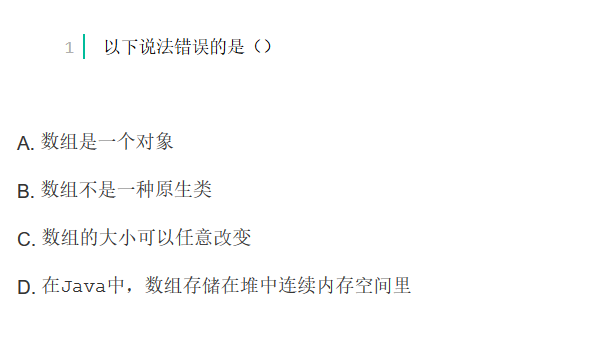
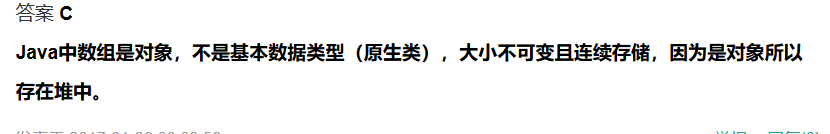


解析：

Java虽然是单继承，但继承具有传递性，所以子类，孙子类的对象可以赋值给父类的引用。Object obj = new everything()

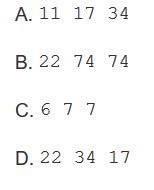


解析：



Test.main() 函数执行后的输出是（）

1. **public** **class** Test {
2. **public** **static** **void** main(String [] args){
3. System.out.println(**new** B().getValue());
4. }
5. **static** **class** A{
6. **protected** **int** value;
7. **public** A(**int** v) {
8. setValue(v);
9. }
10. **public** **void** setValue(**int** value){
11. **this**.value = value;
12. }
13. **public** **int** getValue(){
14. **try**{
15. value++;
16. **return** value;
17. } **catch**(Exception e){
18. System.out.println(e.toString());
19. } **finally** {
20. **this**.setValue(value);
21. System.out.println(value);
22. }
23. **return** value;
24. }
25. }
26. **static** **class** B **extends** A{
27. **public** B() {
28. **super**(5);
29. setValue(getValue() - 3);
30. }
31. **public** **void** setValue(**int** value){
32. **super**.setValue(2 \* value);
33. }
34. }
35. }

答案：D

首先，super()函数指的是调用父类的构造方法

①

new B()

执行B的构造函数，第一行是super(5);

此时执行的是A的构造函数，A的构造函数调用的是setValue()方法，由于B重写了A的这个方法，

所以！！！执行的是B的 setValue()方法。

即传入的参数是2\*5=10

此时，因为super，所以调用的是父类的 setValue()方法，即value=10

第一行执行完毕。

第二行是 setValue(getValue()-3);

B没有getValue()方法，故执行父类的此方法，

try返回的是value=10+1=11，保存在临时栈中

finally中调用this的方法，这个this指的是B的对象，又重写，故就是B的 setValue()方法

value=2\*11=22，第一个打印到屏幕上的数字

接下来参数 getValue()-3=11-3=8

传入B的 setValue()方法

此时value=2\*8=16

至此，new B()执行结束

②

new B(). getValue()

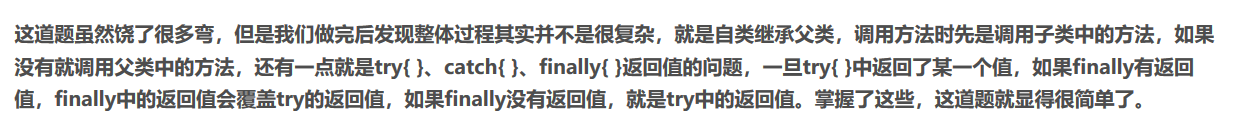
B没有 getValue()，故执行A的 getValue()

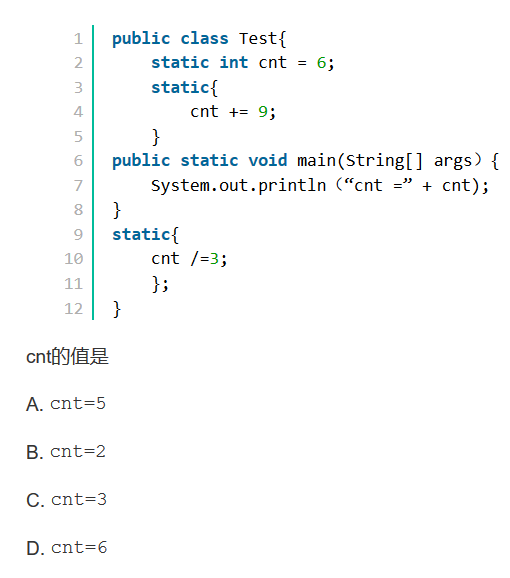
try返回16+1=17，保存到临时栈中

finally调用B的 setValue()方法

value=17\*2=34，第二个打印到屏幕上面的数字

最后主函数打印返回值，也就是try保存到临时栈的17





**解析：静态变量，静态代码块是按代码顺序依次执行。**