我们如何理解一个类/结构体以及指针呢？

（我们不考虑重载运算符等高级操作）

一个类/结构体，由成员组成

一个类可以保含0～多个数据成员和0～多个成员函数

类/结构体，本身是一个类型，类似int，float，double，long long， char等

比如

struct A {

int 数据成员1;

void 成员函数1(/\*参数\*/) {

/\*参数\*/

}

} b; //这里可以不声明一个对象，但分号不能省略

A c;

int main() {

A b;

}

"." 表示 "的" ，比如 苹果a.颜色; 人1.吃(苹果a);

"&" 表示 取地址（作为运算），或者引用（在声明时）

比如 scanf("%d", &苹果a.重量); //取地址

int a = 5, &b = a; // 此处b是a的引用，即给a一个别名b，两者其实是一个变量

"\*"表示取出指针的对象(作为运算)，或者指针（在声明时）

比如 int \* pointer\_to\_a = &a;//声明了一个指向int的指针类型变量pointer\_to\_a，并将&a作为初值赋给它

此时表达式 (\*pointer\_to\_a) 和 a 完全等价 //\*作为运算

“->”取出指针指向对象的成员

"指针->数据成员" 和 "(\*指针).数据成员" 完全等价

"指针->成员函数(参数列表)" 和 "(\*指针).成员函数(参数列表)" 完全等价

但和(\*指针)->成员 以及\*指针->成员不等价

struct 人 {

int 重量;

void 吃(苹果 &目标苹果) {//因为是引用，所以和参数中的苹果完全一致

重量 += 目标苹果.重量;

目标苹果.重量 = 0;

}

void 凭空吃(苹果 目标苹果) {//这里的目标苹果是被复制过来的一个局部变量，

重量 += 目标苹果.重量;

目标苹果.重量 = 0; //这里修改了一个局部变量的值，

}

void 指针吃(苹果 \*目标苹果的指针) {//相当于告诉你苹果在哪，你取吃

重量 += (\*目标苹果的指针).重量;

(\*目标苹果的指针).重量 = 0;

与下面两行代码等价

重量 += 目标苹果的指针->重量;

目标苹果的位置.指针-> = 0;

}

}人1;

比如

窗口 窗口1; //表示有一个窗口类型的对象，其代号为窗口1

int int1; //表示有一个int类型的对象，其代号为int1

int1 = 窗口1.高度; // 直接翻译为：将“窗口1的高度”赋值给int1

窗口1.高度 = int1; // 直接翻译为：将int1 赋值给“窗口1的高度”

窗口1.弹出对话框(5, false); //调用窗口1的函数 “弹出对话框”, 函数的参数列表赋值为 {5, false};

窗口 窗口2;

窗口2 = 窗口1; //将窗口2 赋值为窗口1

窗口 窗口3;

窗口3.父窗口 = &窗口1; //将指向窗口1的指针赋值给“窗口3的负窗口”

窗口1.子窗口 = &窗口3;

上面两句等价为

窗口3.设置父窗口(&窗口1);

也等价为

窗口3.设置父窗口\_奇怪写法(&窗口1);

//则有以下等价的关系

窗口1 $ \equiv$ \*窗口3.父窗口

窗口1.弹出对话框(5, false); $ \equiv$ \*窗口3.父窗口.弹出对话框(5, false); $ \equiv$ 窗口3.父窗口->弹出对话框(5, false);

int1 = 窗口1.高度; $ \equiv$ int1 = (\*窗口3.父窗口).高度; $ \equiv$ int1 = 窗口3.父窗口->高度

窗口1.高度 = int1; $ \equiv$ (\*窗口3.父窗口).高度 = int1; $ \equiv$ 窗口3.父窗口->高度 = int1

struct 窗口 {

窗口 \*父窗口, \*子窗口;

int 高度;

bool 设置父窗口(窗口 \*父窗口) {

if (!父窗口) {

return false;

}

this->父窗口 = 父窗口; //假如参数名“父窗口”改为父窗口\_,则本句可以改为 父窗口 = 父窗口\_;

父窗口->子窗口 = this;

return true;

}

bool 设置父窗口\_奇怪写法(窗口 \*父窗口) {

if (!父窗口) {

return false;

}

this->父窗口 = 父窗口;

父窗口 = this;//将参数里的父窗口这个指针类型变量赋值为 指向((被要求执行函数)的当前对象)的指针

父窗口->父窗口->子窗口 = 父窗口;

return true;

}

void 弹出对话框(int 参数1, bool 参数2) {//瞎写的一个函数

if (参数2) printf("message :%d\n", 参数1);

if (父窗口) {//如果该变量非0

printf("父窗口高度为 %d\n", 父窗口->高度);//struct内部的函数可以直接访问自己的数据，不用写 窗口3.父窗口->高度，如果一定要指定本对象的什么什么可以用this->父窗口->高度

这里的this相当于反身代词“我”，

}

}

};