07/09/2012

Exception Handling in MITK

Alfred Franz

GERMAN CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Exception Handling: Basics

- MITK now has a unified concept for error handling: **Exceptions**

- Use is very simple:

  - Throw an exception

```
mitkThrow() << "Here comes the error message";
```

  - Catch an exception

```
try
    {
    //code which might throw an exception
    }
catch(const mitk::Exception& e)
    {
    //handle mitk exception here
    }
```

Please catch exceptions as const references for better performance!

Exception Handling: Concept

- Errors during runtime: your program/library should NOT crash or only give error messages.
- Established way for error handling: Exceptions
- In case of Error: an exception is thrown, overlying classes can catch it and handle the error
- But: only catch exceptions which can be handled in a proper way
  - ➢ Therefore exceptions are specifiable by their type

~~`catch (...)`~~

literature:
[1] Tim Bailey, „C++ Exceptions“, 2006
[2] Herb Sutter, „Pragmatic Look at Exception Specifications“, 2009
[3] Herb Sutter, „GotW #65 Try and Catch Me“, 2009
[4] Diane M. Strong et al., „Exceptions and Exception Handling in Computerized Information Processes“, 1995
[5] Christophe Dony, „Exception Handling and Object-Oriented Programming: towards a synthesis.”, 1990

MITK Exception Handling Concept

- MITK exception handling is based on ITK exception handling
- Thus:
  - Macros are used for throwing exceptions to remember filename and linenumber
  - mitk::Exception objects are child classes of itk::ExceptionObject

Exception Handling: Advanced Options

- Exception Message Streaming
- Documentation of Exceptions
- Catching Exceptions
- Defining and Using Specialized Exceptions

Exception Handling: Message Streaming

- Exception message / description: use streaming operator <<
- Object information can also be added

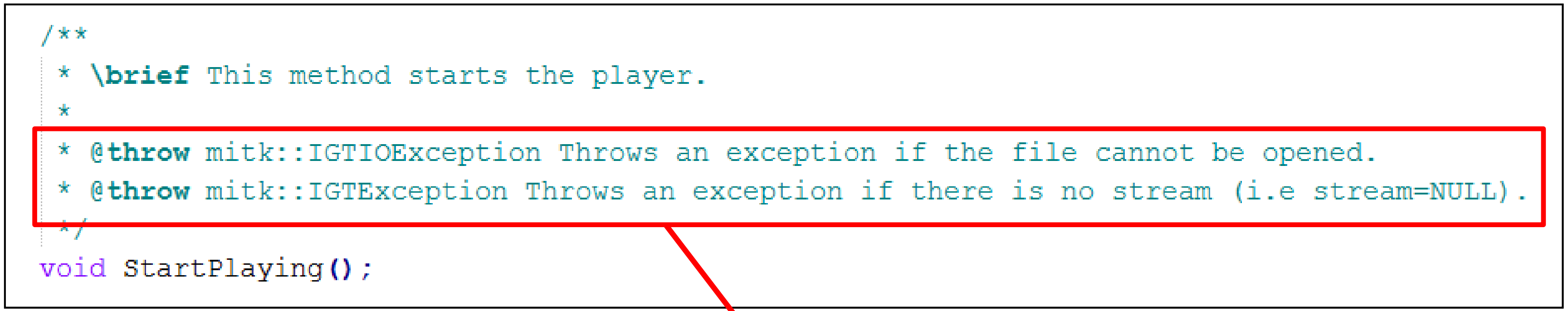
```
mitkThrow() << "Show the values of some variables:" << m_MyObject->GetSize() << m_MyInteger;
```

- Get this message later when catching the exception:

```
catch (const mitk::Exception& e)
    {
    std::string message = e.GetDescription();
    }
```

Exception Handling: Documentation

- Don‘t forget to document exceptions when documenting your methods!



```
/**
 * \brief This method starts the player.
 *
 * @throw mitk::IGTIOException Throws an exception if the file cannot be opened.
 * @throw mitk::IGTException Throws an exception if there is no stream (i.e stream=NULL).
 */
void StartPlaying();
```

Very important if users of your class want to catch and handle exceptions in the right way.

Exception Handling: Catching Exceptions

- Catch exceptions in c++ style
- Separate between different exceptions using multiple catch-blocks

```
try
    {
    //[...]
    }
catch(const mitk::IGTIOException& e)
    {
    //handle IGTIO exceptions here
    }
catch(const mitk::IGTException& e)
    {
    //handle IGT exceptions here
    }
```

- Sometimes rethrowing of the exception is needed, then use the rethrow macro

```
catch(mitk::Exception& e)
    {
    mitkReThrow(e) << "Message that will be appended to the exception (optional)";
    }
```

→ *The mitkReThrow macro modifies the exception so don‘t use “const“ in this case.*

Defining and Using Specialized Exceptions

Defining specialized exceptions:

- All MITK exceptions should be subclasses of mitk::exception
- Use mitkExceptionClassMacro to implement new exception classes

```
#include <mitkCommon.h>
/** Documentation of exception class */
class mitk::MySpecializedException : public mitk::Exception
{
public:
    mitkExceptionClassMacro(mitk::MySpecializedException,mitk::Exception);
};
```

Using/throwing specialized exceptions:

```
mitkThrowException(mitk::MySpecializedException) << "this is error info";
```

Question

```
a)  throw mitk::Exception("") << "Here was an error!";

b)  mitkThrow() << "Here was an error!";
```

What is the correct way to throw an exception in MITK?

1. a) because it is standard c++
2. b) because a) does not even compile
3. a) because b) is nonsense
4. b) because it remembers filename and line number