

Day09 面向对象-封装继承

Key Point

- 封装/ 数据隐藏
- 继承的基本语法
- 访问修饰符
- 对象创建过程
- super 关键字
- 方法覆盖

练习

1. (继承、this 和super 关键字) 有以下代码

```

class Super{
    public Super(){
        System.out.println("Super()");
    }
    public Super(String str){
        System.out.println("Super(String)");
    }
}
class Sub extends Super{
    public Sub(){
        System.out.println("Sub()");
    }
    public Sub(int i){
        this();
        System.out.println("Sub(int)");
    }
    public Sub(String str){
        super(str);
        System.out.println("Sub(String)");
    }
}
public class TestSuperSub{
    public static void main(String args[]){
        Sub s1 = new Sub();
        Sub s2 = new Sub(10);
        Sub s3 = new Sub("hello");
    }
}

```

写出该程序运行的结果。

2. (super) 看下面代码，写出程序运行的结果

```

class Super{
    public void m1(){
        System.out.println("m1() in Super" );
    }
    public void m2(){
        System.out.println("m2() in Super" );
    }
}
class Sub extends Super{
    public void m1(){
        System.out.println("m1() in Sub");
        super.m1();
    }
}
public class TestSuperSub{
    public static void main(String args[]){
        Sub s = new Sub();
        s.m1();
        s.m2();
    }
}

```

3、按要求写代码

类Animal

属性：name

方法：eat () //打印输出" animal eat()"

sleep()//打印输出" Animal sleep()"

1、写出Animal类

2、提供有参无参构造方法，并创建对象，调用eat和sleep方法

4、按要求写代码

类Dog继承Animal

方法：run () //打印输出" Dog run"

shout()//打印输出" Dog shout"

sleep()//重写父类的方法，打印输出 "Dog sleep"

swim()//打印输出" Dog swim"

1、写出Dog类

2、在Dog类的无参构造方法中指明创建父类对象时，使用父类的有参构造方法，将父类对象name值赋值为" Dog"

3、创建Dog类对象，并调用sleep、swim方法

5、按要求写代码

类Cat继承Animal

方法：climb()//打印输出" Cat Climb"

shout()//打印" Cat shout"

sleep()//重写父类中的方法，打印输出" Cat sleep"

1、写出Cat类

2、在Cat类无参构造指明创建父类对象时，使用父类的有参构造方法并将父类对象的name属性赋值值" Cat"

3、创建Cat对象，并调用sleep、climb方法

6.(访问修饰符)有如下代码

```

//MyClass.java
package corejava.chp7;
public class MyClass{
    private int value;
    public MyClass(){}
    MyClass(int value){
        this.value = value;
    }
    public int getValue(){
        return value;
    }
    public void setValue(int value){
        this.value = value;
    }
}

//TestMyClass1.java
package corejava.chp7;
public class TestMyClass1{
    public static void main(String args[]){
        MyClass mc1 = new MyClass();
        MyClass mc2 = new MyClass(10);
        System.out.println(mc1.value);
        System.out.println(mc2.value);
    }
}

//TestMyClass2.java
package corejava.temp;
import corejava.chp7.*;
public class TestMyClass2{
    public static void main(String args[]){
        MyClass mc1 = new MyClass();
        MyClass mc2 = new MyClass(10);
        System.out.println(mc1.value);
        System.out.println(mc2.value);
    }
}

```

以上代码有哪些地方编译出错？假设不允许修改MyClass 类，那应该如何修改？

7. (继承、访问修饰符) 有如下代码

```
1) //MyClass.java
2) package corejava.chp7;
3) public class MyClass{
4)     int value;
5) }
6)
7) //MySubClass.java
8) package corejava.temp;
9) import corejava.chp7.MyClass;
10) public class MySubClass extends MyClass{
11)     public MySubClass(int value){
12)         this.value = value;
13)     }
14) }
```

选择正确答案：

- A. 编译通过
- B. 编译不通过，应把第12 行改成super.value = value;
- C. 编译不通过，应把第12 行改成super(value);
- D. 编译不通过，可以为MySubClass 增加一个value 属性
- E. 编译不通过，把第4行改为protected int value; 把第12 行改为super.value = value;

8. (继承、对象构造过程) 有以下代码

```

class Meal{
    public Meal(){
        System.out.println("Meal()");
    }
}
class Lunch extends Meal{
    public Lunch(){
        System.out.println("Lunch()");
    }
}
class Vegetable {
    public Vegetable(){
        System.out.println("Vegetable()");
    }
}
class Potato extends Vegetable{
    public Potato(){
        System.out.println("Potato()");
    }
}
class Tomato extends Vegetable{
    public Tomato(){
        System.out.println("Tomato()");
    }
}
class Meat{
    public Meat(){
        System.out.println("Meat()");
    }
}
class Sandwich extends Lunch{
    Potato p = new Potato();
    Meat m = new Meat();
    Tomato t = new Tomato();
    public Sandwich(){
        System.out.println("Sandwich()");
    }
}
public class TestSandwich{
    public static void main(String args[]){
        Sandwich s = new Sandwich();
    }
}

```

写出这段代码的输出结果。

9. (默认构造函数) 有以下代码

```

class Super{
}

class Sub extends Super{
    public Sub(){}
    public Sub(String str){
        super(str);
    }
}

```

问：该程序应该如何修改才能编译通过？

10. (方法覆盖) 有如下代码

```

class Super{
    int method(){return 0;}
}

class Sub extends Super{
    // 1
}

```

在//1 处，能编译通过的代码为：

- A. public int method(){return 0;}
- B. void method(){}
- C. void method(int n){}

11. (方法覆盖) 有如下代码

```

class Super{
    private void method(){}
}

class Sub extends Super{
    //1
}

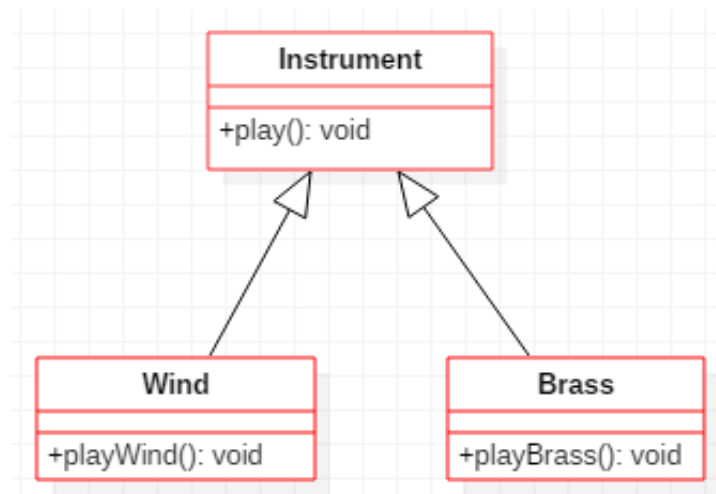
```

在//1 处，能编译通过的代码为：

- A. public int method(){return 0;}
- B. void method(){}
- C. void method(int n){}
- D. private void method(){}

12、说出this和super的区别

13、按类图要求写出所有的类



Instrument表示乐器类，play方法打印输出“ 弹奏乐器”

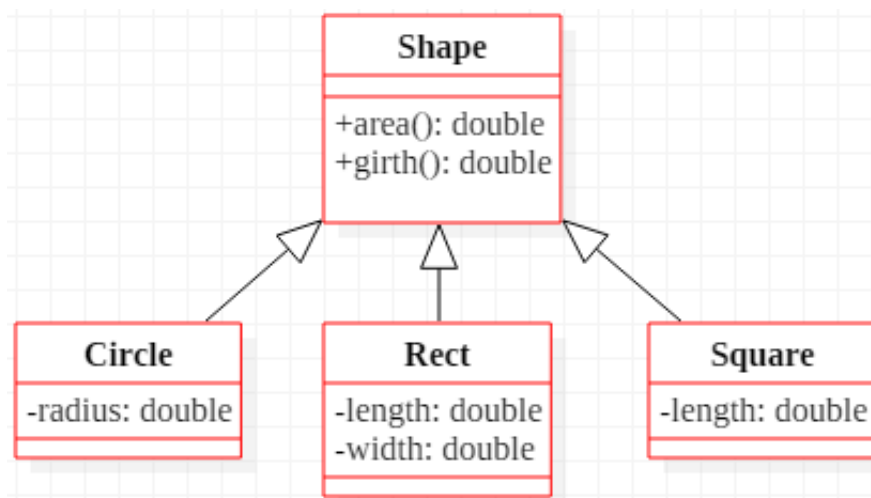
Wind继承Instrument，重写play方法，打印输出“ 弹奏Wind”

并提供另外一个playWind方法，打印输出“ 调用wind的play2方法”

Brass继承Instrument，重写play方法，打印输出“ 弹奏Brass”

并提供另外一个playBrass方法，打印输出“ 调用Brass的play2方法”

14. (封装、继承) 有以下几个类，根据下面的继承关系，用 Java 代码实现。



- a) Circle 类 (圆形) , 属性 : 半径 ; 方法 : 求周长、求面积
- b) Rect 类 (矩形) , 属性 : 长、宽 ; 方法 : 求周长、求面积
- c) Square 类 (正方形) , 属性 : 边长 ; 方法 : 求周长、求面积

提示 :

- 1) 三个子类各自有其属性 , 圆 : radius半径 矩形 : length长 width宽 正方形 : length边长;
 - 2) 父类中的周长(girth)和面积(area)的方法统一返回0.0 , 在三个子类中重写两个方法
- 圆 周长公式 : $3.14 * 2 * r$ 面积公式 : $3.14 * r * r$
矩形 周长公式 : $(length + width) * 2$ 面积公式 $length * width$
正方形 : 周长公式 $length * 4$ 面积公式 $length * length$

15. (封装、继承、super) 某公司的雇员分为以下若干类 :

Employee : 这是所有员工总的父类 , 属性 : 员工的姓名,员工的生日月份。方法 : getSalary(intmonth) 根据参数月份来确定工资 , 如果该月员工过生日 , 则公司会额外奖励100 元。

SalariedEmployee : Employee 的子类 , 拿固定工资的员工。属性 : 月薪

HourlyEmployee : Employee 的子类 , 按小时拿工资的员工 , 每月工作超出160小时的部分按照1.5 倍工资发放。属性 : 每小时的工资、每月工作的小时数

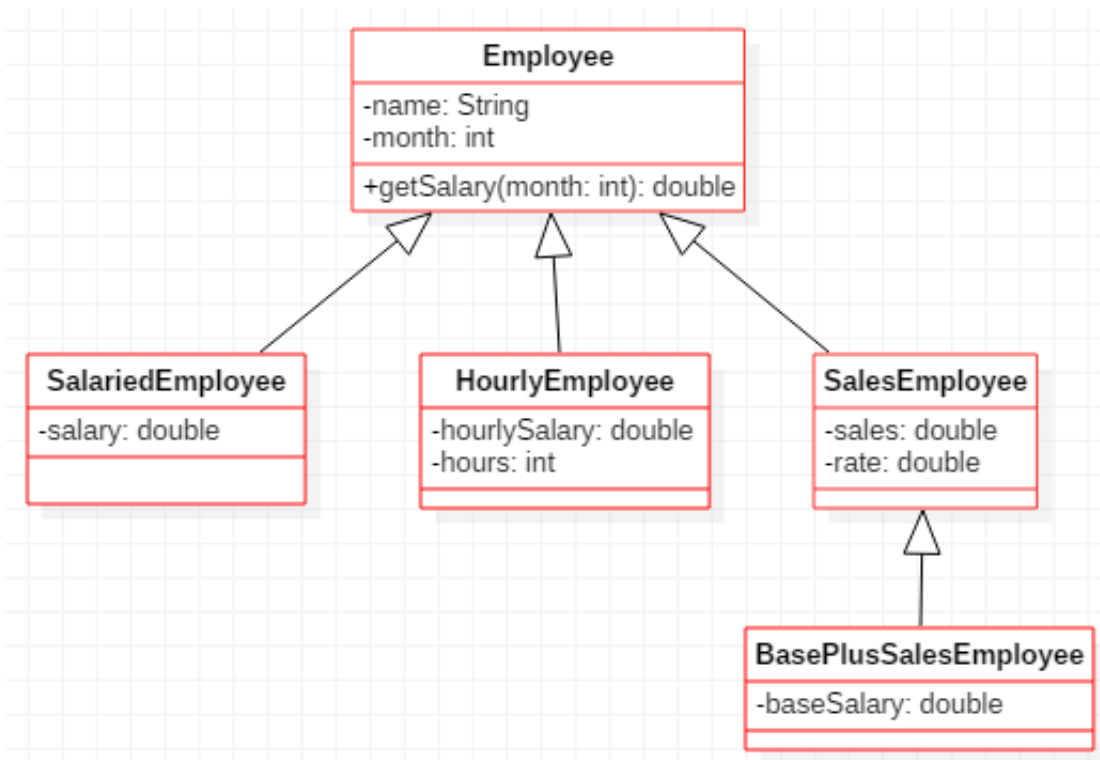
SalesEmployee : Employee 的子类 , 销售人员 , 工资由月销售额和提成率决定。属性 : 月销售额、提成率

BasePlusSalesEmployee : SalesEmployee 的子类 , 有固定底薪的销售人员 , 工资由底薪加上销售提成部分。属性 : 底薪。

根 据 要 求 创 建 SalariedEmployee 、 HourlyEmployees 、 SaleEmployee 和 BasePlusSalesEmployee四个类的对象各一个 , 并计算某个月这四个对象的工资。

注意 : 要求把每个类都做成完全封装 , 不允许非私有化属性。

类图如下 :



16、

Java是（ ）（单|多）继承的语言，适用关键字（ ）定义2个类间的继承关系，子类和父类必须满足（ ）(is a | like a | has a)的关系

17、写出4种访问修饰符，并写出各自修饰的属性及方法的使用范围，以及是否可以被继承到！！

19、

设计一个学生类Student,包括的属性有姓名name，年龄age，学位degree。
由学生类 派生出本科生类Undergraduate和研究生类Graduate2个子类，
本科生类包含的属性有专业specialty，
研究生类包括的属性有研究方向studyDrection。
每个类都有相关数据的输出方法say，并完成对属性的封装。最后在一个测试类中对设计的类进行测试。创建各自类的对象，并调用say方法
要求测试结果如下：

姓名：王军
年龄：23
学位：本科
专业：工业自动化

姓名：刘君
年龄：27
学位：硕士
研究方向：网络技术

