C++语言作业

17 计基 杨添宝 320170941671

销售公司

头文件：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | /\* SalesCompany.h \*/  #ifndef \_SALES\_COMPANY\_H\_  #define \_SALES\_COMPANY\_H\_  class Employee {  protected:  char number[20];  char name[20];  float basicSalary;  public:  Employee(const char[] = "\0", const char[] = "\0", float = 2000);  void input();  void print();  };  class Salesman: public Employee {  protected:  static float commrate; //提成比例  int sales; // 销售额  float salary;  public:  Salesman(int = 0);  void input();  void pay();  void print();  };  class Salesmanager: public Salesman {  private:  float jobSalary;  public:  Salesmanager(float = 3000);  void input();  void pay();  void print();  };  #endif /\* SalesCompay.h \*/ |

cpp文件：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58 | /\* SalesCompany.cpp \*/  #include "SalesCompany.h"  #include <iostream>  #include <cstring>  // 一般员工  Employee::Employee(const char number[], const char name[], float basicSalary) {  strcpy(this->number, number);  strcpy(this->name, name);  this->basicSalary = basicSalary;  }  void Employee::input()  {  std::cout << "姓名：";  std::cin >> name;  std::cout << "编号：";  std::cin >> number;  }  void Employee::print() {  std::cout << "一般员工：" << name << std::endl  << "编号：" << number << std::endl  << "工资：" << basicSalary << std::endl;  }  // 销售员工  float Salesman::commrate = 0.005;  Salesman::Salesman(int sales) {  this->sales = sales;  }  void Salesman::input() {  Employee::input();  std::cout << "销售额：";  std::cin >> sales;  }  void Salesman::pay() {  salary = basicSalary + sales \* commrate;  }  void Salesman::print() {  pay();  std::cout << "销售员工：" << name << std::endl  << "编号：" << number << std::endl  << "工资：" << salary << std::endl;  }  // 销售经理  Salesmanager::Salesmanager(float jobSalary) {  this->jobSalary = jobSalary;  }  void Salesmanager::input() {  Employee::input();  std::cout << "销售额：";  std::cin >> sales;  }  void Salesmanager::pay() {  salary = jobSalary + sales \* commrate;  }  void Salesmanager::print() {  pay();  std::cout << "销售经理 ：" << name << std::endl  << "编号：" << number << std::endl  << "工资：" << salary << std::endl;  } |

测试代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | /\* main.cpp \*/  #include <iostream>  #include "SalesCompany.h"  using namespace std;  int main()  {  cout << "基本员工" << endl;  Employee emp1;  emp1.input();  emp1.print();  cout << "---------------------------------" << endl;  cout << "销售员" << endl;  Salesman emp2;  emp2.input();  emp2.print();  cout << "---------------------------------" << endl;  cout << "销售经理" << endl;  Salesmanager emp3;  emp3.input();  emp3.print();  return 0;  } |

形状

头文件：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 | /\* Shape.h \*/  #ifndef \_SHAPE\_H\_  #define \_SHAPE\_H\_  class Shape {  protected:  char name[10];  float area; // 面积  public:  const char \*getName() const;  float getArea() const;  void setArea(float);  Shape(float = 0, const char[] = "形状");  void print() const;  };  class TwoDimShape: public Shape {  protected:  float perimeter; // 周长  public:  TwoDimShape(float = 0, float = 0, const char[] = "二维形状");  float getPerimeter() const;  void setPerimeter(float);  void print() const;  };  class ThreeDimShape: public Shape {  protected:  float volume; //体积  public:  ThreeDimShape(float = 0, float = 0, const char[] = "三维形状");  float getVolume() const;  void setVolume(float volume);  void print() const;  };  class Rectangle: public TwoDimShape {  private:  float length, width;  public:  Rectangle(float, float);  float getLength() const;  void setLength(float length);  float getWidth() const;  void setWidth(float width);  void print() const;  };  class Triangle: public TwoDimShape {  private:  float a, b, c; //三角形三边  public:  Triangle(float, float, float);  void SetSide(float, float, float);  void print() const;  };  class Circle: public TwoDimShape {  private:  float radius;  public:  Circle(float);  float getRadius() const;  void setRadius(float radius);  void print() const;  };  class Cuboid: public ThreeDimShape {  private:  float length, width, height;  public:  Cuboid(float, float, float);  float getLength() const;  void setLength(float length);  float getWidth() const;  void setWidth(float width);  float getHeight() const;  void setHeight(float height);  void print() const;  };  class Sphere: public ThreeDimShape {  private:  float radius;  public:  Sphere(float);  float getRadius() const;  void setRadius(float radius);  void print() const;  };  #endif /\* Shape.h \*/ |

cpp文件：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168 | /\* Shape.cpp \*/  #include "Shape.h"  #include <cstring>  #include <iostream>  #include <cmath>  // 形状  Shape::Shape(float area, const char name[]) {  this->area = area;  strcpy(this->name, name);  }  void Shape::print() const {  std::cout << "形状名称：" << name << std::endl  << "面积：" << area << std::endl;  }  const char\* Shape::getName() const {  return name;  }  float Shape::getArea() const {  return area;  }  void Shape::setArea(float area) {  Shape::area = area;  }  // 二维形状  TwoDimShape::TwoDimShape(float perimeter, float area, const char name[]): Shape(area, name) {  this->perimeter = perimeter;  }  float TwoDimShape::getPerimeter() const {  return perimeter;  }  void TwoDimShape::setPerimeter(float perimeter) {  TwoDimShape::perimeter = perimeter;  }  void TwoDimShape::print() const {  Shape::print();  std::cout << "周长：" << perimeter << std::endl;  }  // 三维形状  ThreeDimShape::ThreeDimShape(float volume, float area, const char name[]): Shape(area, name) {  this->volume = volume;  }  float ThreeDimShape::getVolume() const {  return volume;  }  void ThreeDimShape::setVolume(float volume) {  ThreeDimShape::volume = volume;  }  void ThreeDimShape::print() const  {  Shape::print();  std::cout << "体积：" << volume << std::endl;  }  // 矩形  Rectangle::Rectangle(float length, float width)  : TwoDimShape(  2 \* (length + width),  length \* width, "矩形") {  if (length <= 0 || width <= 0)  throw "长方体的长和宽必须大于0";  this->length = length;  this->width = width;  }  float Rectangle::getLength() const {  return length;  }  void Rectangle::setLength(float length) {  Rectangle::length = length;  }  float Rectangle::getWidth() const {  return width;  }  void Rectangle::setWidth(float width) {  Rectangle::width = width;  }  void Rectangle::print() const {  TwoDimShape::print();  std::cout << "长：" << length << std::endl  << "宽：" << width << std::endl;  }  // 三角形  Triangle::Triangle(float a, float b, float c)  : TwoDimShape(  a + b + c,  0, "三角形") {  if (a + b <= c || a + c <= b || b + c <= a)  throw "a, b, c不能构成三角形";  float p = (a + b + c) / 2;  TwoDimShape::area = sqrt(p \* (p - a) \* (p - b) \* (p - c));  this->a = a;  this->b = b;  this->c = c;  }  void Triangle::SetSide(float a, float b, float c)  {  this->a = a;  this->b = b;  this->c = c;  }  void Triangle::print() const {  TwoDimShape::print();  std::cout << "三边长分别为：" << a << "、" << b << "、" << c << std::endl;  }  // 圆形  Circle::Circle(float radius)  : TwoDimShape(  static\_cast<float>(2 \* M\_PI \* radius),  static\_cast<float>(M\_PI \* radius \* radius), "圆形") {  this->radius = radius;  }  float Circle::getRadius() const {  return radius;  }  void Circle::setRadius(float radius) {  Circle::radius = radius;  }  void Circle::print() const {  TwoDimShape::print();  std::cout << "半径：" << radius << std::endl;  }  // 长方体  Cuboid::Cuboid(float length, float width, float height)  :ThreeDimShape(  length \* width \* height,  2 \* (length \* width + length \* height + width \* height), "长方体"){  this->length = length;  this->width = width;  this->height = height;  }  float Cuboid::getLength() const {  return length;  }  void Cuboid::setLength(float length) {  Cuboid::length = length;  }  float Cuboid::getWidth() const {  return width;  }  void Cuboid::setWidth(float width) {  Cuboid::width = width;  }  float Cuboid::getHeight() const {  return height;  }  void Cuboid::setHeight(float height) {  Cuboid::height = height;  }  void Cuboid::print() const {  ThreeDimShape::print();  std::cout << "长：" << length << std::endl  << "宽：" << width << std::endl  << "高：" << height << std::endl;  }  // 球体  Sphere::Sphere(float radius)  : ThreeDimShape(  static\_cast<float>(4 / 3.0 \* M\_PI \* radius \* radius \* radius),  static\_cast<float>(4 \* M\_PI \* radius \* radius), "球体") {  this->radius = radius;  }  float Sphere::getRadius() const {  return radius;  }  void Sphere::setRadius(float radius) {  Sphere::radius = radius;  }  void Sphere::print() const {  ThreeDimShape::print();  std::cout << "半径：" << radius << std::endl;  } |

测试代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | /\* main.cpp \*/  #include <iostream>  #include "Shape.h"  using namespace std;  int main()  {  Shape s0;  s0.print();  cout << "-----------------------------" << endl;  TwoDimShape s1;  s1.print();  cout << "-----------------------------" << endl;  ThreeDimShape s2(100, 80, "ThreeDimShape");  s2.print();  cout << "-----------------------------" << endl;  Rectangle s3(3, 4);  s3.print();  cout << "-----------------------------" << endl;  Triangle s4(3, 4, 5);  s4.print();  cout << "-----------------------------" << endl;  Circle s5(2);  s5.print();  cout << "-----------------------------" << endl;  Cuboid s6(7, 8, 9);  s6.print();  cout << "-----------------------------" << endl;  Sphere s7(3);  s7.print();  return 0;  } |