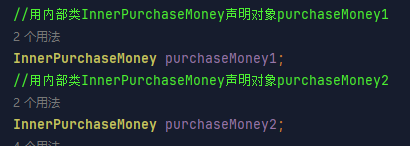
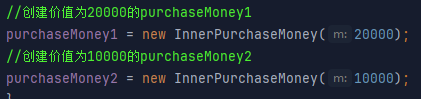
**第N次作业**

上机实践

（1）

1. class MobileShop{
2. *//用内部类InnerPurchaseMoney声明对象purchaseMoney1*
3. InnerPurchaseMoney purchaseMoney1;
4. *//用内部类InnerPurchaseMoney声明对象purchaseMoney2*
5. InnerPurchaseMoney purchaseMoney2;
6. private int mobileAmount;
7. MobileShop(){
8. *//创建价值为20000的purchaseMoney1*
9. purchaseMoney1 = new InnerPurchaseMoney(20000);
10. *//创建价值为10000的purchaseMoney2*
11. purchaseMoney2 = new InnerPurchaseMoney(10000);
12. }
13. void setMobileAmount(int m){
14. mobileAmount=m;
15. }
16. int getMobileAmount(){
17. return mobileAmount;
18. }
19. class InnerPurchaseMoney{
20. int moneyValue;
21. InnerPurchaseMoney(int m){
22. moneyValue=m;
23. }
24. void buyMobile(){
25. if(moneyValue>=20000){
26. mobileAmount -=6;
27. System.out.println("用价值"+moneyValue+"的内部购物券买了6部手机");
28. } else if (moneyValue<20000&&moneyValue>=10000) {
29. mobileAmount -=3;
30. System.out.println("用价值"+moneyValue+"的内部购物券买了3部手机");
31. }
32. }
33. }
34. }
35. public class newYear {
36. public  static void main(String[] args){
37. MobileShop shop= new MobileShop();
38. shop.setMobileAmount(30);
39. System.out.println("手机店现在有"+shop.getMobileAmount()+"部手机");
40. shop.purchaseMoney1.buyMobile();
41. shop.purchaseMoney2.buyMobile();
42. System.out.println("手机店现在有"+shop.getMobileAmount()+"部手机");
43. }
44. }





（2）

1. class Goods{
2. boolean isDanger;
3. String name;
4. Goods (String name){
5. this.name = name;
6. }
7. public void setIsDanger(boolean boo){
8. isDanger = boo;
9. }
10. public boolean isDanger(){
11. return isDanger;
12. }
13. public String getName(){
14. return name;
15. }
16. }
17. class DangerException extends Exception{
18. String messages;
19. public DangerException(){
20. messages = "这是危险品";
21. }
22. public void toShow(){
23. System.out.println(messages+" ");
24. }
25. }
26. class Machine{
27. public void checkBag(Goods goods) throws DangerException{
28. if(goods.isDanger){
29. DangerException danger = new DangerException();
30. *//抛出danger*
31. throw danger;
32. }
33. }
34. }
35. public class check {
36. public static void main(String[] args) {
37. Machine machine = new Machine();
38. Goods apple = new Goods("苹果");
39. apple.setIsDanger(false);
40. Goods explosive = new Goods("炸药");
41. explosive.setIsDanger(true);
42. try{machine.checkBag(explosive);
43. System.out.println(explosive.getName()+"被允许通过");
44. }
45. catch (DangerException e){
46. *//e调用toShow方法*
47. e.toShow();
48. System.out.println(explosive.getName()+"被禁止通过");
49. }
50. try {
51. machine.checkBag(apple);
52. System.out.println(apple.getName()+"被允许通过");
53. }
54. catch (DangerException e){
55. e.toShow();
56. System.out.println(apple.getName()+"被禁止通过");
57. }
58. }
59. }





1. 内都类的外嵌类的成员变量在内部类中仍然有效吗?

**在Java编程中，内部类可以访问其外部类的所有成员，包括成员变量和方法，即使内部类实例是在外部类的作用域之外创建的。这意味着，如果你在外部类中定义了一个成员变量，那么这个变量对于内部类来说是可见并且可访问的。**

1. 内部类中的方法也可以调用外嵌类中的方法吗？

**可以，内部类不仅可以访问外部类的成员变量，还可以直接调用外部类的所有方法，无论这些方法的访问修饰符如何（private、protected、default或public）。**

1. 内部类的类体中可以声明类变量和类方法吗？

**内部类可以声明实例变量和实例方法，但不能声明类变量（也称为静态变量或静态成员）和类方法（静态方法）。这是因为内部类本身是依赖于外部类实例的，因此它没有自己的静态上下文。**

5、请说出下列程序的输出结果。

1. public class cry
2. {
3. public void cry(){
4. System.out.println("大家好");
5. }
6. }
7. class cryTest {
8. public static void main(String[] args) {
9. cry hello = new cry(){
10. public void cry(){
11. System.out.println("大家好，我是派大星");
12. }
13. };
14. hello.cry();
15. }
16. }

