

```

class Parent
{
public:
    Parent(int a) {
        this->a = a;
    }

    virtual void func(int a) ✓
    {
        cout << "Parent::func(int)... " << endl;
    }

    void func(int a, int b, int c)
    {
        // ...
    }

private:
    int a;
};

class Child : public Parent
{
public:
    Child(int a, int b) : Parent(a) ✓
    {
        this->b = b;
    }

    Virtual void func(int a)
    {
        cout << "Child: func(int)... " << endl;
    }

    Virtual void func(int a, int b) {
        cout << "Child :func(int ,int )..." << endl;
    }

private:
    int b;
};

void myFunc(Parent *pp)
{
    pp->func(10);
}

int main(void)
{
    Parent *pp = new Parent(10);
    Parent *cp = new Child(100, 200);

    myFunc(pp);
    myFunc(cp);

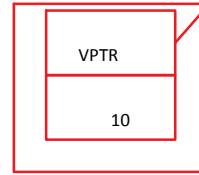
    return 0;
}

```

如果说一个类有virtual 虚函数关键字
在编译器给这个对象开辟空间的时候，
会默认增加一个指针，vptr

Parent p(10);

a



Parent::虚函数表

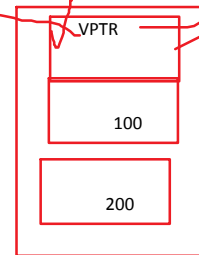
virtual void Parent:: func(int a)

Parent*pp = &c;
Pp->func(10);

Child c(100,200)

a

b



Child::虚函数表

Virtual void Child::func(int a)

Virtual void Child::func(int a, int b)

Virtual void Child::func(int a,int b,int c)

Parent*pp = &p;
Pp->a;
Pp->func(10, 20, 30); func(int)

Parent *pp = &c;
Pp->func(10);
Pp->func(100,200)
Pp->func(100,200,300)