5. C++重载、重写和重定义的区别

1. 重载：函数名相同，函数的参数个数、参数类型或参数顺序三者中必须至少有一种不同。函数返回值的类型可以相同，也可以不相同。发生在一个类内部，不能跨作用域。
2. 重写：也叫做覆盖，一般发生在子类和父类继承关系之间。子类重新定义父类中有相同名称和参数的虚函数。(override)
3. 重定义：也叫做隐藏，子类重新定义父类中有相同名称的非虚函数（参数列表可以不同），指派生类的函数屏蔽了与其同名的基类函数。可以理解成发生在继承中的重载。

如果一个派生类，存在重定义的函数，那么，这个类将会隐藏其父类的方法，除非你在调用的时候，强制转换为父类类型，才能调用到父类方法。否则试图对子类和父类做类似重载的调用是不能成功的。

重写需要注意：

1. 被重写的函数不能是static的。必须是virtual的
2. 重写函数必须有相同的类型，名称和参数列表
3. 重写函数的访问修饰符可以不同。

重定义规则如下：

1. 如果派生类的函数和基类的函数同名，但是参数不同，此时，不管有无virtual，基类的函数被隐藏。
2. 如果派生类的函数与基类的函数同名，并且参数也相同，但是基类函数没有vitual关键字，此时，基类的函数被隐藏（如果相同有Virtual就是重写覆盖了）。

```
#include<iostream>

using namespace std;

class Animal
{
public:
    void func1(int tmp)
    {
        cout << "I'm an animal -" << tmp << endl;
    }

    void func1(const char *s)//函数的重载
    {
        cout << "I'm an animal func1 -" << s << endl;
    }

    virtual void func2(int tmp)
    {
        cout << "I'm virtual animal func2 -" << tmp << endl;
    }

    void func3(int tmp)
    {
        cout << "I'm an animal func3 -" << tmp << endl;
```

```
    }
};

class Fish :public Animal
{
public:
    void func1()//函数的重定义 会隐藏父类同名方法
    {
        cout << "I'm a fish func1" << endl;
    }

    void func2(int tmp) //函数的重写， 覆盖父类的方法 override
    {
        cout << "I'm a fish func2 -" << tmp << endl;
    }

    void func3(int tmp) { //函数的重定义 会隐藏父类同名方法
        cout << "I'm a fish func3 -" << tmp << endl;
    }
};

int main()
{
    Fish fi;
    Animal an;

    fi.func1();
    // 由于是重定义 父类的方法已经被隐藏
    // 需要显示声明，重载不能跨作用域
    fi.Animal::func1(1);
    dynamic_cast<Animal *>(&fi)->func1(11); // 强转之后即可调用到父类被隐藏的方法
    dynamic_cast<Animal *>(&fi)->func1("hello world"); // 强转之后即可调用到父类被隐
藏的方法
    fi.func2(2);    // 调用子类
    dynamic_cast<Animal *>(&fi)->func2(22); // 调用"子类方法"(因为是虚函数，会被子类
覆盖)
    dynamic_cast<Animal *>(&fi)->func3(222); // 调用父类
    fi.func3(2222); // 调用子类

    cout << endl << " ************ " << endl;
    an.func1(1);
    an.func1("I'm an animal");
    an.func2(1);
    system("pause");
    return 0;
}
```

输出结果：

```
I'm a fish func1
I'm an animal -1
```

```
I'm an animal -11
I'm an animal func1 -hello world
I'm a fish func2 -2
I'm a fish func2 -22
I'm an animal func3 -222
I'm a fish func3 -2222

 ************
I'm an animal -1
I'm an animal func1 -I'm an animal
I'm virtual animal func2 -1
```

参考来源：C++_重载、重写和重定义的区别_fzzjoy的专栏-CSDN博客_c++重载和重写的区别