2/5/2014

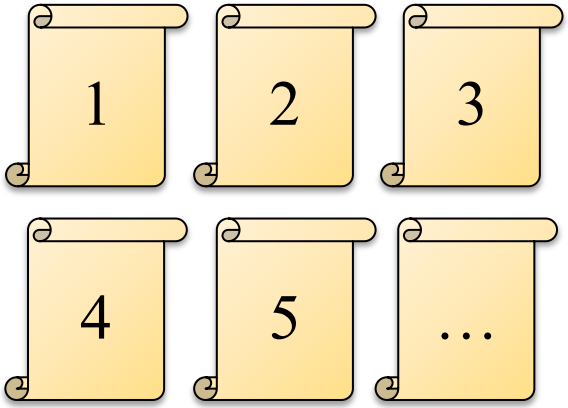
CppUnit
A Testing Framework

Thomas Kilgus

dkfz.
DEUTSCHES
KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT

What is a Unit test?

- Small piece of code which runs automatically in memory
- Unit test = **Isolation**
- One Unit test for exactly **one** thing
- No database- or internet-connection, system calls, hard disc access nor threads are used



CppUnit – Example Step by Step

- Testing Marco header contains useful macros
- mitkTestFixture is the base class
- Dependency to CppUnit is automatically added inside testdrivers, but not outside!

```
#include <mitkTestingMacros.h>
#include <mitkTestFixture.h>

…

MITK_TEST_SUITE_REGISTRATION(mitkImageEqual)
```

Testclass

```
class mitkImageEqualTestSuite : public mitk::TestFixture
{
CPPUNIT_TEST_SUITE(mitkImageEqualTestSuite);
MITK_TEST(Equal_CloneAndOriginal_ReturnsTrue);
MITK_TEST(Equal_InputIsNull_ReturnsFalse);
MITK_TEST(Equal_DifferentImageGeometry_ReturnsFalse);
MITK_TEST(Equal_DifferentPixelTypes_ReturnsFalse);
…
CPPUNIT_TEST_SUITE_END();
…
```

- Inherit from TestFixture
- Create a suite
- Register your test methods

setUp()

```
mitk::Image::Pointer m_Image;
mitk::Image::Pointer m_AnotherImage;

void setUp()
{
//generate a gradient test image
m_Image = itk::ImageGenerator::GenerateGradientImage
        <unsigned char>(3u, 3u, 1u);
m_AnotherImage = m_Image->Clone();
}
```

- Use setUp() to freshly initialize each test
- Testclass may have members

tearDown()

```
void tearDown()
{
m_Image = NULL;
m_AnotherImage = NULL;
}
```

- Use tearDown to clean up memory for each test

2 simple tests

```
void Equal_CloneAndOriginal_ReturnsTrue()
{
MITK_ASSERT_EQUAL( m_Image, m_Image->Clone(),
"A clone should be equal to its original.");
}

void Equal_InputIsNull_ReturnsFalse()
{
mitk::Image::Pointer image = NULL;
MITK_ASSERT_NOT_EQUAL( image, image, "Input is
NULL. Result should be false.");
}
```

Another test

```
void Equal_DifferentImageGeometry_ReturnsFalse()
{
mitk::Point3D origin;
origin[0] = 0.0;
origin[1] = 0.0;
origin[2] = mitk::eps * 1.01;
m_AnotherImage->GetGeometry()->SetOrigin(origin);

MITK_ASSERT_NOT_EQUAL( m_Image, m_AnotherImage,
"One origin was modified. Result should be
false.");
}
```

How to load data?

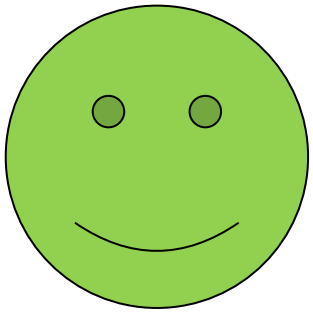
```
private:
std::string m_ImagePath;

public:
void setUp()
{
m_ImagePath =
GetTestDataFilePath("Pic3D.nrrd");
}
```

- Get your data from MITK-Data (what about MBI-Data?)
- No more custom tests necessary in CMakeLists.txt

Don‘t use deprecated methods anymore!

- MITK_TES_CONDITION_REQUIRED
- MITK_TES_CONDITION

- CPPUNIT_ASSERT



- Questions?