**C++面向对象模型初探**

1. 基础知识

|  |
| --- |
| C++中的class从面向对象理论出发，将变量(属性)和函数(方法)集中定义在一起，用于描述现实世界中的类。从计算机的角度，程序依然由数据段和代码段构成。  **C++编译器如何完成面向对象理论到计算机程序的转化？**  换句话：C++编译器是如何管理类、对象、类和对象之间的关系  具体的说：具体对象调用类写的方法，那，c++编译器是如何区分，是那个具体的类，调用这个方法那? |
| 思考一下程序结果 |
| #include "iostream"  using namespace std;  class C1  {  public:  int i; //4  int j; //4  int k; //4  protected:  private:  }; //12  class C2  {  public:  int i; //4  int j; //4  int k; //4    static int m; //4  public:  int getK() const { return k; } //4  void setK(int val) { k = val; } //4  protected:  private:  }; //24 16 12  struct S1  {  int i;  int j;  int k;  }; //  struct S2  {  int i;  int j;  int k;  static int m;  }; //  int main()  {  printf("c1:%d \n", sizeof(C1));  printf("c2:%d \n", sizeof(C2));  printf("s1:%d \n", sizeof(S1));  printf("s2:%d \n", sizeof(S2));    system("pause");  } |

1. 用内存四区解释 C++属性和方法

|  |
| --- |
| 1）C++类对象中的成员变量和成员函数是分开存储的  成员变量：  普通成员变量：存储于对象中，与struct变量有相同的内存布局和字节对齐方式  静态成员变量：存储于全局数据区中  成员函数：存储于代码段中。  问题出来了：很多对象公用一块代码？代码是如何区分具体对象的那？  换句话说：int getK() const { return k; }，代码是如何区分，具体obj1、obj2、obj3对象的k值？ |
| 2）C++编译器对普通成员函数的内部处理 |
|  |
| 请仔细思考，并说出你的总结！ |

3、总结：

|  |
| --- |
| **1、**C++类对象中的成员变量和成员函数是分开存储的。C语言中的内存四区模型仍然有效！  **2、C++中类的普通成员函数都隐式包含一个指向当前对象的this指针。**  3、静态成员函数、成员变量属于类  静态成员函数与普通成员函数的区别  静态成员函数不包含指向具体对象的指针  普通成员函数包含一个指向具体对象的指针 |