

11/25/2016

The `override` specifier¹ from C++11

Clemens Hentschke
Bug squashing seminar

¹ `override` is no keyword in C++ (to avoid smartass discussions)

dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer

The problem

```
class Base
{
    [...]
    virtual void do();
    int doMore(int x);
};

class Derived : public Base
{
    [...]
    void do() const;
    int doMore(unsigned int x);
};
```

- Code compiles
- But neither do(), nor doMore(...) override their base class function!

The solution

```
class Derived : public Base
{
    [...]
    void do() const; //ok, but overrides nothing
    void do() const override //Compiler Error
    void do() override //ok
};
```

- Use the C++11 **override** keyword
- It checks if the function overwrites a base class function
 - If not: a compiler error is thrown!

Override details

- **override** checks if
 - The base class function is virtual
 - Function names are identical
 - Parameters are identical
 - Constness is identical
 - Return types are identical
 - **What also works**
 - `void override(); //compiler is ok, reviewer not`
 - `int override = 42; //compiler is ok, reviewer not`
 - `~Derived() override; //ok`
- `override` is no keyword, but should not be used as variable or function name

Mitk Macros¹

`mitkClassMacro(className, SuperClassName)`

- `GetNameOfClass()` must be defined in SuperClass
- `virtual std::vector<std::string> GetClassHierarchy() const override`
- Example: `mitkClassMacro(Image, SlicedData);`

`mitkClassMacroItkParent(className, SuperClassName)`

- `virtual std::vector<std::string> GetClassHierarchy() const`
- Example: `mitkClassMacroItkParent(CovarianceMatrixCalculator, itk::Object)`

`mitkClassMacroNoParent(className)`

- does not expect anything
- Example: `mitkClassMacroNoParent(FileReader)`

Using `mitkClassMacro` where you should have used `mitkClassMacroItkParent`:
Compile error: 'method' : method with override specifier 'override' did not override any base class methods

¹ introduced by Marco

Conclusion

- Use `override` everywhere you can to specify overriding functions
 - Compiler error is far better than odd runtime behavior
- Clang-tidy has option to add overrides automatically to existing code base

Questions?



dkfz.

GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION



50 Years – Research for
A Life Without Cancer