# 第7章

### 选择题

1. C
2. A
3. D

若运行该程序时的参数个数为0，则第8行捕捉到算术异常，A选项会发生，然后跳过其余catch语句块，程序结束；若参数个数大于0，则第10行必会发生数组下标越界异常，B选项正确；第11行代码不可能运行，C选项错误；catch语句块不会同时运行，D选项错误。

1. D
2. A

try语句块后可以跟catch或finally语句块，A选项正确；B选项catch后没有声明捕捉的异常类型；C选项中，捕捉的异常子类应放在异常父类之前；D选项，理由同A，try语句块不能单独出现。

1. C

捕捉到异常后，finally语句块始终执行。

1. AD

try中代码正常执行后，finally语句块始终执行。

### 简答题

1. try、catch、finally用于捕捉并处理异常，try语句中是可能出现异常的代码，catch定义了异常发生时的处理逻辑，finally语句无论异常是否发生都会执行。throw、throws用于抛出异常，throw语句产生一个异常对象并向上抛出，如方法中可能出现异常但又不想使用try-catch-finally立即处理，则可以在方法声明的末尾添加throws语句将异常交给上层代码处理。
2. 程序运行结果：

034

解析：try语句块中会抛出算术异常ArithmeticException，被捕获处理后进入finally语句块执行，finally执行后继续执行try-catch-finally以后的代码。

1. Java的异常处理分为抛出异常和捕获异常两个步骤：
   1. 当程序运行违反语义规则时，将会抛出（throws）异常，异常信息被封装进一个异常对象。若抛出的异常是受检查的编译期异常，则必须捕获处理或继续声明抛出（throws），否则对于不受编译器检查的运行期异常，代码可不做处理。
   2. 异常抛出后，Java运行时系统会沿着方法的调用栈反向查找异常处理代码，这个过程称为捕获（catch）异常。
2. 当需要捕获多个可能发生的异常时，异常子类的catch语句应早于异常父类的catch语句，否则编译系统会报错。因为当异常发生时，Java会按照catch语句的出现顺序依次查找匹配的异常类型，若父异常类型先匹配到，则自异常类型的代码永远不会被执行。