C++语言作业

17 计基 杨添宝 320170941671

销售公司

头文件：

```
   /* SalesCompany.h */
 1 #ifndef _SALES_COMPANY_H_
 2 #define _SALES_COMPANY_H_
 3
 4 class Employee {
 5 protected:
 6     char number[20];
 7     char name[20];
 8     float basicSalary;
 9 public:
10     Employee(const char[] = "\0", const char[] =
   "\0", float = 2000);
11     void input();
12     void print();
13 };
14
15 class Salesman: public Employee {
16 protected:
17     static float commrate; //提成比例
18     int sales;  // 销售额
19     float salary;
20 public:
21     Salesman(int = 0);
22     void input();
23     void pay();
24     void print();
25 };
26 class Salesmanager: public Salesman {
27 private:
```

```
28      float jobSalary;
29  public:
30      Salesmanager(float = 3000);
31      void input();
32      void pay();
33      void print();
34  };
35
36  #endif /* SalesCompay.h */
```

cpp 文件：

```
    /* SalesCompany.cpp */
 1  #include "SalesCompany.h"
 2  #include <iostream>
 3  #include <cstring>
 4  // 一般员工
 5  Employee::Employee(const char number[], const char
    name[], float basicSalary) {
 6      strcpy(this->number, number);
 7      strcpy(this->name, name);
 8      this->basicSalary = basicSalary;
 9  }
10  void Employee::input()
11  {
12      std::cout << "姓名：";
13      std::cin >> name;
14      std::cout << "编号：";
15      std::cin >> number;
16  }
17  void Employee::print() {
18      std::cout << "一般员工：" << name << std::endl
19                << "编号：" << number << std::endl
20                << "工资：" << basicSalary << std::endl;
21  }
22  // 销售员工
```

```cpp
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：

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

形状

头文件：

```
   /* Shape.h */
 1 #ifndef _SHAPE_H_
 2 #define _SHAPE_H_
 3
 4 class Shape {
 5 protected:
 6     char name[10];
 7     float area; // 面积
 8 public:
 9     const char *getName() const;
10     float getArea() const;
11     void setArea(float);
12     Shape(float = 0, const char[] = "形状");
13
14     void print() const;
15 };
16
17 class TwoDimShape: public Shape {
18 protected:
19     float perimeter;    // 周长
20 public:
21     TwoDimShape(float = 0, float = 0, const char[]
   = "二维形状");
22     float getPerimeter() const;
23     void setPerimeter(float);
24     void print() const;
25 };
26
27 class ThreeDimShape: public Shape {
28 protected:
29     float volume;   //体积
30 public:
31     ThreeDimShape(float = 0, float = 0, const
   char[] = "三维形状");
32     float getVolume() const;
```

```
33      void setVolume(float volume);
34      void print() const;
35  };
36
37  class Rectangle: public TwoDimShape {
38  private:
39      float length, width;
40  public:
41      Rectangle(float, float);
42      float getLength() const;
43      void setLength(float length);
44      float getWidth() const;
45      void setWidth(float width);
46      void print() const;
47  };
48
49  class Triangle: public TwoDimShape {
50  private:
51      float a, b, c;  //三角形三边
52  public:
53      Triangle(float, float, float);
54      void SetSide(float, float, float);
55      void print() const;
56  };
57
58  class Circle: public TwoDimShape {
59  private:
60      float radius;
61  public:
62      Circle(float);
63      float getRadius() const;
64      void setRadius(float radius);
65      void print() const;
66  };
67
68  class Cuboid: public ThreeDimShape {
69  private:
70      float length, width, height;
```

```
71 public:
72     Cuboid(float, float, float);
73     float getLength() const;
74     void setLength(float length);
75     float getWidth() const;
76     void setWidth(float width);
77     float getHeight() const;
78     void setHeight(float height);
79     void print() const;
80 };
81 
82 class Sphere: public ThreeDimShape {
83 private:
84     float radius;
85 public:
86     Sphere(float);
87     float getRadius() const;
88     void setRadius(float radius);
89     void print() const;
90 };
91 
92 #endif /* Shape.h */
```

cpp 文件：

```cpp
/* 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();
```

```
35      std::cout << "周长: " << perimeter << std::endl;
36  }
37  // 三维形状
38  ThreeDimShape::ThreeDimShape(float volume, float
    area, const char name[]): Shape(area, name) {
39      this->volume = volume;
40  }
41  float ThreeDimShape::getVolume() const {
42      return volume;
43  }
44  void ThreeDimShape::setVolume(float volume) {
45      ThreeDimShape::volume = volume;
46  }
47  void ThreeDimShape::print() const
48  {
49      Shape::print();
50      std::cout << "体积: " << volume << std::endl;
51  }
52  // 矩形
53  Rectangle::Rectangle(float length, float width)
54                      : TwoDimShape(
55                          2 * (length + width),
56                          length * width, "矩形") {
57      if (length <= 0 || width <= 0)
58          throw "长方体的长和宽必须大于 0";
59      this->length = length;
60      this->width = width;
61  }
62  float Rectangle::getLength() const {
63      return length;
64  }
65  void Rectangle::setLength(float length) {
66      Rectangle::length = length;
67  }
68  float Rectangle::getWidth() const {
69      return width;
70  }
71  void Rectangle::setWidth(float width) {
```

```
72      Rectangle::width = width;
73  }
74  void Rectangle::print() const {
75      TwoDimShape::print();
76      std::cout << "长：" << length << std::endl
77                << "宽：" << width << std::endl;
78  }
79  // 三角形
80  Triangle::Triangle(float a, float b, float c)
81                      : TwoDimShape(
82                          a + b + c,
83                          0, "三角形") {
84      if (a + b <= c || a + c <= b || b + c <= a)
85          throw "a, b, c 不能构成三角形";
86      float p = (a + b + c) / 2;
87      TwoDimShape::area = sqrt(p * (p - a) * (p - b)
    * (p - c));
88      this->a = a;
89      this->b = b;
90      this->c = c;
91  }
92  void Triangle::SetSide(float a, float b, float c)
93  {
94      this->a = a;
95      this->b = b;
96      this->c = c;
97  }
98  void Triangle::print() const {
99      TwoDimShape::print();
100     std::cout << "三边长分别为：" << a << "、" << b <<
    "、" << c << std::endl;
101 }
102 // 圆形
103 Circle::Circle(float radius)
104                 : TwoDimShape(
105                     static_cast<float>(2 * M_PI *
    radius),
```

```
106                     static_cast<float>(M_PI * radius *
    radius), "圆形") {
107     this->radius = radius;
108 }
109 float Circle::getRadius() const {
110     return radius;
111 }
112 void Circle::setRadius(float radius) {
113     Circle::radius = radius;
114 }
115 void Circle::print() const {
116     TwoDimShape::print();
117     std::cout << "半径: " << radius << std::endl;
118 }
119 // 长方体
120 Cuboid::Cuboid(float length, float width, float
    height)
121                 :ThreeDimShape(
122                     length * width * height,
123                     2 * (length * width + length *
    height + width * height), "长方体"){
124     this->length = length;
125     this->width = width;
126     this->height = height;
127 }
128 float Cuboid::getLength() const {
129     return length;
130 }
131 void Cuboid::setLength(float length) {
132     Cuboid::length = length;
133 }
134 float Cuboid::getWidth() const {
135     return width;
136 }
137 void Cuboid::setWidth(float width) {
138     Cuboid::width = width;
139 }
140 float Cuboid::getHeight() const {
```

```
141     return height;
142 }
143 void Cuboid::setHeight(float height) {
144     Cuboid::height = height;
145 }
146 void Cuboid::print() const {
147     ThreeDimShape::print();
148     std::cout << "长: " << length << std::endl
149               << "宽: " << width << std::endl
150               << "高: " << height << std::endl;
151 }
152 // 球体
153 Sphere::Sphere(float radius)
154                : ThreeDimShape(
155                    static_cast<float>(4 / 3.0 * M_PI
    * radius * radius * radius),
156                    static_cast<float>(4 * M_PI *
    radius * radius), "球体") {
157     this->radius = radius;
158 }
159 float Sphere::getRadius() const {
160     return radius;
161 }
162 void Sphere::setRadius(float radius) {
163     Sphere::radius = radius;
164 }
165 void Sphere::print() const {
166     ThreeDimShape::print();
167     std::cout << "半径: " << radius << std::endl;
168 }
```

测试代码：

```cpp
/* 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;
}
```