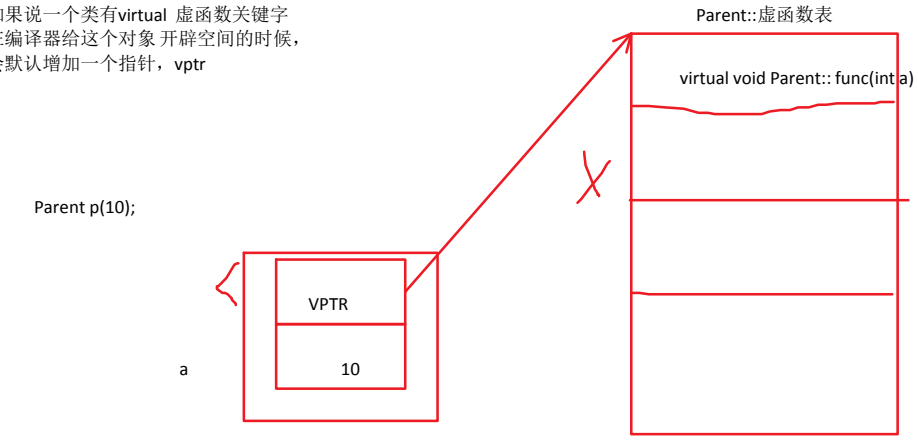
1 多态的虚函数表

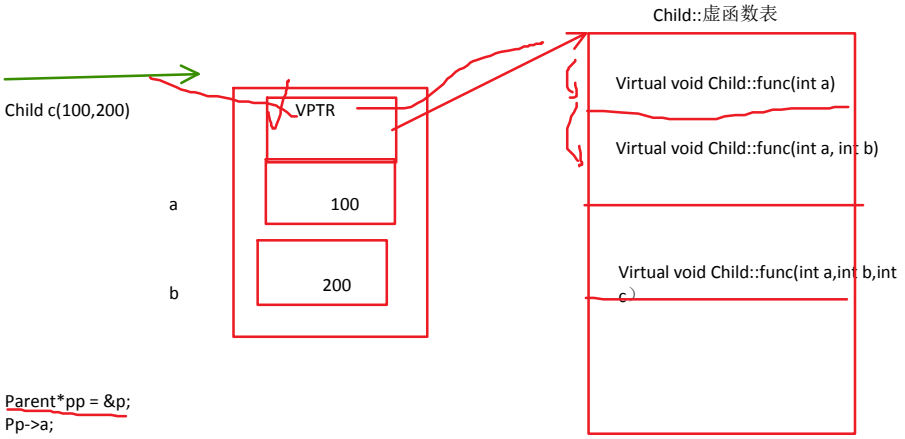
2016年4月19日 9:19

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

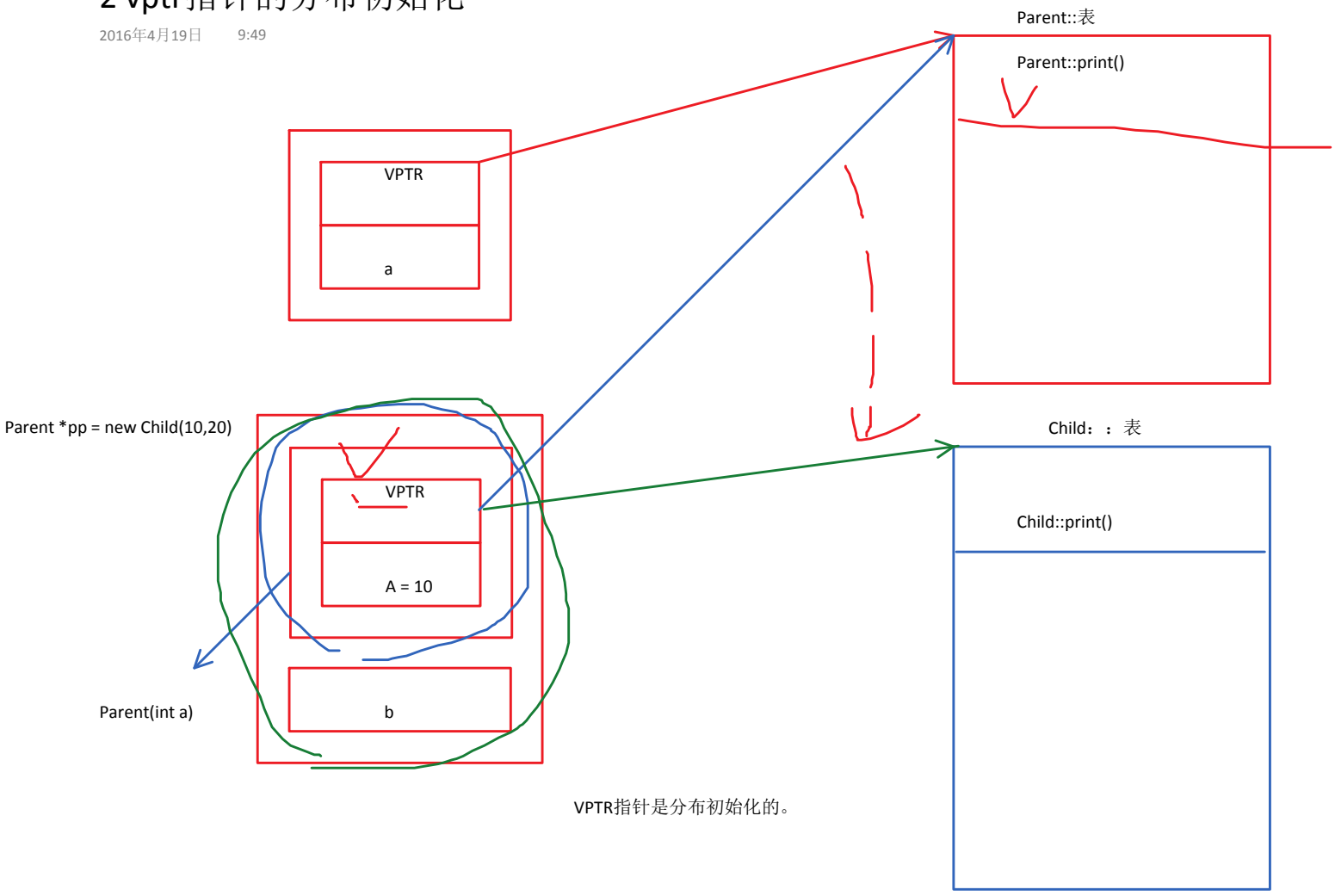


Parent*pp = &p;
Pp->a;
Pp->func(10，20, 30); func(int)

Parent *pp = &c;
Pp->func(10);
Pp->func(100.200)
Pp->func(100,200,300)

2 vptr指针的分布初始化

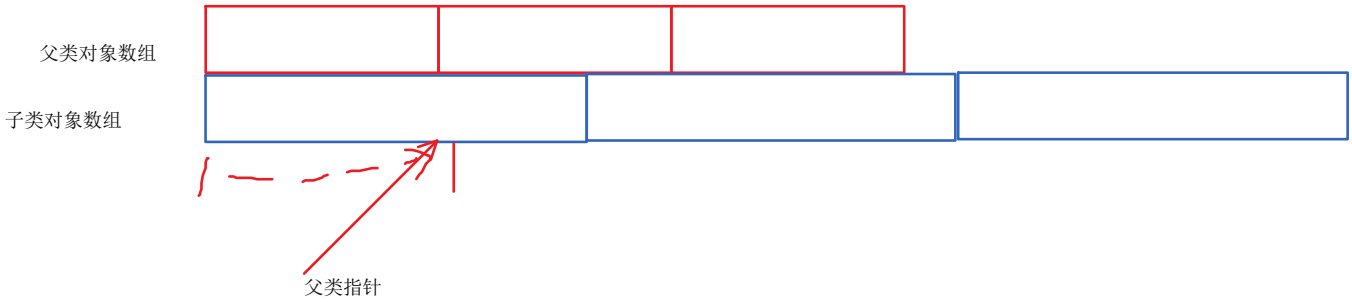
2016年4月19日 9:49



VPTR指针是分布初始化的。

3 父类指针和子类指针的步长

2016年4月19日　　10:47



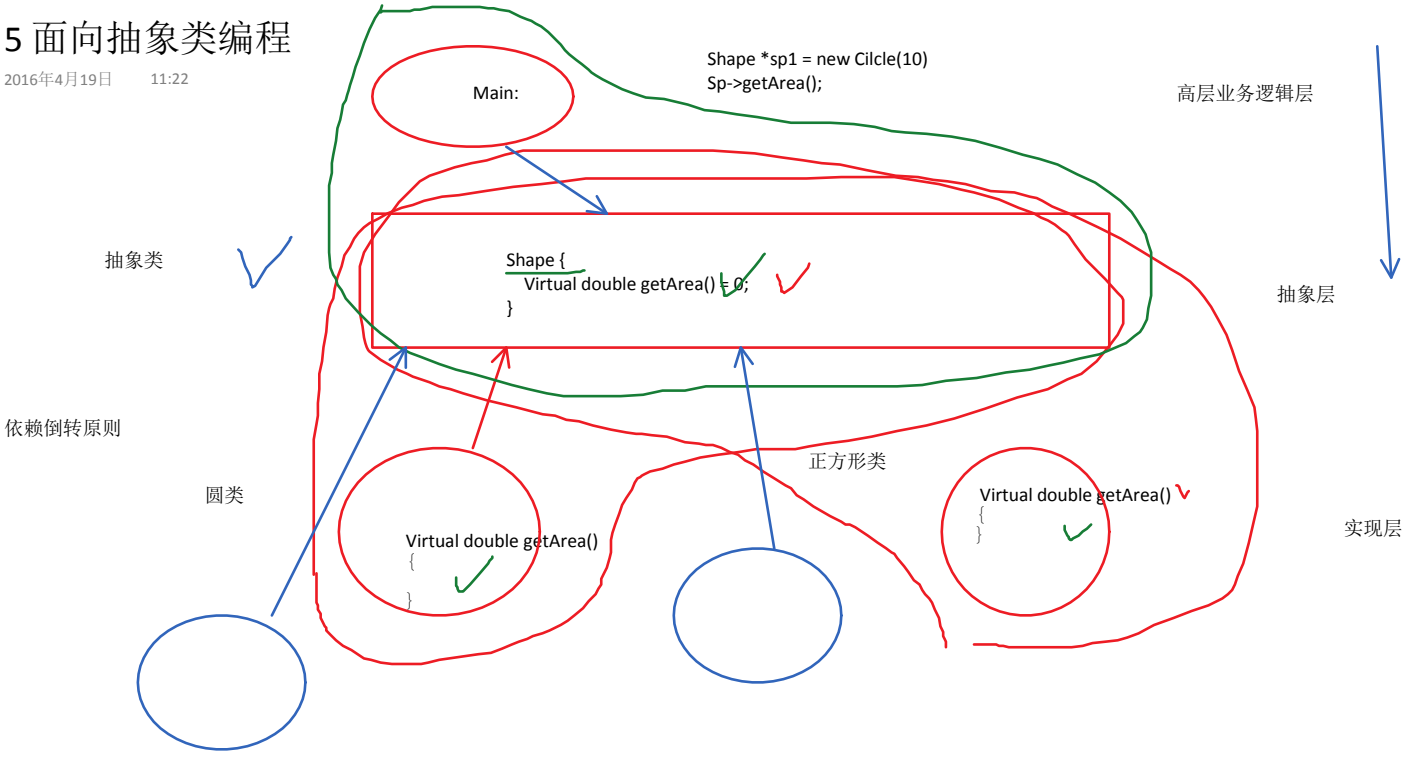
父类指针

4 面向抽象类编程

2016年4月19日 10:58

5 面向抽象类编程

2016年4月19日 11:22



6 抽象类的小案例

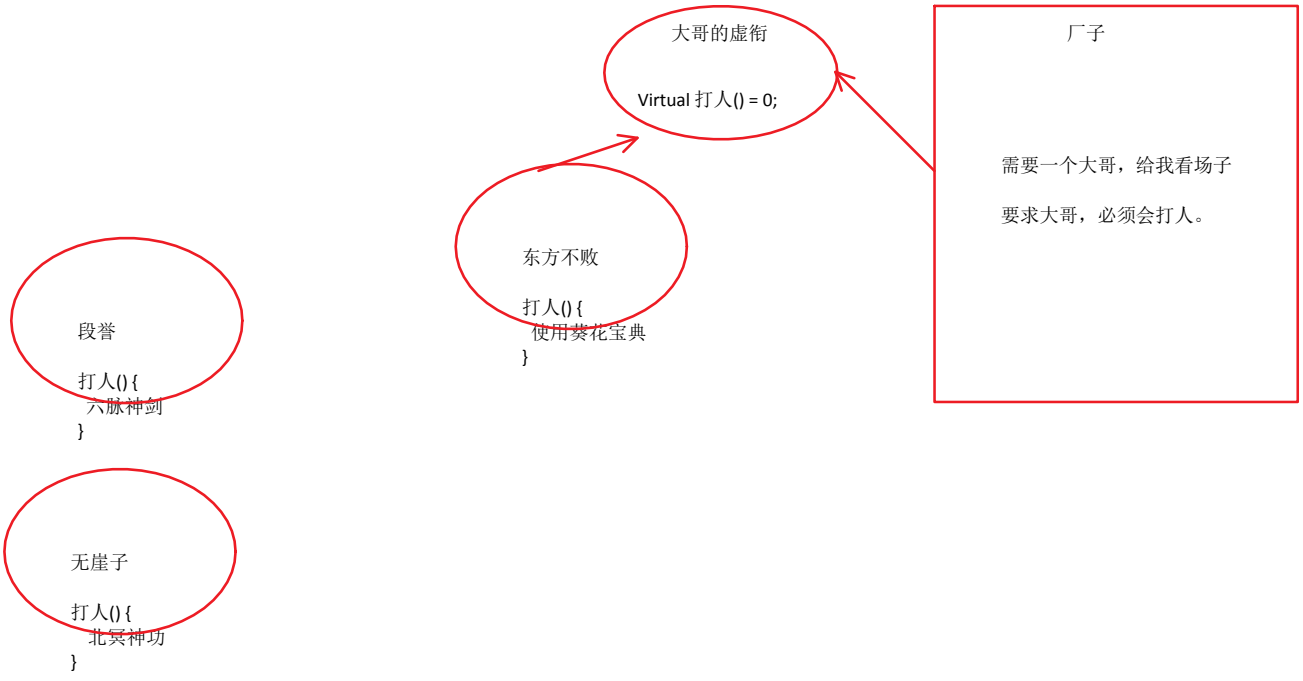
2016年4月19日　　15:01

想成为大哥的人

1. 要有继承。 东方不败继承大哥的虚衔。
2. 要有（纯）虚函数重写。 重写打人，用葵花宝典实现。

3. 父类指针指向子类对象。 BOSS： 大哥，你给我去打人。

BOSS：不需要关系大哥叫啥名，会什么招式 大哥给我去打人。
命名： 大哥 你给我去打人。

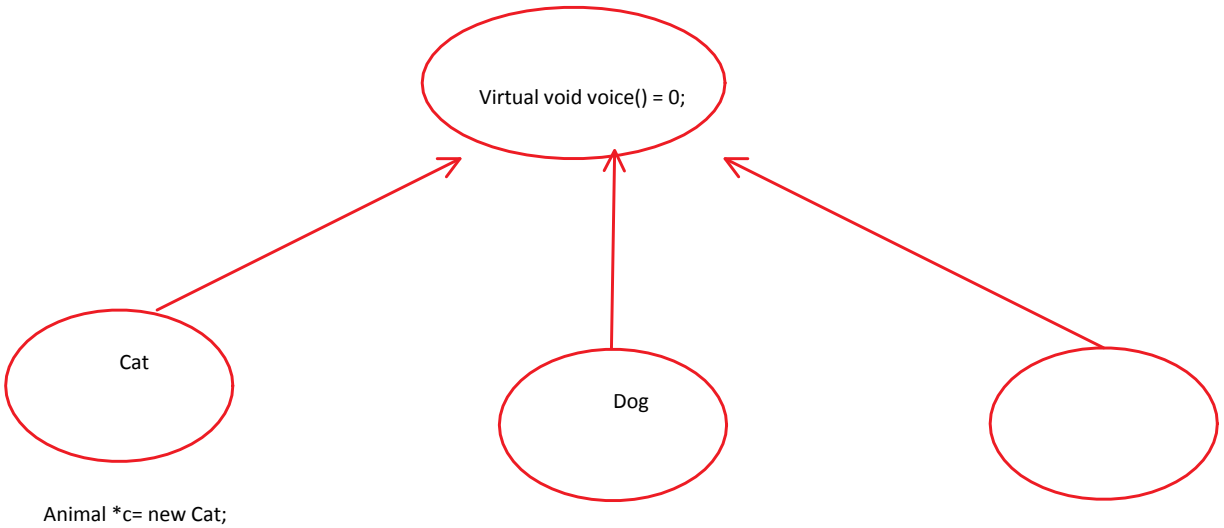


7 面向抽象编程-动物园案例

2016年4月19日　　15:31

抽象类 Animal

多文件形式。



Animal *c= new Cat;
c->voice();

Main: 要求面向抽象类编程。

8 面向抽象类编程--电脑组装案例

2016年4月19日　　16:13