

Nachdenkzettel: Exceptions

1. Was ist falsch mit folgendem Code? Benennen Sie die Fehler und beheben Sie diese.

```
public void doSomething() {
    try {
        lock(); // lock some resource
        // open some resource
        // try to change some things
        // fool around a bit
    } catch (Exception e) {

    } catch (ConcurrentModificationException e) {
        System.out.println("bad stuff going on today!") }
    finally {
        return;
    }
}
```

2. Was ist die Ausgabe des folgenden Programms?

```
public class TestException1 {

    public static void main(String[] args) {
        try {
            badMethod();
            System.out.print("A"); }
        catch (Exception ex) {
            System.out.print("B"); }
        finally {
            System.out.print("C");
        }

        System.out.print("D");

    }

    public static void badMethod() {
        throw new Error();
    }
}
```

3. Wann sollten Sie einen re-throw einer `Exception` implementieren?

4. In einem Web-Shop wird zur Laufzeit festgestellt, dass eine Kunden-ID nicht in der Datenbank vorhanden ist. Ist dies

- eine System-Exception?
- eine Custom-Exception?
- keine Exception?

Warum?

5. Was ist der Vorteil von Exceptions gegenüber dem Auswerten von Fehlerwerten im Return?