

VS学习 (1)

删除链表记得加**free** () ;

3.

Scanf(“格式化”, 地址表) //第二个参数是变量地址, 不是变量

Printf() 数组名 或者变量名

数组名并不完全等于指针 不同函数的处理不一样。

4.

new一个指针 或者结构体之类的 要初始化比如tree中 不止初始化数据, 还要把left right设置成null;

5.

函数指针, 运用的时候出现可以使 &函数名或者定义一个函数指针指针, 再指向函数。

```
void(*oper1)(mytree::tree*);
```

```
oper1 = &(my1.oper);
```

```
my1.tree_dlr(a, &my1.oper);
```

```
my1.tree_ldr(a, oper1);
```

6.

```
file.Open(_T("userdata.dat"),CFile::modeRead,NULL);
```

开始的时候没有_T 就提示没有匹配的参数列表 还是两种字符串的问题

注意字符(串)有两种类型, char(*)和wchar_t(*)型, 分别用于**多字节字符**和**unicode**字符, 两种类型分别有一套对应的字符串处理函数(包括C/C++标准

库, WindowsAPI, MFC等)。MFC的类和函数会根据工程设置(默认为wchar_t)自动匹配, 因此这时Format要求的字符串参数为wchar_t*, 所以要在字符串前加上“L”表示是wchar_t型字符串, 或者加上_T(), _T同样会根据工程设置自动匹配char或wchar_t。

7.

未定义的基类错误

头文件在包含顺序上不要成闭合的环状, 他们结构顺序最好应该是树。

因为编译头文件的时候是有顺序的。

8.

全整数**除法**得到的依旧是整数

9.

重载: 给两个函数定义相同的文件名

重载运算符

```
Const 类 operator + (const 类& a1, const 类&a2) ;
```

重载的运算符不是类的成员函数 所以不能访问私有成员 所以需要取值和赋值函数

如果在类内重载函数, 则可以访问。

重载运算符的一个限制条件是至少一个操作数是类类型。

10.

构造函数没有返回值 `void`也不行

构造函数一般隐式的调用，显示调用也可以 对象=构造函数（参数）没有参数也得加括号，当无参构造函数的时候 不加括号

默认构造函数是当没有定义构造函数的时候系统自动添加的，但是如果定义了一个带参数的构造函数，那么就没有默认的构造函数，如果定义一个对象 没有参数则错误 所以可以加一个空的不带参数的构造函数

11.

Static

一个类中所有对象共享的一个变量成为 **静态成员**

遇到的问题有 &错误 将类中的函数前加个 `static`就可以了。

经常遇到函数形参与实参不一致 有时候在自己定义的函数那里加一个 `static` 就可以 因为有的要求函数作为参数时是静态的

如果一个函数不访问类中的任何对象的数据，但却希望此函数作为类的成员。可以将函数声明为静态成员函数，可以直接通过类调用 也可以使用对象来调用

类：`::静态成员函数` 对象.静态成员函数

静态成员变量的初始化在类外

例：`int 类：::静态成员=0；`

Static 只在成员函数声明中出现 定义中不再出现 静态函数只能引用静态成员变量

12.

Const修饰符 就是采用引用调用的时候 为了避免错误的修改类的成员变量而加的限制

1. 在函数参数中加上**const** 这样函数就不会修改这个引用 成为常参数
2. 在类的函数后面加上**const** 那么这个函数就不会修改对象的成员变量的值
3. 如果使用**const** 在函数声明和定义的地方都得加上**const**

13.

向量 vector

向量：可以看做是程序运行中大小可以改变的数组。

声明 `vector<int> v;`

若想给向量添加新的元素，必须使用成员函数 `push_back`

`v.pushu_back(10);` 10是要添加的数 这是在末尾添加

用**size** 获得大小

14.

类类型的变量和**类类型的对象**是不同的

15.

友元函数是非成员函数，但是和成员函数有同样的

友元函数，friend

在类中 函数声明定义的时候前面加个 friend

16.

自动类型转换，由构造函数进行的自动类型转换十分常见，当重载的函数不匹配的时候会去构造函数找有没有合适的变量来进行类型转换。在构造函数的参数中如果有需要被改变的类型，那这个类型对应的类型会做个转换。将运算符重载函数以非成员函数的方式进行声明和定义前后都可以类型转换，但为了同时方便调用类内成员，将运算符以友元函数进行重载比较好。

17.

在我自己新建的一个类中添加 库文件不带 但是在zxcreatDlg.Cpp 这个文件中添加库就对，不知道为啥 后来加上 `#include "zxcreatDlg.h"` 也不对 只能恢复成原来的样子，不知道为啥。

库的包含不知道为啥一直不对 明明添加了路径 可是就是不对

18.

引用， 在返回类型之后加上 &

19.

无法解析的外部符号

编译时出现类似这样的错误：Dlrcode.obj : error LNK2019: 无法解析的外部符号 `_readRegmark`，该符号在函数 `_AboutDlgProc@16` 中被引用。这种错误的本质是链接器无法在已编译的obj、lib或dll文件中找到函数定义。

20.

.h .lib .dll

include .h 在c/c++ 常规 附加包含目录 编译的时候用到

Lib 链接的时候用到 链接器-> 常规->附加库目录 输入里面有附加依赖项

Dll 运行的时候需要用到

在vc++下面也可以设置 附加包含目录和库目录 但是建议在 c/c++ 和link那里设置

21

提取文件夹下文件的文件名 `DIR *.* /B >LIST.TXT`

22..

字符串的转换

从cstring 到 char * 或者const char *

`char fpath[1024];`

`wsprintfA(fpath, "%ls", oldcstring);`

23.

获取键盘的函数

`getchar()` 显示回文 就是会在屏幕上显示 需要按下回车才会接收

getch ()

24.

模板 template

函数模板 使函数更具有通用性

```
template <class T, class T2>
void swap(T aa,T bb)
{
    .....
}
```

为了避免编译器的不支持 最好将函数模板 函数声明 函数定义放在一个文件中

类模板

```
template <class T >
class aa
{
    public:
        T bb;
        T swp(T cc);
}

template <class T >
aa<T>::swp(T cc)
{
    .....
}
```

用类模板声明对象的时候必须指出参数类型T的具体数据类型。

```
aa<int> www;
```

25.

结构体也可以有构造函数

```
struct ListNode
{
    int val;
    ListNode *next;
    ListNode(int x) : val(x), next(NULL) {} //这里应该是构造函数 初始化 val为x next 为 null
};

ListNode preHead(0)
```

26.

指针基础

```
int aa=5;
int *p=aa; 错误   在定义指针的时候是  int*p   p=   ;
int *p=&aa;    right
*p=aa;         right
```

在结构体中

```
struct MyStruct
{
    int a;
};
MyStruct *bb;
bb->a = 9;      //是正确的
MyStruct   bb;
bb->a=9;        //是错误的  必须为指针
```

27.

头文件的添加

为了使用max min 需要添加 #include<algorithm>

28.

为了是**打印**出开的数据能够显示而不是一**闪而过**，在打印后加上**cin.get()**;这样就会等待读取键值。

29. 使用using namespace std 编译命令

命名空间的作用是 确保不同的库 如果有相同的函数名，调用的时候编译器就不知道该调用哪一个，所以加上命名空间

例如 pcl: : list();
 opencv::list();

但是如果每个函数前都加上 std: : 就比较麻烦 所以采用 using namespace std

30.

cout 相对于**printf**来说更智能 在c++中最好用cout

