Lab Presentation Question 2.10

VE482 Team 1

UM-SJTU Joint Institute

November 7, 2018

Virtual Function

Specify a member function as virtual for a base class will guarantee the triggering of dynamic dispatch in run time if the function is overridden in derived classes

Definition

Dynamic dispatch is the process of selecting which implementation of a polymorphic operation to call at run time.

Virtualness of a function can be inherited in a derived class.

Virtual Function Usage


```
class A {
    ...
    void func1(){std::cout << "A::func1";}
    virtual void func2(){std::cout << "A::func2";}
    ...
}
class B: public A {
    ...
    void func1(){std::cout << "B::func1";}
    void func2(){std::cout << "B::func2";}
    ...
}
int main(){
    std::unique_ptr<A> pt = std::make_unique<B>();
    pt->func1(); //Outputs "A::func1"
    pt->func2(); //Outputs "B::func2"
}
```

Abstract Class

Definition

An abstract class is a generic class used as a basis for creating different objects that conform to its protocol, or the set of operations it supports.

An abstract class usually contains

- Some member data and methods which are commonly shared by its derived objects
- Some methods declared but are not implemented(abstract function,usually declared as virtual type fname(args)=0)

Note: An abstract class cannot be instantiated but a pointer to it can own an instance of its derived class, through which polymorphism is achieved.

Friend Keyword

The keyword "friend" is used to grant access of private members of a class. Usually it can be used in two diferent aspects.

- Declare a non-member function in a class as a friend.(Often in operator overloading)
- Declare another class as a friend.

Risk of Using Friend

Using a friend keyword means that something outside a specific class can access the "internal" world of a class, which might violates the principle of encapsulation.

Override Specifier

Features

- Override specifier should be added right after the function declaration in a derived class (type funcname() override;)
- This keyword will guarantee that this function will be virtual and will override a virtual function in base class
- If any of the requirements is not satisfied, the program will not compile

Final Specifier

Features

- Override specifier should be added right after the function declaration in a derived class (type funcname() final;)
- This keyword will guarantee that this function will override a virtual function in base class but cannot be overridden again
- If any of the requirements is not satisfied, the program will not compile

Purpose of the two specifiers

1. Avoid potential mistakes of developers who might forget to implement an override or have something wrong in overriding.

2. Make the implementation more explicit to possible readers.

References

- web.mit.edu/merolish/ticpp/Chapter15.html
- en.wikipedia.org/wiki/Dynamic_dispatch
- www.techopedia.com/definition/17408/abstract-class
- en.cppreference.com/w/cpp/keyword/friend
- www.cprogramming.com/tutorial/friends.html
- en.cppreference.com/w/cpp/language/override
- en.cppreference.com/w/cpp/language/final