C++代码

[基础知识： 2](#_Toc21450)

[函数func： 3](#_Toc4935)

[结构体struct： 3](#_Toc3300)

[对象class： 3](#_Toc4995)

[命名空间namespace： 4](#_Toc4658)

[头文件： 4](#_Toc22448)

[原生cstdlib： 4](#_Toc5895)

[时间ctime： 4](#_Toc27711)

[文件fstream： 5](#_Toc10958)

[交互iostream： 5](#_Toc3242)

[矢量vector： 5](#_Toc17903)

**基础知识：**

|  |  |  |
| --- | --- | --- |
| **内存分区** | **代码区** | **二进制代码** |
| **全局区** | **全局变量，静态变量 (static)，常量 (const)** |
| **栈区** | **函数参数，局部变量** |
| **堆区** | **自由分配、释放 (new)** |

|  |  |  |
| --- | --- | --- |
| **注释** | **// String** | **单行** |
| **/\* String \*/** | **多行** |
| **声明** | **# define <Key> <Value>** | **定义宏常量 (只进行字符替换)** |
| **extern <Type> <Key>** | **声明引用** |
| **类型** | **typeid(Key).name()** | **类型名称** |
| **typedef <oldType> <newType>** | **类型重命名** |
| **内存** | **sizeof(Type | Key)** | **返回占用字节** |
| **delete[] <Key>** | **释放数组 (需要检查指针为空)** |
| **系统** | **system(cmd)** | **输入命令，输出响应** |

**常量指针 (指向值不变)：const int\* p;**

**指针常量 (指向地址不变)：int\* const p;**

**指针声明：int \*p = &a;**

**变量别名：int &b = a;**

|  |  |
| --- | --- |
| **分支循环：** | |
| **<Bool> ? <true\_out> : <false\_out>** | **条件赋值，返回变量 (三目运算符)** |
| **while、if、else if、else** | **分支常用语句** |
| **do {cmds} while (Bool)** |
| **switch <Exp> {**  **case <Res1>: ...**  **case <Res2>: ...**  **default: ...**  **}** | **满足激活条件时执行余下所有语句** |
| **for (<init>; <cond>; <step>)** | **变量循环获得序列里的值** |
| **for (auto <Key>: <Seq>)** |
| **break** | **跳出循环体** |
| **continue** | **返回循环语句** |

|  |  |  |  |
| --- | --- | --- | --- |
| **转义序列：** | | | |
| **振铃字符\a** | **换行字符\n** | **垂直制表符\v** | **退格字符\b** |
| **水平制表符 (以4个字符为一列)\t** | | | **回车字符\r** |

|  |  |  |  |
| --- | --- | --- | --- |
| **运算优先级：** | | | |
| **作用域解析** | **::** | | |
| **成员访问 (有序)** | **. (直接)，-> (间接)** | | |
| **地址** | **\* (解引用；先左结合)，& (取地址)** | | |
| **位翻转** | **~** | | |
| **乘除** | **\*，/，% (int)** | | |
| **加减** | **+，-** | **b=++a** | **a自增，返回结果** |
| **b=a++** | **返回结果，a自增** |
| **位移动** | **<<，>>** | | |
| **位运算 (有序)** | **&，^，|** | | |
| **比较操作符** | **>，>=，<，<=，==，!=** | | |
| **逻辑运算符** | **!，&&，||** | | |

**# -：改符号位，并取补码**

**数组 (首地址)：<Type> <Key>[] = {x1, x2, ...};**

**类型转换：<Type> (<Key>)**

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型：** | | | |
| **修饰** | **signed (带符号)** | **unsigned (无符号)** |  |
| **整型** | **int16\_t (short)** | **int32\_t (int)** | **int64\_t (long long)** |
| **浮点** | **float (4 byte)** | **double (8 byte)** |  |
| **字符** | **char (1 byte)** | **wchar\_t (2 byte)** |  |
| **其它** | **bool (1 byte)** | **void (空类型)** | **指针 (8 byte)** |
| **自动** | **auto (需赋值)** | **decltype(exp)** |  |

**函数func：**

**返回值不可是局部变量的引用**

**函数重载：返回值相同，形参不同**

**# define <Func>(x, y) (x+y)**

**# define <Func>(x, y) (x=y)**

**void <Func>(const <Type> <Key>) // 静态形参不可修改**

**结构体struct：**

|  |  |
| --- | --- |
| **定义** | **struct <Type> {...}** |
| **创建** | **<Type> <Key> = {m1, m2}** |
| **<Type> <Key>[ ] = {{ }, { }, ...}** |

**对象class：**

|  |  |  |  |
| --- | --- | --- | --- |
| **static** | **成员变量** | **static修饰 (需类外初始化)** | **<rType> <Type>::<attr> = <value>** |
| **成员方法** | **static修饰函数返回值，只能访问静态成员变量** | |
| **const** | **对象** | **只能调用常方法** | |
| **成员方法** | **只修改mutable修饰的成员变量** | **<rType> <Func>(\*args) const** |
| **virtual** | **成员方法** | **虚函数，可被基类访问，覆写时在参数后加override** | |
| **抽象方法** | **virtual <rType> <Func>(\*args) = 0;** | |
| **初始化** | **<Type> <Key>(\*args)** | | |
| **<Type> <Key> = <Type>(\*args)** | | |
| **<Type> <Key> = {\*args}** | | |
| **成员方法** | **自身的指针常量this** | | |
| **访问权限** | **private** | **私有 (默认)，不可继承，可被友元访问** | |
| **protected** | **保护，可继承，可被友元访问** | |
| **public** | **公有** | |
| **继承方式** | **class T: private B** | **私有继承，(protected, public) → private** | |
| **class T: protected B** | **保护继承，public → protected** | |
| **class T: public B** | **公有继承** | |
| **class T: virtual B** | **虚继承，成员变量可重载** | |
| **友元声明** | **friend <rType> <Func>(\*args)** | | |
| **friend class <Type>** | | |

**可在全局重载函数**

|  |  |  |
| --- | --- | --- |
| **基本方法** | **<Type>(\*args): <attr>(value), { }** | **构造函数** |
| **<Type>(const <Type>& <Key>){ }** | **拷贝函数，默认浅拷贝** |
| **~<Type>(\*args){ }** | **析构函数，在此释放堆区数据** |
| **<rType> operator()(\*args)** | **仿函数** |
| **friend ostream& operator<<**  **(ostream& os, const <Type>& <Key>)** | **流左移函数** |
| **比较方法** | **bool operator==(<Type>& other)** | **==, !=, <, <=, >, >=** |
| **运算方法** | **<Type> operator+(<dType>& other)** | **+, -, \*, /, %, <<, >>** |
| **<Type> operator++(int)** | **自增 (x++)，return <new>** |
| **<Type>& operator++()** | **自增 (++x)，return \*this** |
| **<Type>& operator=(<Type> const& other)** | **赋值，return \*this** |

**模板template：**

|  |  |
| --- | --- |
| **定义** | **template<class T>** |
| **函数** | **fun<T>(\*args)** |

**命名空间namespace：**

|  |  |  |
| --- | --- | --- |
| **定义** | **namespace <ns> {...}** | |
| **调用** | **using namespace <ns>;** | **导入所有变量** |
| **using <ns>::<key>;** | **导入某一变量** |

**头文件：**

**#ifndef \_<FILE>\_H**

**#define \_<FILE>\_H**

**// 程序段**

**#endif**

**#pragma once**

**// 程序段**

**时间ctime：**

|  |  |
| --- | --- |
| **时间操作：** | |
| **time(&time\_t)** | **获得当前秒数** |
| **\_sleep(ms)** | **睡眠指定时间** |

|  |  |  |
| --- | --- | --- |
| **时间元组\*tm：** | | |
| **tm\_year** | **%Y** | **2001** |
| **tm\_mon** | **%m** | **01~12** |
| **%b** | **Jan ~ Dec** |
| **%B** | **January ~ December** |
| **tm\_mday** | **%d** | **1~31** |
| **tm\_hour** | **%H** | **0~23** |
| **tm\_min** | **%M** | **0~59** |
| **tm\_sec** | **%S** | **0~61** |
| **tm\_wday** | **%a** | **Mon ~ Sun (0 ~ 6)** |
| **%A** | **Monday ~ Sunday** |
| **tm\_yday** | **年中第几天** | **1 ~ 366** |
| **tm\_isdst** | **是否夏令时** | **0,1,-1 (代表夏令时)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **时间转换：** | | | |
| **秒数**  **long long** | **ctime(&seconds)** | **秒数 → 字符串** | **%a %b %d %H:%M:%S %Y** |
| **gmtime(&seconds)** | **秒数 → 时间元组** | **格林威治时间** |
| **localtime(&seconds)** | **当地时间** |
| **时间元组**  **\*tm** | **mktime(t\_tuple)** | **时间元组 → 秒数** | |
| **strftime(char[], size, format, t\_tuple)** | **时间元组 → 字符串** | |

**文件fstream：**

|  |  |  |
| --- | --- | --- |
| **ios** | **默认模式** | **in 只读模式** |
| **写入模式**  **(若文件存在)** | **out 覆盖** |
| **app 追加写入** |
| **特殊模式** | **binary 二进制模式** |

|  |  |  |  |
| --- | --- | --- | --- |
| **std** | **fstream(file, mode)** | | **以写入模式打开文件** |
| **实例方法** | **<< String** | **写入数据** |
| **>> String** | **读出单行** |
| **read(string, size)** | **读出定长数据** |
| **is\_open()** | **打开状态** |
| **close()** | **关闭文件** |

**IO流iostream：**

|  |  |  |  |
| --- | --- | --- | --- |
| **std** | **常量** | **endl** | **换行符** |
| **IO** | **cout << String** | **输出字符串** |
| **cerr << String** | **输出报错信息** |
| **cin >> String** | **读取输入** |

**随机random：**

**字符串string：**

|  |  |  |
| --- | --- | --- |
| **std** | **getline(istream, string)** | **读出单行 (无数据时返回false)** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **std** | **string <Key> = <Value>** | | **声明字符串** | **赋值** |
| **string <Key>(Value)** | | **副本** |
| **string <Key>(repeat, Char)** | | **重复** |
| **实例方法** | **size()** | **返回长度** | |

**矢量vector：**

|  |  |  |  |
| --- | --- | --- | --- |
| **std** | **vector<Type> <Key>(len\_or\_arr, ele\_if\_arr)** | | **换行符** |
| **实例方法** | **size()** | **返回元素个数** |
| **front() / back()** | **首/尾元素** |
| **push\_back() / pop\_back()** | **尾部添加/删除元素** |
| **insert(pos\_iter, n=1, x)** | **插入元素** |
| **erase(beg\_iter, end\_iter=NULL)** | **删除元素** |
| **begin() / end() / rbegin() / rend()** | **指定位置迭代器** |