

c++ 总结

W.J.Z

1 变量和基本类型

1、基本内置类型：

C++ 基本数据类型为算术类型 (布尔、字符、整型、浮点型) 和空类型。

| 类型 | 最小尺寸 |
|-------------|----------|
| bool | 未知 |
| char | 8 位 |
| wchar_t | 16 位 |
| char16_t | 16 位 |
| char32_t | 32 位 |
| int | 16 位 |
| long | 16 位 |
| long long | 32 位 |
| float | 6 位有效数字 |
| double | 10 位有效数字 |
| long double | 10 位有效数字 |

技巧 1: 执行浮点数运算选 double

技巧 2: 数值不为负时选 unsigned

技巧 3: 整型运算选择 int 和 long long

2、变量声明与定义

变量初始化时创建变量并赋予一个初始值，赋值为擦除变量当前值赋予一个新值，推荐初始化使用 `int sum(0);` 方式。

为了配合 c++ 分离式编译，使变量为整个程序所知，使用 `extern` 关键字对变量进行声明。变量只能被定义一次，但可以被多次声明。

3、引用和指针的区别

引用为引用对象创建了一个别名，指针则是创建了对象。指针无需在创建时进行赋初值，引用必须制定引用对象。指针可以在整个生命周期指向不同的对象，而应用有且只有一个。

```
int *p = &a; %复合类型写法一
int* p = &a; %复合类型写法二
const int a =10; %a 值不能改变
int *const p = &b;%不能改变p的值
%*放在const 前面说明指针是一个常量
const int* p = &a;%a 值不能改变
const int* const p = &a;%a和p 值都不能改变
```

4、头文件

`#define` 定义预处理变量, `ifdef ifndef` 检查预处理变量是否定义, 如果检查为真, 执行后续操作直到遇见 `#endif`。

2 面向对象的基本特性

1. 类和对象的申明
2. 构造函数
3. 析构函数
4. 内联成员函数
5. 复制构造函数
6. 类的组合
7. 结构体
8. 联合体
9. 枚举类

3 数据共享与保护

1. 静态数据成员
2. 静态函数成员
3. 友元函数和友元类
4. 常对象、常成员、常引用
5. 多文件结构
6. 预编译命令

4 继承和派生

1. 单继承、多继承、继承方式
2. 基类与派生类转换

5 面向对象编程

对于某些函数，基类希望它的派生类各自定义适合自身的版本，此时基类就将这些函数声明成虚函数，派生类必须在其内部对其所有重新定义的虚函数进行声明，任何构造函数之外的非静态函数函数都可以是虚函数，如果基类把一个函数声明为虚函数，则该函数在派生类中隐式地也是虚函数，派生类对象不能直接初始化从基类继承而来的成员 (每个类控制自己成员的初始化过程)，首先初始化基类的部分，然后按照声明的顺序依次初始化派生类的成员。

如果基类定义了一个静态成员，则在整个继承体系中只存在该成员的唯一定义。C++11 新标准中可以使用 `override` 关键字来说明派生类中的虚函数，将共同基类覆盖这位虚基类，这时从不同的路径继承过来的同名数据成员在内存中就有一个副本，同一个函数名也只有一个映射。