|  |  |
| --- | --- |
| 头文件 | 函数方法、模板类 |
| iostream | 包含函数cin cout |
| iomanip | 包含控制符setprecision() setfill() setw()控制设置精度、填充字符、字符宽度 |
| fstream | 包含open()等关于文件的函数工具 |
| cctype | 包含转换字母大消息转换等工具 |
| complex | complex 复数工具及运算 |
| valarray | valarray 数值数组 面向数值计算  slice（1，4，3）下标指定 表示第1 4 7 10个数 |
| array | array 数值数组 代替内置数组 |
| vector | vector 数值数组 面向容器操作 |
| initializer\_list | initializer\_list该对象用于将一系列值传递给构造函数或其他函数，如求和、求平均  该模板类还包括begin() end() size()成员函数 |
| Stdlib.h/cstdlib.h | 包含abort（）、exit（）等异常相关函数 |
| Exception | 定义了异常类，可用作基类，有一个what()的虚方法 |
| Stdexcpt | 定义类logic\_error\runtime\_error异常类  Logic\_error 包含 domain\_error(不在定义域） invalid\_argument(传递意料之外的值) length\_error(没有足够的空间) out\_of\_bounds(指示索引错误)  Runtime\_error 包含range\_error overflow\_error underflow\_error  Logic\_error表明可以用编程修复的， Runtime\_error表明存在无法避免的问题 |
| New | 包含bad\_alloc类 从exception公有派生而来，返回由new引发的异常 |
| Typeinfo | 包含bad\_cast异常，由关键词dynamic\_cast引发  包含bad\_typeid异常，由关键词typeid引发 |

### 头文件与模板类

基础数据类型

1.iostream文件包含cout cin函数

2.需要using namespace std命名空间

3.或者包含iostream.h或者使用std::cin >> 或者 std::cout <<

数据类型

1.整型

int 4字节

short 2字节

long 4字节

long long 8字节

2.无符号整形

unsigned int/short/long/long long

\*cout默认10进制输出显示， 如果要其他方式，在显示之前加上 cout<<hex 或者 cout<<oct

3.字符与小整数

bool 1位

char 1字节

wchar\_t 宽字符类型

char16\_t char32\_t 新增类型U'xwy' 表示32类型 u'xwy' 表示16常量

\*限定符const 声明常量 名称常用首字母大写或全部大写

4.浮点数

float 32位

double 至少48位 常为64位

long double 常为80 96 128

\*cout只能显示6位数！！！例如a = 123.456789 只显示123.456

5.

运算符 + - \* / % 优先级相同的时候，乘除从左向右

文件 iostream climits //该文件可用于自动防止超出整型限制

类 istream ostream

成员函数cin cout cout.put();

//release 2.0之前，cout<<'M'无法输出字符常量，为了输出字符常量而建立的，现已修改

复合数据类型

1.数组（与C几乎一致）

重点char test[12] = {} //可将所有元素置0

2.字符串

C风格字符串，末尾必为空字符串\0

cin 为面向单词的输入方式

cin.get().get() //常用用法

cin.getline(name,20) 为面向行的输入方式

重点：输入字符串时cin会把换行符留在输入队列中，get也是。而getline()则用空字符来代替换行符。恢复输入使用cin.clear()

3.string类

string a = {"aaa bbb ccc"};

字符串拼接直接使用

str = str1+str2;

其他操作

strcat(str1,str2) //拼接 安全方式strncat()

strcpy(str1,str2) //复制 安全方式strncpy()

strlen(str1) //返回字符值

想用cout输出string必须包含头文件string而不是string.h

4.结构体struct

5.共用体union

6.枚举enum

7.指针

\*一定要在对指针应用\*运算符之前，将指针初始化为一个确定的、适当的地址

函数重点

1.函数原型与调用

2.函数传入基本类型值

3.函数传入数组

4.函数传入二维数组

5.函数传入字符串

6.函数传入string

7.函数传入结构体

8.函数传入arry对象

\*重点：指针、const等在传入传出中的应用

1.函数指针（获取、声明、使用）

2.深入了解--函数指针数组，指向函数数组指针的指针

\*声明时可使用auto和typedef来简化输入量

\*深入函数

1.内联函数（inline 用于较短函数，类似define）

2.引用变量（用作函数形参）

数值小 值传递

数组 指针传递

结构体 指针或者引用

类对象 const引用

3.函数默认参数定义

4.函数重载（用于多个不同类型数据 基本相同功能）

5.函数模板（使用类型名来代替具体的类型int\double等 多个数据类型 相同算法）

6.重载的模板