



计算机与软件工程学院

上机实验报告

**（ 2019/2020 学年 第 2 学期 ）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | C/C++程序设计 | | | |
| 课程代码 | 190901019 | | | |
| 任课教师 | 王晓明 | | | |
| 学生姓名 | 刘唐 | 年级 | 19级 |
| 学号 | 3120190971181 | 专业 | 计算机类 |
| 实验成绩 |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 实验5：类与继承 | | | 实验地点 | Home🏡 |
| 实验类型 | 设计 | 实验学时 | 2 | 实验日期 | 2020/6/3 |
| * 撰写注意：版面格式已设置好（不得更改），填入内容即可。 | | | | | |
| 1. 实验目的   掌握类与派生类的定义与使用。 | | | | | |
| 1. 实验内容 2. 实验任务1   实现一个简单数组类，要求数组类初始化为五个整型空间，并且实现数组类内部返回最大值和最小值函数。  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   键盘输入5个整数   1. **数据存储**（输入数据在内存中的存储）   对象中的可改变大小的私有成员   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   申请一个IntArrsy类，内声明公有的设置值函数，大小函数，求最大最小值函数以及两个参数  主函数内用size=5直接初始化申请长度为5的数组对象  用输入流写入5个数组内容  通过求大小值输出最大最小值   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）      1. 实验任务2   讨论继承与派生的权限问题 。  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   无   1. **数据存储**（输入数据在内存中的存储）   无   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   分别建立公有继承和私有继承  1.尝试在公有继承派生类内部访问基类以及自身内部数据  2.尝试在私有继承派生类内部访问基类以及自身内部数据  3.尝试在公有继承派生类外部访问基类以及自身内部数据  4.尝试在私有继承派生类外部访问基类以及自身内部数据  //哎呀就是**地毯式尝试访问**  //保护成员的权限属性较简单，就不加了   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）     **//有输出就意味着内部访问成功**  **//其中666意味着在外部访问成功**  **//v1连内部访问都不行，外部访问肯定不行了**   1. 实验任务3     程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   已经放进去了  5个学生数据   1. **数据存储**（输入数据在内存中的存储）   对象数组   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   对比每个学生的分数  max函数内输出最大   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大） | | | | | |
| 1. 实验环境 2. 操作系统：Windows10 3. 开发工具：visual studio 2019 4. 实验设备：PC | | | | | |
| 1. 提交文档   提交本实验报告（电子版），文件名命名：实验X\_报告\_学号\_姓名.doc  教师将**批阅后（有分数）**的全体学生实验报告刻入一张光盘存档，保证光盘可读。 | | | | | |
| 1. 附：源代码   //前方超多行预警  实验1，类实现int数组  #include<iostream>  using namespace std;  class IntArray  {  public:  void setArray(int len, int\* in);  int getSize();  int setVal(int pos, int val);  int getMaxVal();  int getMinVal();  private:  int\* data;  int size;  };  int main()  {  IntArray array;  int i, size = 0, v, max, min;  int in[] = { 1,2,3,4,5 };  array.setArray(5, in);//先进行初始化  size = array.getSize();  cout << "请输入" << size << "数字：" << endl;  for (int i = 0; i < size; i++)  {  cin >> v;  array.setVal(i, v);//初始化数组  }  max = array.getMaxVal();  min = array.getMinVal();  cout << "最大值为：" << max << endl;  cout << "最小值为：" << min << endl;  return 0;  }  int IntArray::getSize() {  return size;//获取数组大小  }  int IntArray::setVal(int pos, int val)  {  if ((pos < 0) || (pos > size))return -1;//检查位置是否合法  data[pos] = val;//pos是位置0，1，2，3，val是值  return 0;  }  int IntArray::getMaxVal() //得到最大值  {  int temp = data[0];  for (int i = 0; i < size; i++)  {  if (data[i] > temp) {  temp = data[i];  }  }  return temp;  }  int IntArray::getMinVal() {//得到最小值  int temp = data[0];  for (int i = 0; i < size; i++)  {  if (data[i] < temp) {  temp = data[i];  }  }  return temp;  }  void IntArray::setArray(int len, int\* in) {  size = len;//大小空间性分配  data = new int[len];//把data初始化了len这么长的数据空间  for (int i = 0; i < len; ++i)data[i] = in[i];  }  实验2讨论继承与派生的权限问题  #include<iostream>  using namespace std;  class Base  {  int v1;  public:  int v2;  Base(int a = 0, int b = 0)  {  v1 = a;  v2 = b;  }  };  class Derived:public Base //公有继承  {  int v3;  public:  int v4;  Derived(int a = 0, int b = 0)  {  v3 = a;  v4 = b;  }  void func()  {  cout << "公有继承内部访问："<<endl;  //cout << v1 << endl;  //v1是Base的私有成员，任何继承不能访问！  cout << "基类公有成员v2:" << v2 << endl; //基类的公有成员，以公有继承，可以访问  cout << "派生类私有成员v3:" << v3 << endl; //派生类的私有成员，派生类内部可以访问  cout << "派生类公有成员v4:" << v4 << endl; //派生类的公有成员，派生类内外都可以访问  }  };  class Derived2 : private Base  {  int v5;  public:  int v6;  Derived2(int a = 0, int b = 0)  {  v5 = a;  v6 = b;  }  void func()  {  cout << "私有继承内部访问：" << endl;  //cout << v1 << endl;  //v1是Base的私有成员，任何继承不能访问！  cout <<"基类公有成员v2："<< v2 << endl; //基类的公有成员，派生内部可以访问  cout <<"派生类私有成员v5: "<< v5 << endl; //派生类的私有成员，派生类内部可以访问  cout <<"派生类公有成员v6："<< v6 << endl; //派生类的公有成员，派生类内外都可以访问  }  };  int main(void)  {  //讨论公有继承访问权限：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Derived obj = Derived(5, 6);//定义的时候：  /\* v1=0  v2=0  v3=5  v4=6 \*/  //obj.v1 = 666;  //1.私有数据通过公有继承下来，在派生类的内部都不能访问。  obj.v2 = 666; //2.从Base公有继承，可以在外部访问。  //obj.v3 = 666;  //3.Dervied的私有成员，在类的外部不能访问。  obj.v4 = 666; //4.obj自己的公有成员，可以在外部访问  //外部访问后：  /\* v1=0  v2=666  v3=5  v4=666 \*/  obj.func();  //公有继承内部访问后：输出v2，v3，v4  //讨论私有继承访问权限：\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Derived2 obj2 = Derived2(5, 6);//定义的时候：  /\* v1=0  v2=0  v5=5  v6=6 \*/  //obj.v1 = 666;  //1.私有数据不能访问。  //obj2.v2 = 666;  //2.从Base私有继承，只能在内部访问。  //obj.v5 = 666;  //3.Dervied2的私有成员，在类的外部不能访问。  obj2.v6 = 666; //4.obj自己的公有成员，可以在外部访问  obj2.func();  //私有继承内部访问后：输出v2,v5,v6  return 0;  }  实验三  #include <iostream>  using namespace std;  class Student  {  public:  Student(int n, float s) : id(n), score(s) {}//学生类  int id = 0;  float score = 0;  };  void max(Student\* A, int n);//对象指针  int main()  {  Student stu[5] =  {  Student(19097118111, 75),//发现一个问题：用int做类型  Student(19097118112, 76),//学号就很难很长（值太大）  Student(19097118113, 82),//所以下次就用char型数组吧  Student(19097118114, 87),  Student(19097118115, 75)  };  max(stu, 5);  system("pause");  return 0;  }  void max(Student\* A, int n)//对象指针  {  Student\* p = A, ma(0, 0);  float max;  for (max = A->score; p < A + n; p++)  {  if ((p->score) > max)  {  max = p->score;  ma = \*p;  }  }  cout << "NO.1 学生的信息: \n" <<"ID："<< ma.id << endl << "分数："<<ma.score << endl;  } | | | | | |