

Ch.1 C/C++的特性

二

2017 / 10

类与对象

- struct 和class
- 继承与virtual继承

virtual继承

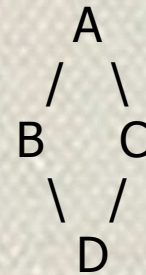
```
void A :: func();
```

```
B : public A
```

```
C : public A
```

```
D : public B, public C
```

二义性



继承的可见性

继承方式/可见性	public	protected	private
public	public	protected	/
protected	protected	protected	/
private	private	private	/

类

- 空类
- 构造、拷贝构造、析构函数
- 初始化列表，比`const`初始化还要早
- `This` 指针

运算符重载

- 其实可以用函数代替
- 重载后符合思维习惯与使用习惯

运算符重载

- 哪些运算符可重载？
 - 大部分的操作符是可以被重载的，例外的只有“.”、“::”、“?:”和“sizeof”
- new/delete 是运算符，可重载，重载的意义
- operator new 函数

函数对象

- 类中重载括号（）
- 类的对象（或临时对象）调用（）运算符
- 跟使用函数很像

重载new/delete的意义

- 可以改变内存的分配
- 特殊用途，比如？

顺便说一下

- 函数指针
- 是一个指向函数的指针
- 指向不同的函数，调用行为发生变化

模板与泛型编程

- 泛化是默认（通用）的处理
- 特化是特殊类型的处理

异常处理

- 数组累加和函数
- 限定条件是元素范围 $[-100, 100]$

异常处理

- 为什么要有异常处理？
- C++可抛出任何类型异常
- 不能把正常的判断用异常的处理方式进行

关键字

- 基本数据类型 char, int, long, short, float, double, signed, unsigned, void, sizeof
- 复合类型 struct, union, enum
- 定义类型 typedef
- 修饰符 auto, const, inline, register, extern, static, volatile
- 流程 if-else, switch, case, default, while, do-while, for, break, continue, return, goto
- 面向对象 class, private, public, protected, this, new, delete, virtual, operator, friend

关键字

- 异常处理 `throw, try-catch`
- 模板 `template`
- 命名空间 `namespace, using`
- 逻辑类型 `bool, true, false`
- 修饰符 `explicit, mutable`
- 运行时刻类型信息 `typeid, typename`
- 强制类型转换 `const_cast, static_cast, dynamic_cast, reinterpret_cast`

Q & A