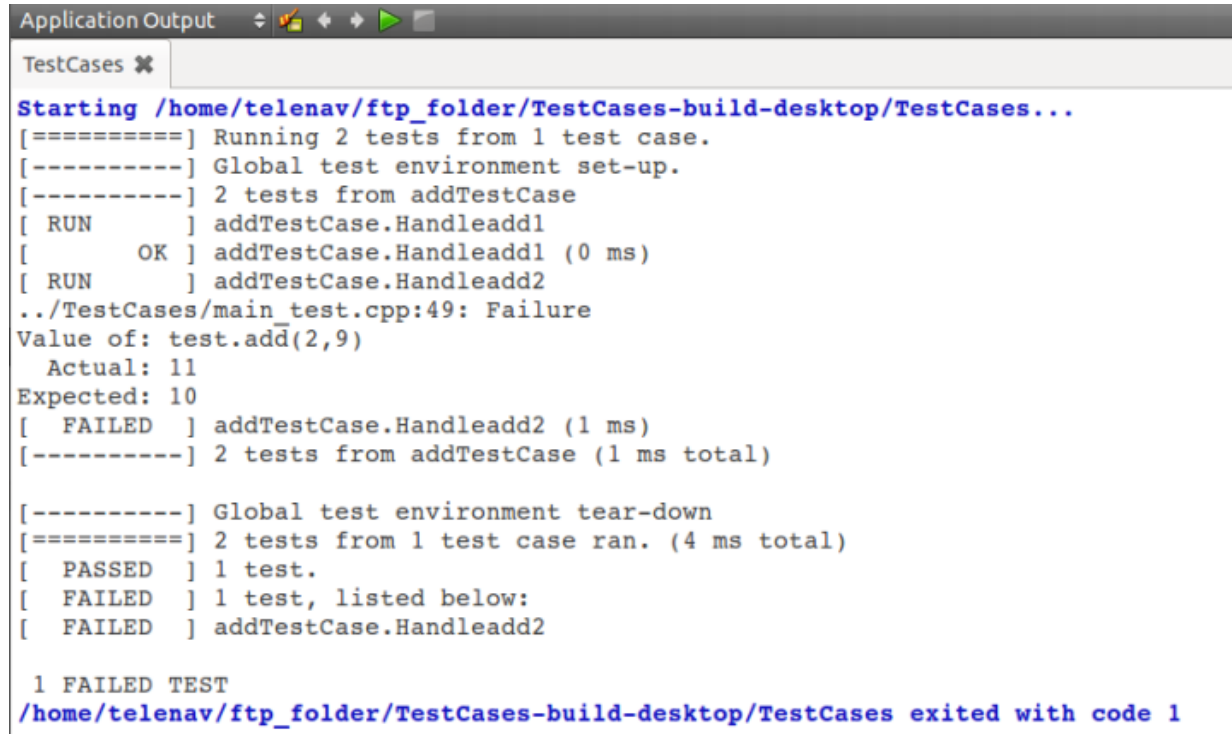
    addTest test;

    EXPECT_EQ(10, test.add(2,9)); //test2 , 故意把expect的值设为错的
```



```
}
```

6. 编译运行TestCases工程，Application Output窗口里输出



```
Application Output
TestCases x
Starting /home/telenav/ftp_folder/TestCases-build-desktop/TestCases...
[=====] Running 2 tests from 1 test case.
[-----] Global test environment set-up.
[-----] 2 tests from addTestCase
[ RUN      ] addTestCase.Handleadd1
[      OK  ] addTestCase.Handleadd1 (0 ms)
[ RUN      ] addTestCase.Handleadd2
../TestCases/main_test.cpp:49: Failure
Value of: test.add(2,9)
  Actual: 11
Expected: 10
[  FAILED  ] addTestCase.Handleadd2 (1 ms)
[-----] 2 tests from addTestCase (1 ms total)

[-----] Global test environment tear-down
[=====] 2 tests from 1 test case ran. (4 ms total)
[  PASSED  ] 1 test.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] addTestCase.Handleadd2

1 FAILED TEST
/home/telenav/ftp_folder/TestCases-build-desktop/TestCases exited with code 1
```

Ok, 可以看到我们设置的两个测试用例，一个成功，一个失败了。

比，我们应该已经了解了如何使用gtest来对qt工程里的代码做单元测试了。

提出疑问，谢谢。

程序员必看！转型人工智能学习大纲，速度来领！

机器学习|深度学习|图像处理|自然语言处理|无人驾驶，这些技术都会吗？看看真正的人工智能师都会那些关键技术？年薪比你高多少！