

## 4.4 重载、覆盖、隐藏\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.4 重载、覆盖、隐藏涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

\4. 重载、覆盖、隐藏

### 一、重载

成员函数被重载的特征：

1. 相同的范围（在同一个类中）
2. 函数名字相同
3. 参数不同
4. virtual 关键字可有可无。

### 二、覆盖

覆盖是指派生类函数覆盖基类函数，特征是：

- 不同的范围（分别位于派生类与基类）
- 函数名字相同；
- 参数、返回值相同
- 基类函数必须有 virtual 关键字

```
#include <iostream>

using namespace std;

class Parent
{
public:
    virtual void vir_function1()
    {
        cout << "virtual_function1" << endl;
    }

    virtual void vir_function2()
```

```

    {

        cout << "vir_function2" << endl;

    }

private:

    int a;

    int b;

};

class Child:public Parent
{

public:

    virtual void vir_function1()

    {

        cout << "virtual_function1" << endl;

    }

    virtual void vir_function2()

    {

        cout << "vir_function2" << endl;

    }

private:

    int c;

};

int main(void)

{

    Child object;

    return 0;

}

```

```

Breakpoint 1, main () at Test.cpp:43
43      Child object;
(gdb) n
45      return 0;
(gdb) set p object on
(gdb) set p pretty on
(gdb) set p array on
(gdb) p object
$1 = (Child) {
  <Parent> = {
    _vptr.Parent = 0x8048968 <vtable for Child+8>,
    a = -1207963648,
    b = 134514875
  },
  members of Child:
  c = -1209176064
}
(gdb) p /a *0x804896803
$2 = {0x804882c <Child::vir_function1()>,
0x8048858 <Child::vir_function2()>,
0x0}

```

```
#include <iostream>

using namespace std;

class Parent
{
    public:

        virtual void vir_function1()
        {
            cout << "Parent virtual_function1" << endl;
        }

        virtual void vir_function2()
        {
            cout << "Parent vir_function2" << endl;
        }

    private:

        int a;
        int b;
};

class Child:public Parent
{
    public:

        virtual void vir_function1()
        {
            cout << "Child virtual_function1" << endl;
        }

        virtual void vir_function2()
        {
            cout << "Child vir_function2" << endl;
        }

    private:

        int c;
};

int main(void)
{
    Child object;
```

```
Parent *p = &object;

p->vir_function1();//Child virtual_function1

p->vir_function2();//Child virtual_function2

return 0;

}
```

### 三、隐藏

这里“隐藏”是指派生类的函数屏蔽了与其同名的基类函数，规则如下：

1. 如果派生类的函数与基类的函数同名，但是参数不同。此时，不论有无 virtual 关键字，基类的函数将被隐藏（注意别与重载混淆）。
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 virtual 关键字。此时基类的函数被隐藏（注意别与覆盖混淆）。

```
#include <iostream>

using namespace std;

class Parent
{
public:
    void vir_function1()
    {
        cout << "Parent virtual_function1" << endl;
    }

    virtual void vir_function2()
    {
        cout << "Parent virtual_function2" << endl;
    }

private:
    int a;
    int b;
};

class Child:public Parent
{
public:
    virtual void vir_function1()
    {
        cout << "Child virtual_function1" << endl;
    }
}
```

```

    }

    virtual void vir_function2(int a)
    {
        cout << "Child virtual_function2" << endl;
    }

private:
    int c;
};

int main(void)
{
    Parent object;

    object.vir_function1();//Parent virtual_function1
    object.vir_function2();//Parent virtual_function2
    cout << "-----" << endl;

    Child child;

    child.vir_function1();//Child virtual_function1(隐藏了父类函数)

    //error:no matching function for call to 'Child::vir_function2()'
    child.vir_function2();//(隐藏了父类函数)
    //maybe you must child.Parent::vir_function2();

    cout << "-----" << endl;

    Parent *p = &child; //behavior depends on type of the pointer
    p->vir_function1();//Parent virtual_function1

    //
    p->vir_function2();//Parent virtual_function2

    return 0;
}

```

## 四、任务

写出可以体现出重载、覆盖、隐藏的代码

- 划线

- 写笔记

学习要认真，笔记应当先



公开笔记 0/1000 提交



Sunny\_SunshineX

删除 编辑

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

