|  |  |
| --- | --- |
| **组号:** | **5组** |

****

信息科学与工程学院课程实验报告

《面向对象程序设计》

|  |  |
| --- | --- |
| 姓名： | 宋肖男 |
| 学号： | 201711990101 |
| 班级： | 计师本1701 |
| 教师： | 张庆科 |
| 时间： | 2018年11月22日 |

**面向对象程序设计实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 宋肖男 | 班级 | 计师本1701 | 学号 | 201711990101 | 组号 | 5组 |
| 时间 | 11.22 | 地点 |  | 周次 |  | 页码 |  |
| 源码 | □ 无源码 □ 文档源码 □ 托管源码 | | | | | | |
| 报  告  内  容  报  告  内  容  报  告  内  容 | 1. **实验目的**   调试运行第5章类的组合，类的依赖2个小节案例程序。   1. **实验内容**   5\_1:    5\_2：    5\_3\_1：    5\_3\_2：       1. **实验步骤**   创建新项目  录入代码  运行总结   1. **实验总结与分析**   5\_1：  对象与他内部的对象成员就有相同的生命周期。当对象被创建时，对象成员也会被创建，对象析构时，对象成员也一同被析构。  创建一个对象时，构造函数的调用次序为：首先调用对象成员的构造函数，再调用对象自身的构造函数，析构时顺序完全相反。  5\_2：  a1对象在a2对象之前被创建。对象成员构造函数的调用次序与初始化表中出现的次序无关，他始终与对象成员在类中定义的次序一致。  5\_3\_1： 依据每次产生的随机数不同，这个程序会有多种运行结果。这里Gambler与Dice类之间就是一种依赖关系。  5\_3\_2：  从语法层面上来讲，此例使用了组合机制，骰子与赌徒具有相同的生命周期，赌徒不存在了，骰子也就不存在了，他们会被同时析构掉。而在依赖机制中，骰子与赌徒是相对独立的，更符合实际情况。   1. **实验源码和源码地址**   5\_1:  1 ﻿#include<iostream>  2 **using namespace** std;  3 **class** A  4 {  5 **public**:  6 **A**()  7 {  8 cout << "创建A" << endl;  9 }  10 ~**A**()  11 {  12 cout << "析构A" << endl;  13 }  14 };  15 **class** B  16 {  17 **public**:  18 **B**()  19 {  20 cout << "创建B" << endl;  21 }  22 ~**B**()  23 {  24 cout << "析构B" << endl;  25 }  26 **private**:  27 A a;  28 };  29 **int** **main**()  30 {  31 B obj;  32 **system**("pause");  33 **return** 0;  34 }  5\_2:  1 ﻿#include<iostream>  2 **using namespace** std;  3 **class** A  4 {  5 **public**:  6 **A**(**int** a)  7 {  8 x = a;  9 cout << "A:x=" << a << endl;  10 }  11 **private**:  12 **int** x;  13 };  14 **class** B  15 {  16 **public**:  17 **B**(**int** a, **int** b, **int** c) :**a2**(b), **a1**(a)*//注意：初始化列表中是成员对象名a1,a2，而不是类A*  18 {  19 y = c;  20 cout << "B:y=" << c << endl;  21 }  22 **private**:  23 A a1, a2;*//有两个对象成员，B类包含了A类*  24 **int** y;  25 };  26 **int** **main**()  27 {  28 B **obj**(10, 20, 30);  29 **system**("pause");  30 **return** 0;  31 }  5\_3\_1:  1 ﻿#include<iostream>  2 #include<ctime>  3 #include<cstdlib>  4 **using namespace** std;  5 **class** Dice*//骰子类*  6 {  7 **public**:  8 **int** **run**()  9 {  10 **return** (**rand**() % 6 + 1);  11 }  12 };  13 **class** Gambler*//赌徒类*  14 {  15 **public**:  16 **void** **play**(Dice a, Dice b, Dice c)  17 {  18 **int** num1, num2, num3;  19 num1 = a.**run**();  20 num2 = b.**run**();  21 num3 = c.**run**();  22 **if** (num1==num2&&num1==num3)  23 cout << num1 << " "<<num2 << " " << num3 << " 豹子" << endl;  24 **else if**(num1+num2+num3>10)  25 cout << num1 << " " << num2 << " " << num3 << " 大" << endl;  26 **else**  27 cout << num1 << " " << num2 << " "<<num3 << " 小" << endl;  28 }  29 };  30 **int** **main**()  31 {  32 Dice d1, d2, d3;  33 Gambler Jack;  34 **srand**(**time**(NULL));  35 Jack.**play**(d1, d2, d3);  36  37 **system**("pause");  38 **return** 0;  39 }  40  5\_3\_2:  1 ﻿#include<iostream>  2 #include<ctime>  3 #include<cstdlib>  4 **using namespace** std;  5 **class** Dice*//骰子类*  6 {  7 **public**:  8 **int** **run**()  9 {  10 **return** (**rand**() % 6 + 1);  11 }  12 };  13 **class** Gambler*//赌徒类*  14 {  15 **public**:  16 **void** **play**()  17 {  18 **int** num1, num2, num3;  19 num1 = a.**run**();  20 num2 = b.**run**();  21 num3 = c.**run**();  22 **if** (num1 == num2&&num1 == num3)  23 cout << num1 << " " << num2 << " " << num3 << " 豹子" << endl;  24 **else if** (num1 + num2 + num3 > 10)  25 cout << num1 << " " << num2 << " " << num3 << " 大" << endl;  26 **else**  27 cout << num1 << " " << num2 << " " << num3 << " 小" << endl;  28 }  29 **private**:  30 Dice a, b, c;  31 };  32 **int** **main**()  33 {  34 Gambler Jack;  35 **srand**(**time**(NULL));  36 Jack.**play**();  37 **system**("pause");  38 **return** 0;  39 }  40  41  **源码地址:** | | | | | | |

：可根据内容自行拓展页面，作业内容尾部尽量不要留有空白