



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6：虚函数与抽象类 | | | 实验地点 | Home🌏 |
| 实验类型 | 设计 | 实验学时 | 2 | 实验日期 | 2020/6/5 |
| * 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   掌握虚函数、抽象类的定义与使用。 | | | | | |
| 1. 实验内容 2. 实验任务1   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） 2. **数据存储**（输入数据在内存中的存储） 3. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图） 4. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大） 5. 实验任务2   人类的本质：  设计人类、英语学生类和复读机类三个类。  具体要求如下：   1. 人类（ Chinese ）    * 它有一个虚函数 greet，函数输出中文问候，即你好。 2. 英语学生类（ EnglishLearner ）    * 继承 Chinese 类，重写 greet 函数，访问性为 public，输出英文问候，即Hello。 3. 复读机类（ Repeater ）    * 继承 Chinese 类，以 public 访问性重写 greet 函数，函数调用 Chinese 类的 greet 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）   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）   调用三个对象的greet函数。  三个类中，  一个是基类， 虚函数greet输出“你好”  另外两个分别公有继承基类  对虚函数的操作：一个重载greet函数：输出“Hello”  另一个重载greet函数：调用基函数的greet函数   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）   **程序内执行输入，每个数字都是double型**  Circle circle(1);  Square square(1);  Rectangle rectangle(1, 1);  Trapezoid trapezoid(1, 1, 1);  Triangle triangle(1, 1);   1. **数据存储**（输入数据在内存中的存储）   每一个类只定义了一个对象  所以各个对象只接收相对应的数字  每个数字都以double型存储  实际存储要素如下：  圆形：半径  方形：边长  矩形：长和宽  梯形：上底下底，高  三角形：底和高   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）   每个派生类中用虚函数返回该图形面积   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   实验二=====================================================  说明：以下代码为头文件，名为“人类的本质.h”  #include <iostream>  using namespace std;  class Chinese//人类  {  public:  virtual void greet()  {  cout << "你好" << endl;  }  };  class EnglishLearner : public Chinese //英语学生类  {  public:  void greet()  {  cout << "Hello" << endl;  }  };  class Repeater : public Chinese //复读机类  {  public:  void greet()  {  Chinese::greet();  }  };  /\*-------------头文件结束-------------\*/  以下代码为主函数文件：“main.cpp”  //主函数不输入输出居然就不用io流诶！  #include "人类的本质.h"  int main()  {  Chinese i;  EnglishLearner e;  Repeater ii;  i.greet();  e.greet();  ii.greet();  }  /\*-------------主文件结束------------\*/  实验三：-----------------------------------------------------------------------  #include <iostream>  using namespace std;  class Shape //抽象形状类  {  public:  Shape() {} //构造函数，本基类无任何数据成员  virtual ~Shape() {} //虚析构函数，只须派生类收尾  virtual double Area() { return 0; }//到时候用输出流来show，这里就不show了  virtual void showShapeName() = 0;//函数名有点长，主要是因为之前一直遵循“输出最好不要再函数内实现，所以加了show的单词。  };  class Circle : public Shape //圆形  {  double radius; //圆形只有周长属性（只用求面积嘛）  public:  Circle(double r) { radius = r; }//圆构造函数接收半径  ~Circle() {} //圆析构函数不用做事  double Area() { return 3.14159 \* radius \* radius; }  void showShapeName() { cout << "圆形的"; }  };  class Square : public Shape //正方形  {  double x; //边长  public:  Square(double a) { x = a; } //边长属性初始化  ~Square() {}  double Area() { return x \* x; }  void showShapeName() { cout << "正方形的"; }  };  class Rectangle : public Shape //矩形  {  double x, y; //长，宽  public:  Rectangle(double a, double b) { x = a; y = b; }//收入边长属性  ~Rectangle() {}  double Area() { return x \* y; }  void showShapeName() { cout << "矩形的"; }  };  class Trapezoid : public Shape //梯形  {  double x, y, z; //上底，下底，高  public:  Trapezoid(double a, double b, double c) { x = a; y = b; z = c; }//收入三要素属性  ~Trapezoid() {}  double Area() { return ((x + y) \* z) / 2; }  void showShapeName() { cout << "梯形的"; }  };  class Triangle : public Shape //三角形  {  double x, y; // 底 ，高 or 高 ，底  public:  Triangle(double a, double h) { x = a; y = h; }//收入底，高  ~Triangle() {}  double Area() { return (x \* y) / 2; }  void showShapeName() { cout << "三角形的"; }  };  int main()  {  Circle circle(1);  Square square(1);  Rectangle rectangle(1, 1);  Trapezoid trapezoid(1, 1, 1);  Triangle triangle(1, 1);  double sum = 0;  Shape\* pt[5] = { &circle, &square, &rectangle, &trapezoid, &triangle };  Shape\*\* p;//等会用双重指针的自增  for (p = pt; p < pt + 5; p++) // p = pt[5] -> 五个对象  { // p->五个对象（数组）->对象内函数  (\*p)->showShapeName(); //相当于 pt[0]->showShapeName();  cout << "面积为 \t" << (\*p)->Area() << endl;  sum += (\*p)->Area();  }  cout << "总面积为 \t" << sum << endl;  system("pause");  return 0;  } | | | | | |