第9章 异常处理

# 判断题

1. 没有编译错误的程序在运行时不会产生异常。( F ) //p155 语义错
2. 异常处理结构中可有多个catch子句，应将捕获父类型异常的catch放在捕获子类型异常的catch之前。( F ) //p162 说明（3）
3. 一个catch可以捕获多种有继承关系的异常，也可以捕获多种无关系的异常( T )。
4. 不会产生异常的语句不可以放在try块中。( F )。
5. try块中一旦某个位置产生异常，其后的语句都不会执行。( T ) //p160 最后一段
6. catch块中的语句不可以产生异常。( F )
7. 当try块中发生的异常与某个catch块的()内类型相同时，该异常对象的引用赋给catch块的()内声明的变量。( T ) //p162 “说明”上面的一段
8. finally块中的语句无论是否产生异常通常都会被执行。( T ) //p160 最后一段
9. 声明main()抛出异常可以纠正某些编译错误，但不能解决任何运行异常。( T )
10. 应该尽量避免异常，例如数组越界和除0，而不是异常发生了再处理。( T )
11. try块中不能出现throw语句。( F ) //p164 例题9.3
12. 用于声明方法抛出异常的是throw关键字。( F ) //p163 倒数8-4行
13. 能够用throw抛出的必须是Throwable类或其子类对象。( T ) //p163 倒数9行
14. 没有出现throw关键字的语句就不会产生异常。( F ) //p161例题9.2 编号12行
15. 方法声明抛出的异常可以是方法体实际抛出异常的子类。( F ) //p163 倒数8-4行
16. 异常处理的主要作用是当异常发生时通过纠正异常使程序继续运行而不终止。( T ) //p156 第5段（2）异常
17. 如果一个方法可能抛出多种异常，则可以声明该方法抛出Exception类异常。( T )
18. 子类中重写的方法不能比父类的方法抛出更多类型的异常。( T ) //p123倒数7-2行
19. 一个try块之后只能对应一个catch块。( F ) //p161例题9.2
20. 通常一个方法的方法体不会既抛出异常又处理该异常。( T ) //p161例题9.2

# 选择题

1. 以下不是使用try-catch进行异常处理的机制和意义的是( B )。
2. 将发生异常和处理异常进行分离
3. 使程序能在发生异常时停止执行
4. 被调方法将异常通知给主调方法
5. 主调方法接收到异常后处理异常
6. 以下通常属于异常处理应用范围的是( D )。
7. 处理编译错误
8. 处理逻辑错误
9. 处理程序运行中的致命错误Error
10. *处理程序运行中的异常Exception*
11. 关于异常类，以下说法错误的是( B )。
12. 异常类都是Throwable的子类
13. 不能自定义异常类
14. 异常类有参数为String型异常信息字符串的构造方法和无参数构造方法
15. 异常类有getMessage()和toString()方法返回异常信息
16. 关于抛出异常，以下说法正确的是( A )。
17. 方法不处理异常时将异常抛给方法的调用者
18. 所有异常只能通过throw关键字主动抛出
19. 只要方法可能抛出异常就一定要通过throws关键字声明抛出异常
20. main()方法不能声明抛出异常否则会有编译错误
21. 下列关于try-catch-finally结构的描述中，错误的是( C )。
22. try块中是可能抛出异常的代码，若某条语句抛出了异常，则其后续语句不再继续执行，而是转到catch进行异常类型匹配
23. 一个try块可以对应多个catch块，用于对多个异常类进行捕获
24. 一个try块只能对应一个catch块，但可对应多个finally块
25. finally语句块一般总会执行
26. 下列关于try和catch的描述中错误的是( A )。
27. 不同的catch代码段是不同的作用域，但可以互相访问其中的局部变量
28. 如果没有异常产生，则所有的catch代码段都被略过不执行
29. try代码段后至少要有一个catch代码段
30. 异常总是由距离产生异常位置最近的匹配的catch代码段处理
31. 当方法遇到异常又不知如何处理时应该( B )。
32. 捕获异常
33. 抛出异常
34. 声明异常
35. 嵌套异常
36. 下列关于catch的描述中错误的是( D )。
37. 一个catch后的()内可有多个用|分隔的异常类捕获多种异常并做统一处理
38. 如果catch后的()内声明捕获父类异常，则该catch块可以捕获子类型异常
39. catch块中也可以抛出异常，甚至可以将捕获到的异常再次抛出
40. 如果两个catch块要捕获的异常有继承关系，捕获父类异常的catch块应在前面
41. 以下程序运行时抛出( C )类型的异常。

String s = null;

System.out.println(s.length());

1. ArithmeticException
2. EOFException
3. NullPointerException
4. IOException
5. 以下程序运行时抛出( D )类型的异常。

int a[] = new int[3];

a[3] = 0;

1. ArithmeticException
2. EOFException
3. NullPointerException
4. ArrayIndexOutOfBoundsException
5. 以下程序运行时抛出( A )类型的异常。

String s = "abc";

System.out.println(s.charAt(3));

1. StringIndexOutOfBoundsException
2. EOFException
3. NullPointerException
4. IOException
5. 以下程序运行时抛出( A )类型的异常。

int a = 1/0;

1. ArithmeticException
2. EOFException
3. NullPointerException
4. ArrayIndexOutOfBoundsException
5. 以下程序运行时输入字母a，则会抛出( B )类型的异常。

Scanner scan = new Scanner(System.in);

int a = scan.nextInt();

1. ArithmeticException
2. InputMismatchException
3. NumberFormatException
4. ArithmeticException
5. 以下程序运行时输入a，则会抛出( C )类型的异常。

Scanner scan = new Scanner(System.in);

int a = Integer.parseInt(scan.nextLine());

1. ArithmeticException
2. InputMismatchException
3. NumberFormatException
4. ArithmeticException
5. 语句System.out.println(Double.parseDouble(" 3 4 "));的运行结果是( B )。
6. 34
7. 抛出NumberFormatException异常
8. 抛出InputMismatchException异常
9. 3
10. 自定义的异常类通常继承( A )。
11. Exception
12. Throwable
13. Error
14. RuntimeException
15. 以下异常中，方法即使不处理也不必声明抛出的是( D )及其子类。
16. Exception
17. Error
18. RuntimeException
19. B和C
20. 以下程序的运行结果是( A )。

public class Test {

static void throwMethod() {

System.out.println("Inside throwMethod.");

throw new Exception ("demo");

}

public static void main(String args[]) {

try {

throwMethod();

} catch (Exception e) {

System.out.println("Caught " + e);

}

}

}

1. 编译错误
2. 运行时抛出异常Exception
3. Inside throwMethod.
4. Caught Inside throwMethod.
5. 关于Java语言异常处理的说法中错误的是( C )。
6. 把各种不同的异常事件进行分类，体现了良好的继承性
7. 把错误处理代码从常规代码中分离出来
8. 可以用异常处理机制替代传统的控制流程
9. 对具有动态运行特性的复杂程序提供了强有力的控制
10. 调用异常类对象的( D )方法可以获取异常信息字符串。
11. toString()
12. getMessage()
13. getLocalizedMessage()
14. 以上方法都可以
15. 如果方法体可能抛出IOException类异常，则以下说法正确的是( D )。
16. 方法如果不处理此异常，则必须声明抛出此异常
17. 方法如果不声明抛出此异常，则必须用try-catch处理异常
18. 如果方法声明抛出此异常，则调用这个方法的方法也必须处理或声明此异常
19. 以上说法都正确
20. 以下程序的横线处可以填写的是( D )。

import java.io.\*;

public class Test {

public static void main(String args[]) throws {

fun1();

fun2();

}

public static void fun1() throws RuntimeException {

throw new RuntimeException();

}

public static void fun2() throws IOException {

throw new IOException();

}

}

1. IOException
2. IOException, RuntimeException
3. Exception
4. 以上都可以
5. 以下不属于RuntimeException的子类的是( D )。
6. ArithmeticException
7. NumberFormatException
8. NullPointerException
9. IOException

**三、写出程序的运行结果**

1） 例题9.2

2） 例题 9.4