

# 继承

胡船长

初航我带你，远航靠自己

# 一、继承基础

1. 基础知识：继承的基本语法
2. 经验传授：继承中的5个注意事项
3. 编码技巧：创造『大整数』类型
4. 一课搞懂：继承权限

## 二、多重继承

1. 多重继承的基础知识
2. 菱形继承：困扰与解决方案
3. 初识：对象模型

# 一、继承基础

1. 基础知识：继承的基本语法
2. 经验传授：继承中的5个注意事项
3. 编码技巧：创造『大整数』类型
4. 一课搞懂：继承权限

# 总览

封装：我该有的和我该做的

继承：叫一声爸爸，开启财富之门

多态：我就是我，是不一样的烟火

# 继承的基础语法

```
class A : public Base
```

# 继承的基础语法

```
class A : public Base
```

A类公有继承自Base类

# 继承的基础语法

```
class A : public Base
```

A类以 Base 类为基础，做功能拓展



# 继承的基础语法

```
class A : public Base
```

A类：派生类，Base类：基类

# 一、继承基础

1. 基础知识：继承的基本语法
2. 经验传授：继承中的5个注意事项
3. 编码技巧：创造『大整数』类型
4. 一课搞懂：继承权限

## 2-1. 继承以后的子类大小

# 结构体-对齐补齐规则

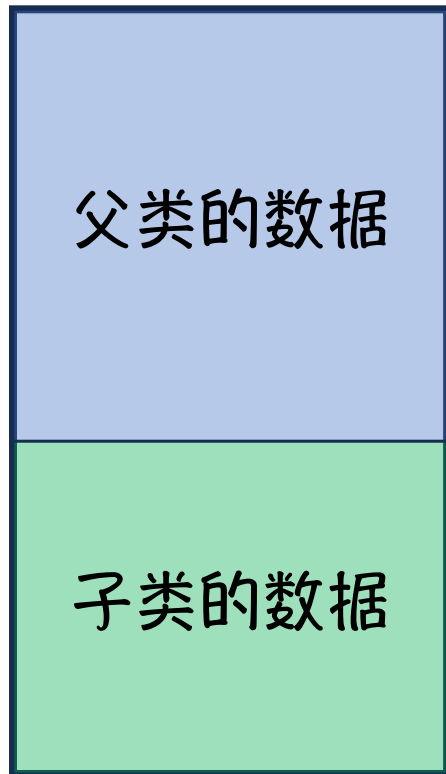
1. 类型都有一个对齐值，内建类型的对齐值等于其自身大小
2. 结构体的对齐值，等于其成员中的最大对齐值
3. 成员被存储在其整数倍的对齐值位置上
4. 可以通过 `#pragma pack` 限制对齐值的最大值

```
1 struct person {  
2     char name[20]; // 姓名  
3     int age;       // 年龄  
4     char gender;   // 性别  
5     float height;  // 身高  
6 };
```

《船说：C语言全能实战课》第6章-第18节：『对齐规则』全解析

# 子类对象的大小

## 子类对象



1. 子类中存储了子类独有的数据以及从父类中继承来的数据
2. 子类的对齐值受父类对齐值影响
3. 简单继承场景中：父类的数据在前，子类数据在后

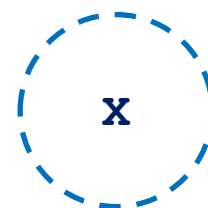
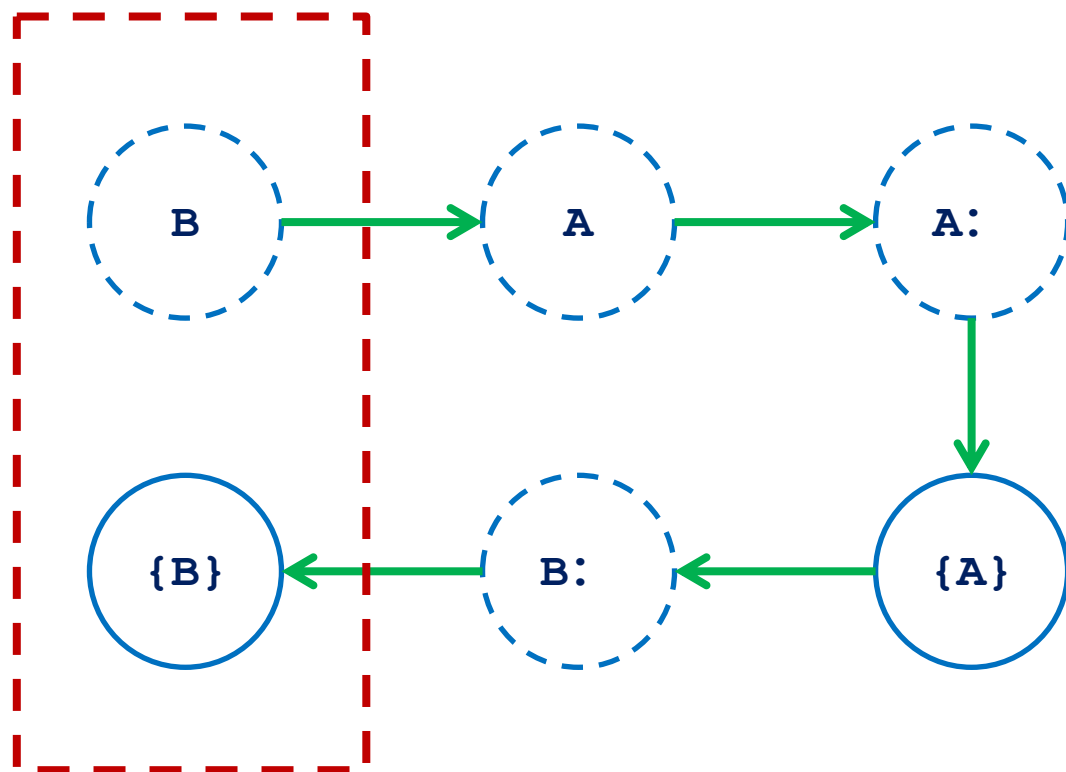
## 2-2. 父类的指针与引用指向子类对象

## 2-3. 初探子类数据的存储结构

## 2-4. 子类与父类的构造与析构顺序



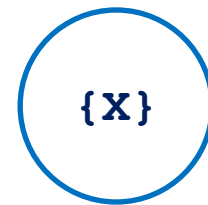
# 基类与派生类的构造顺序



开辟 x 的数据区



执行初始化列表



执行构造逻辑

## 2-5. 继承中的构造、拷贝与赋值

# 一、继承基础

1. 基础知识：继承的基本语法
2. 经验传授：继承中的5个注意事项
3. 编码技巧：创造『大整数』类型
4. 一课搞懂：继承权限

# 为什么会有『大整数』

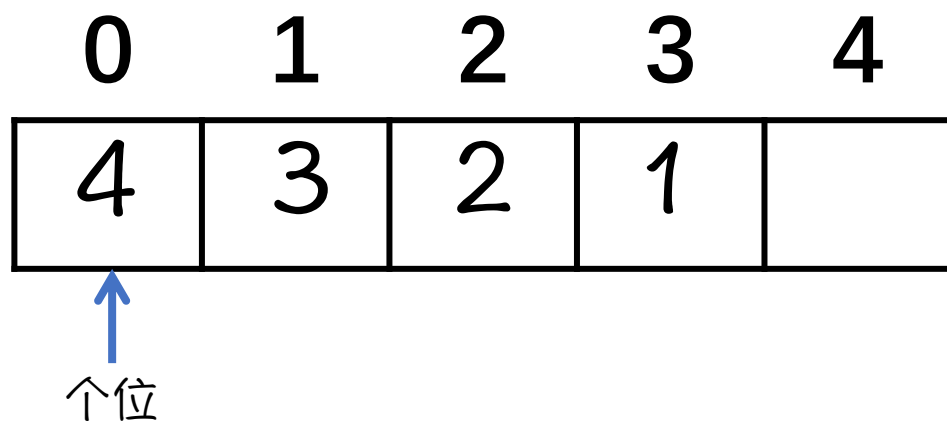
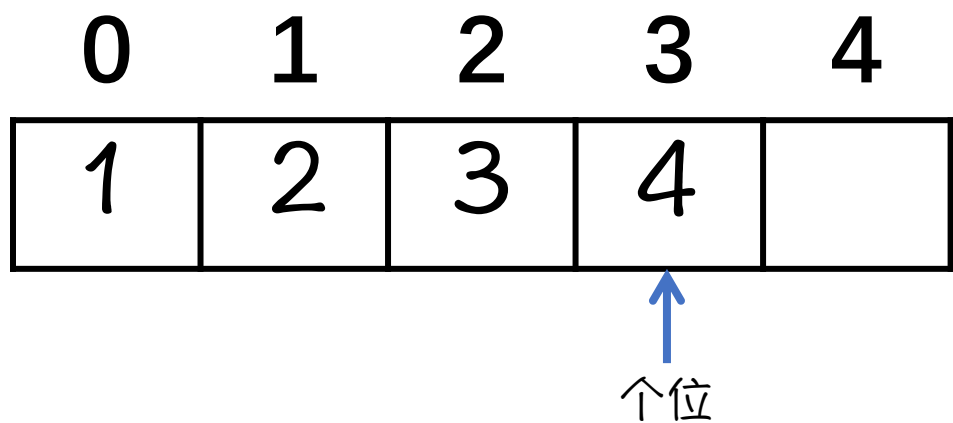
int:  $2^{31} - 1 = 2147483647$

long long:  $2^{63} - 1 = 9223372036854775807$

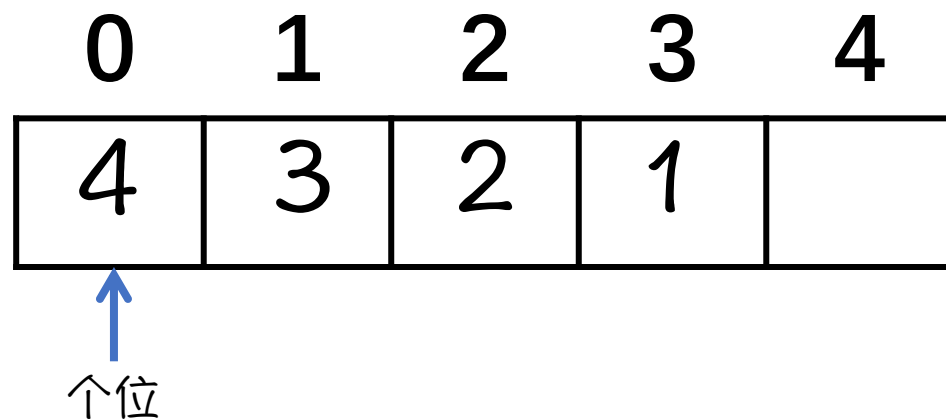
# 怎么存储『大整数』

1 2 3 4

# 怎么存储『大整数』



# 怎么存储『大整数』



# 『大整数』 加法运算

$$997 + 45 = ?$$



# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

--	--	--	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

12	13	9		
----	----	---	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

12	13	9		
----	----	---	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	14	9		
---	----	---	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	14	9		
---	----	---	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	4	10		
---	---	----	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	4	10		
---	---	----	--	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	4	0	1	
---	---	---	---	--



# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = ?$$

---

2	4	0	1	
---	---	---	---	--

# 『大整数』 加法运算

0	1	2	3	4
7	9	9		
5	4			

$$997 + 45 = 1042$$

---

2	4	0	1	
---	---	---	---	--

# 创造『大整数』类型

HZ0J-38：兔子繁殖问题

# 一、继承基础

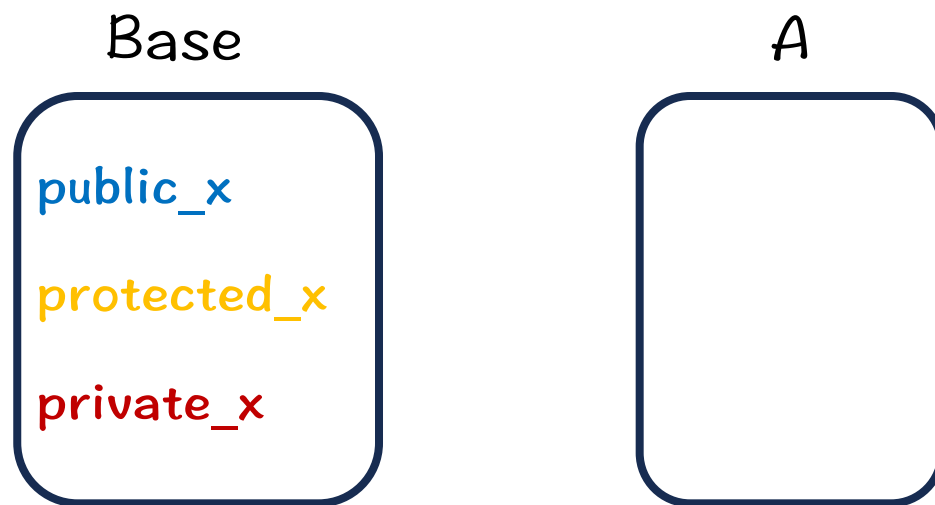
1. 基础知识：继承的基本语法
2. 经验传授：继承中的5个注意事项
3. 编码技巧：创造『大整数』类型
4. 一课搞懂：继承权限

# 【继承权限】影响的究竟是什么？

是否影响子类对于父类中方法和属性的访问？

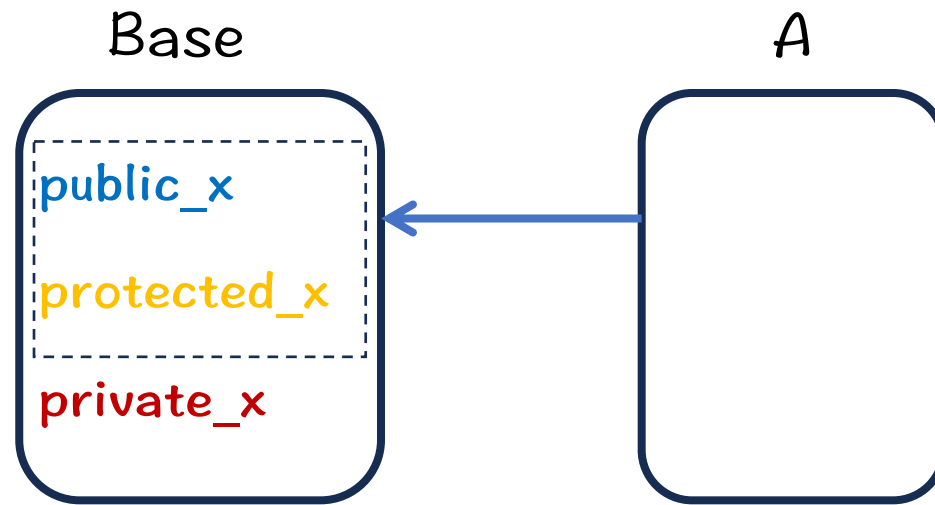
# 【继承权限】影响的究竟是什么？

是否影响子类对于父类中方法和属性的访问？

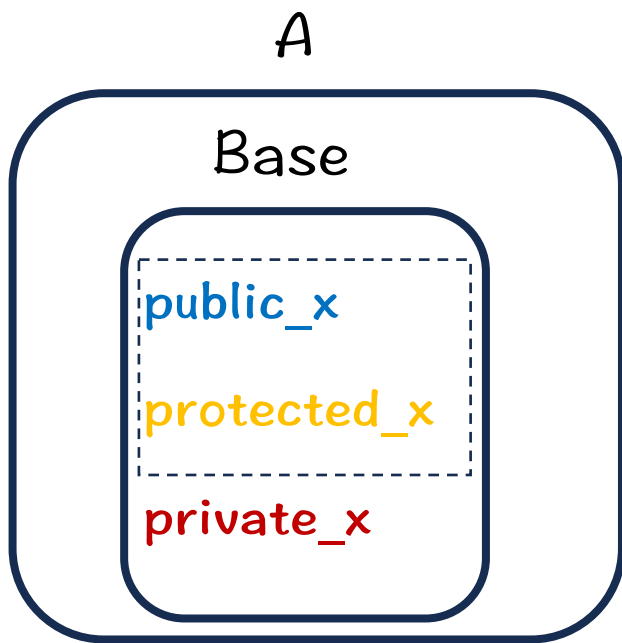


# 【继承权限】影响的究竟是什么？

是否影响子类对于父类中方法和属性的访问？

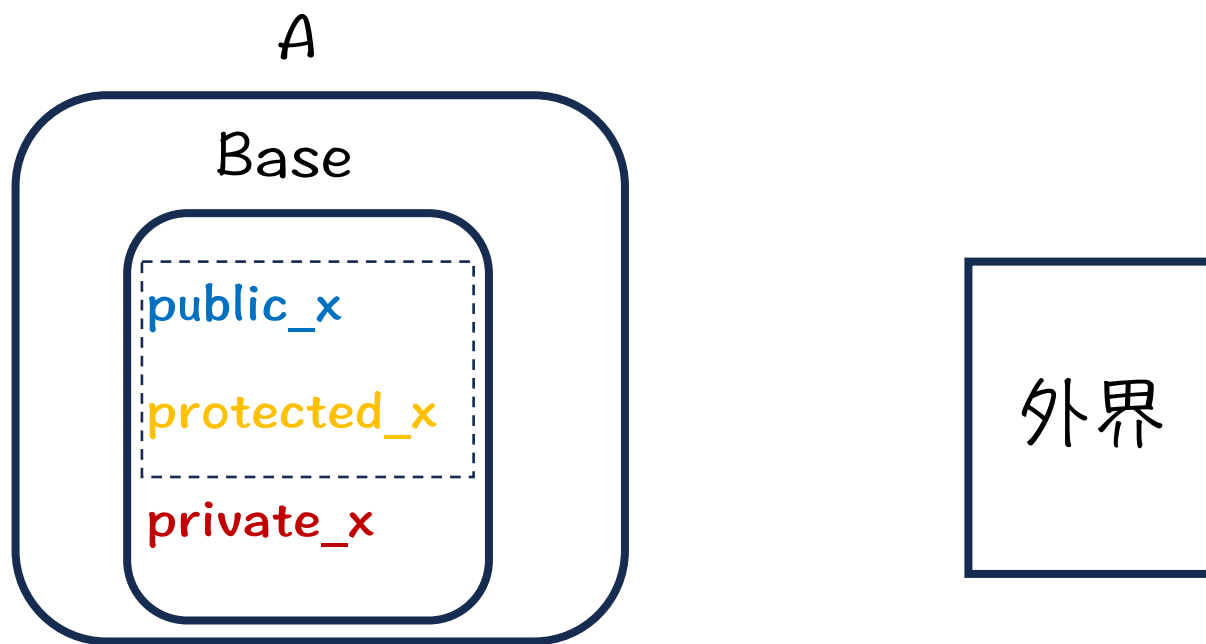


# 【继承权限】影响的究竟是什么？

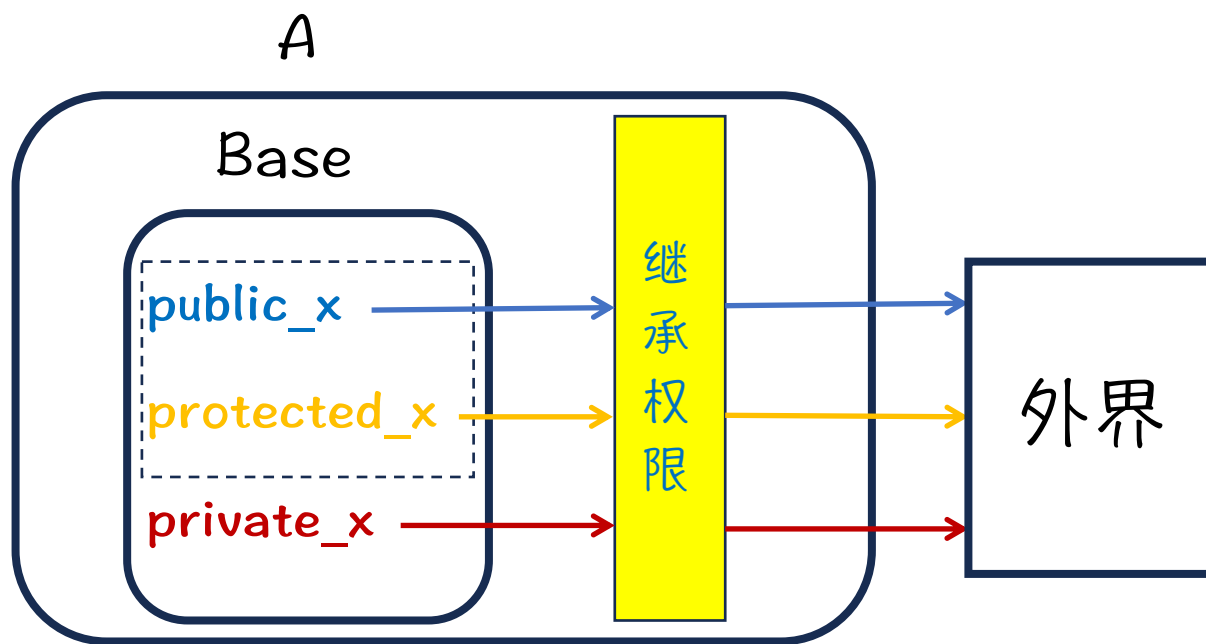




# 【继承权限】影响的究竟是什么？



# 【继承权限】影响的究竟是什么？



# 继承-子类的访问权限

继承权限 

原有权限

	public	protected	private
public	✓	✓	✓
protected	✓	✓	✓
private	✗	✗	✗

# 继承-对外的访问权限

继承权限 

原  
有  
权  
限

	public	protected	private
public	public	protected	private
protected	protected	protected	private
private	private	private	private

# 【继承权限】影响的究竟是什么？

总结：影响子类继承自父类的属性和方法的对外访问权限

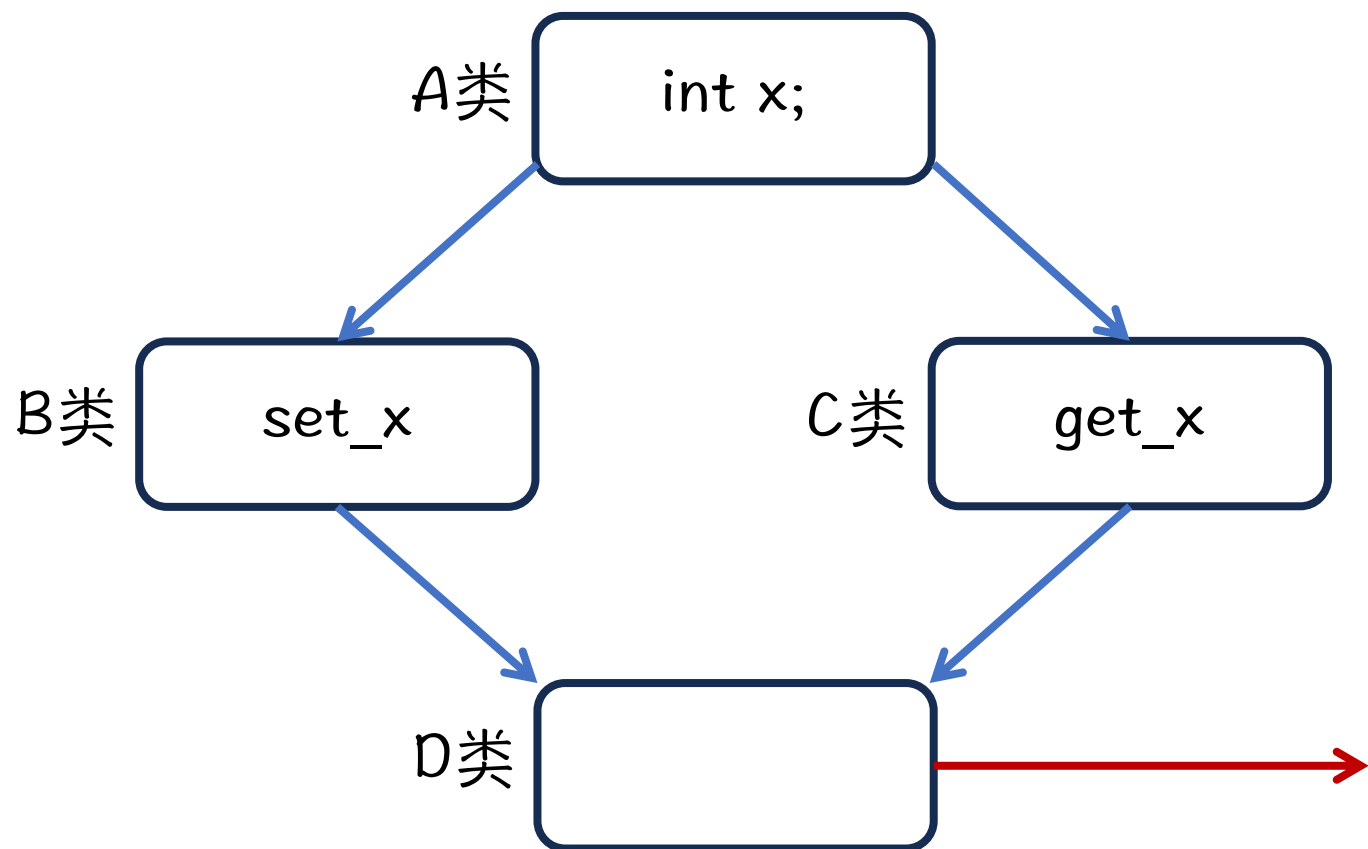
## 二、多重继承

1. 多重继承的基础知识
2. 菱形继承：困扰与解决方案
3. 初识：对象模型

## 二、多重继承

1. 多重继承的基础知识
2. 菱形继承：困扰与解决方案
3. 初识：对象模型

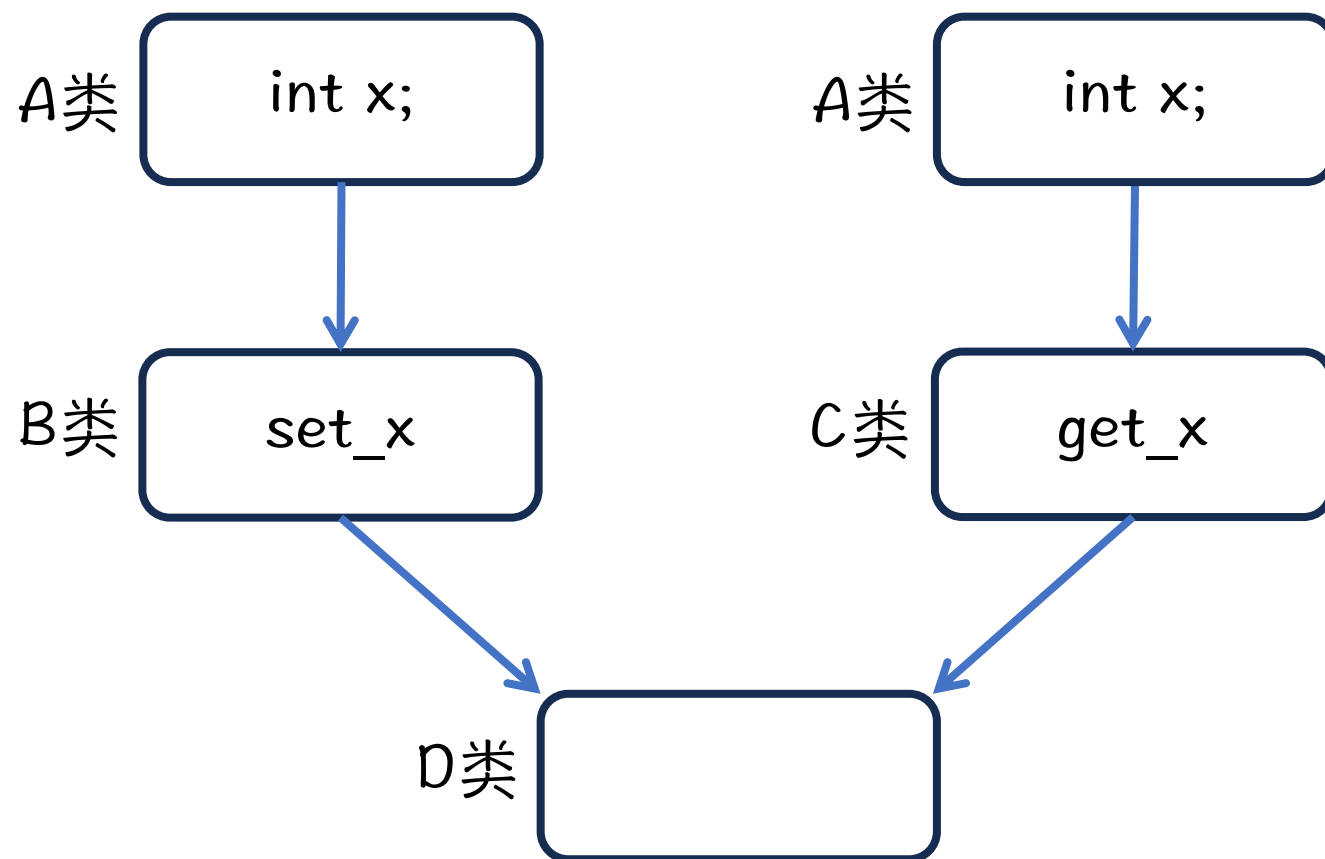
# 菱形继承：你想象中的继承



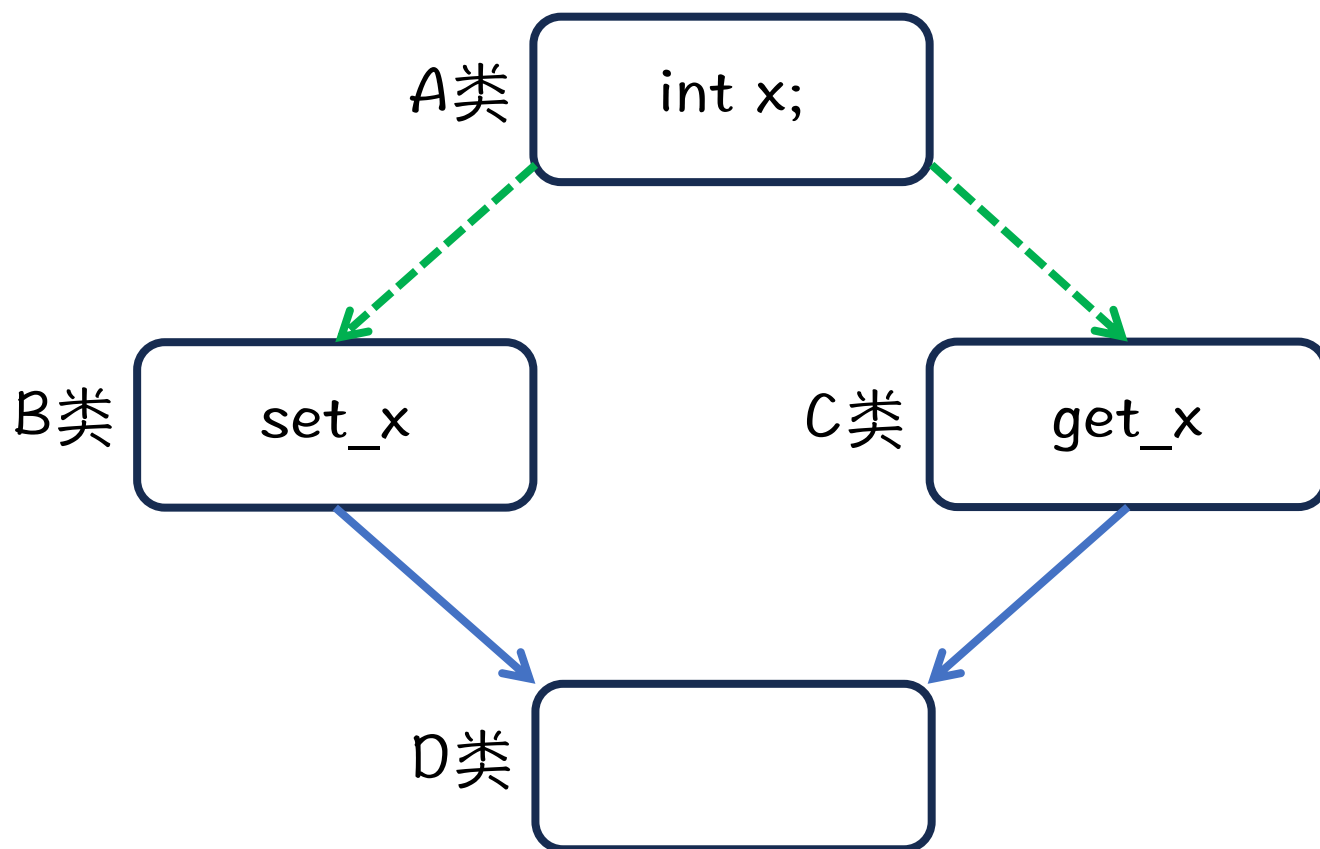
```
D obj;  
obj.set_x(3);  
obj.get_x(); ???
```



# 菱形继承：实际的继承



# 菱形继承：虚继承



## 二、多重继承

1. 多重继承的基础知识
2. 菱形继承：困扰与解决方案
3. 初识：对象模型

# 初识：对象模型

什么是对象模型？

# 结构体-对齐补齐规则

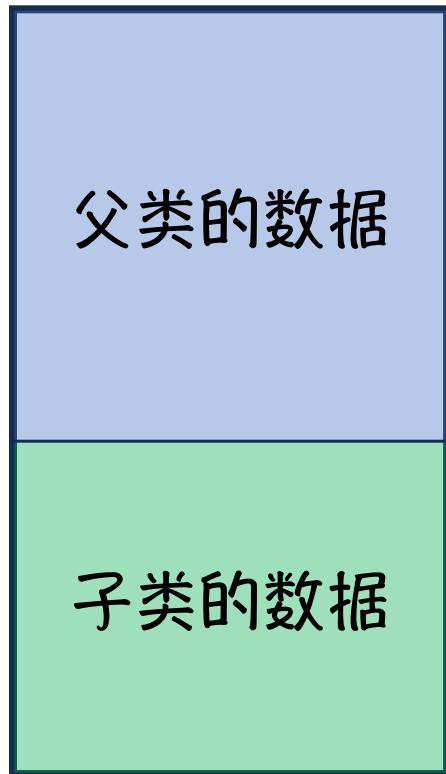
1. 类型都有一个对齐值，内建类型的对齐值等于其自身大小
2. 结构体的对齐值，等于其成员中的最大对齐值
3. 成员被存储在其整数倍的对齐值位置上
4. 可以通过 `#pragma pack` 限制对齐值的最大值

```
1 struct person {  
2     char name[20]; // 姓名  
3     int age;       // 年龄  
4     char gender;   // 性别  
5     float height;  // 身高  
6 };
```

《船说：C语言全能实战课》第6章-第18节：『对齐规则』全解析

# 子类对象的大小

## 子类对象

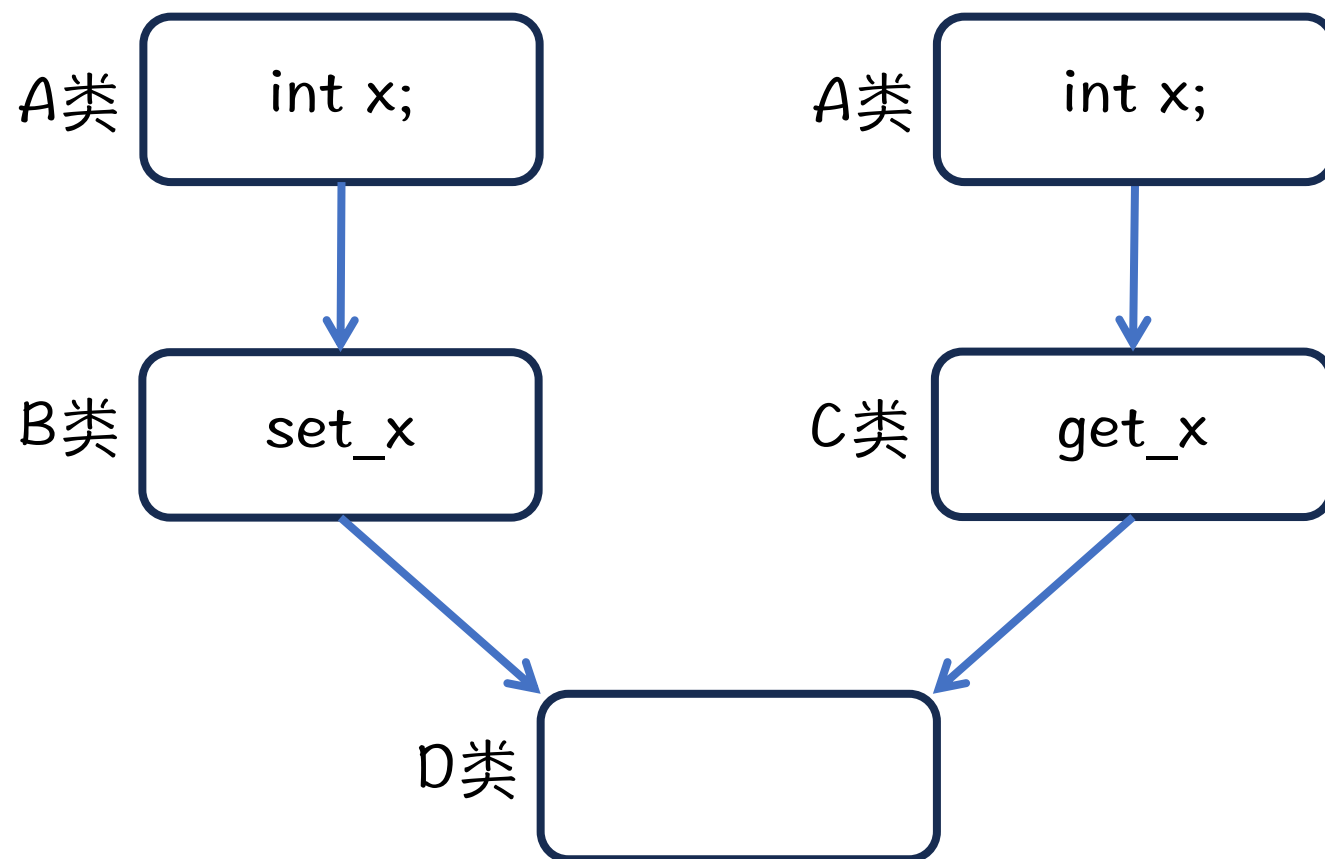


1. 子类中存储了子类独有的数据以及从父类中继承来的数据
2. 子类的对齐值受父类对齐值影响
3. 简单继承场景中：父类的数据在前，子类数据在后

# 初识：对象模型

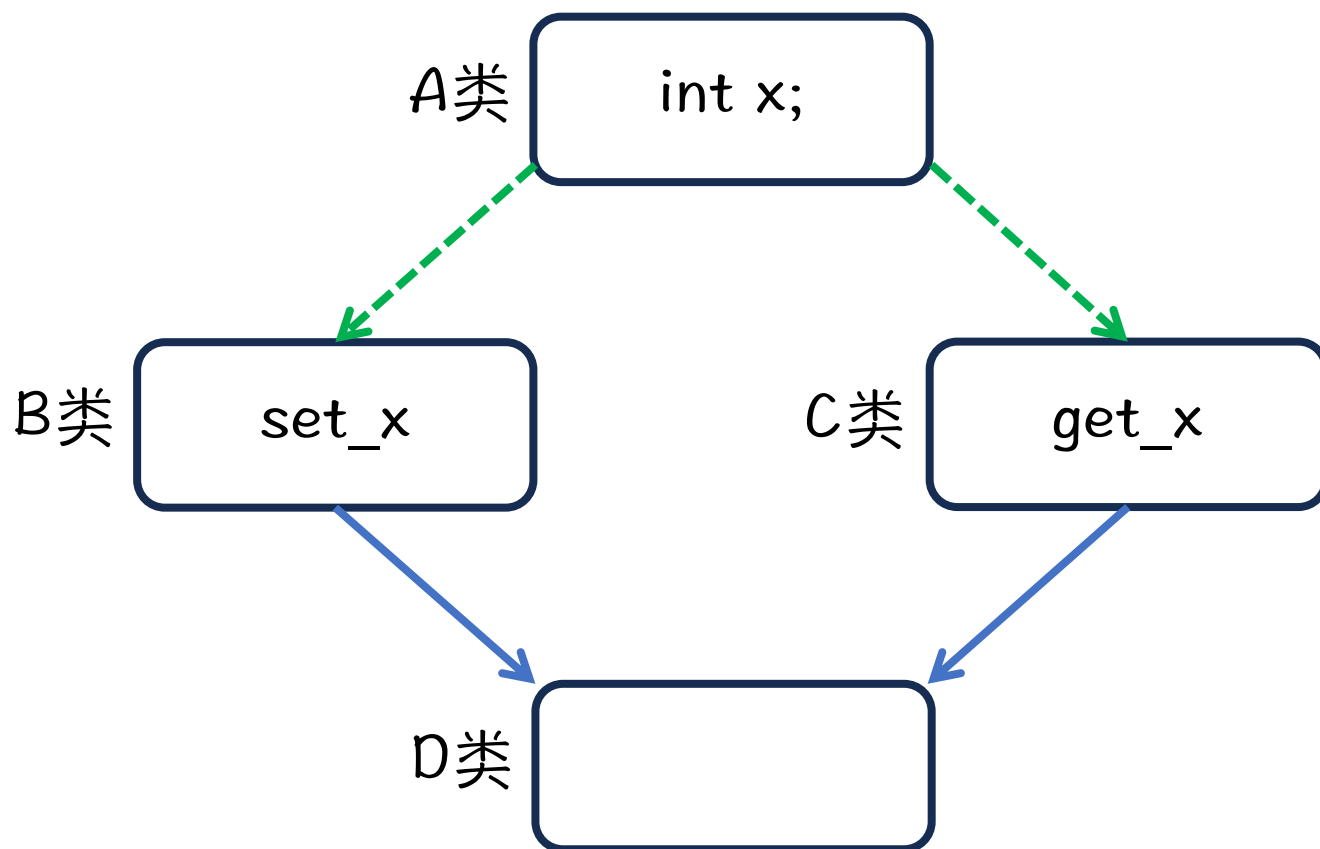
## 探索环境中【虚继承】的对象模型

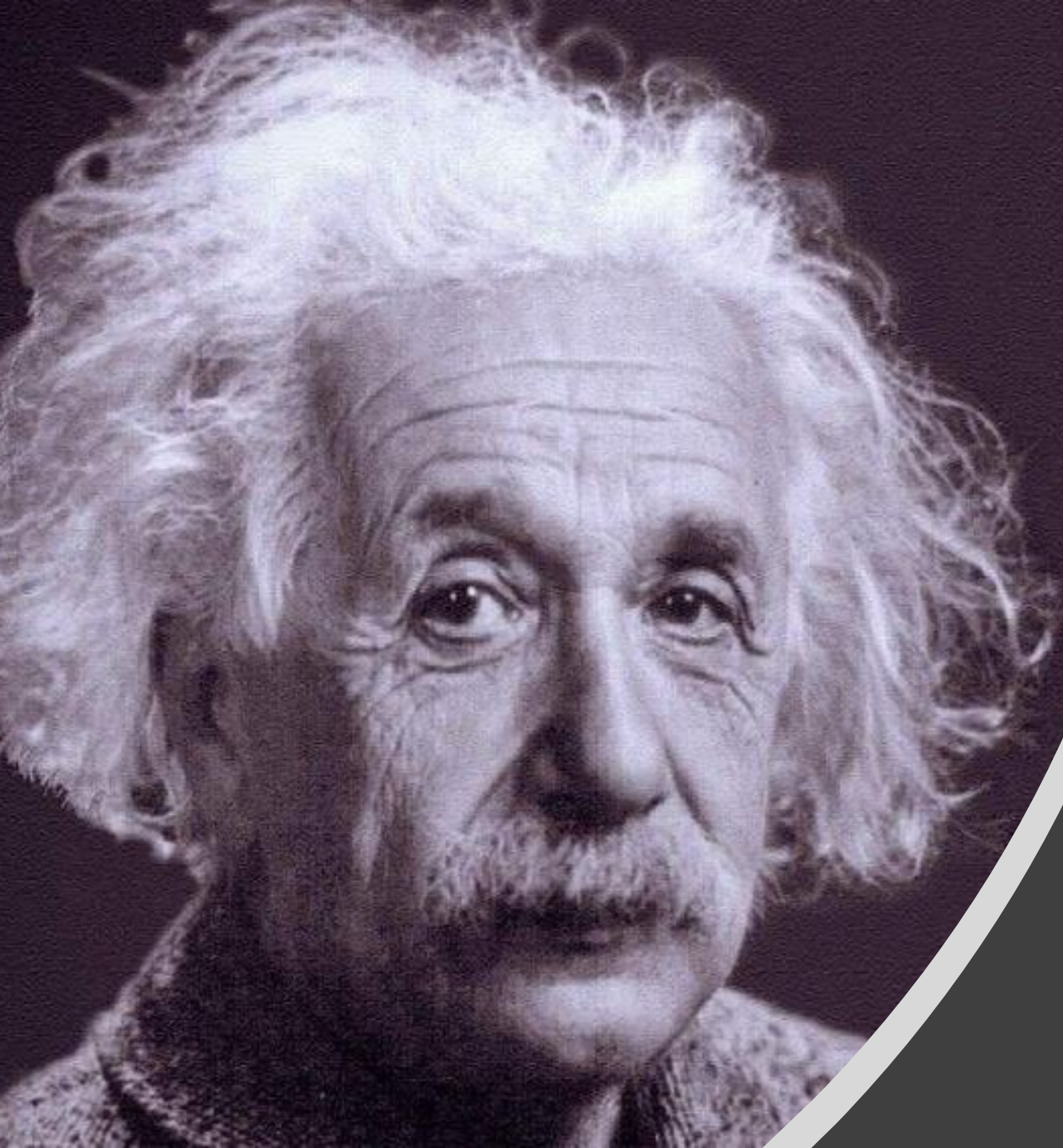
# 菱形继承：实际的继承





# 菱形继承：虚继承





为什么  
会出一样的题目？