

Kontrollfragen A

1. Welche Phasen sind beim Exception Handling zu unterscheiden?
Definieren, Signalisieren, Auslösen, Behandeln
2. Was ist die Hauptaufgabe des eigentlichen Exception Handlers?
Behandeln der Exception
3. Schreiben Sie einen korrekten try-catch Block für einen Aufruf der Methode `parseInt(s, 16)`;

```
int zahl = 0;
try {
    zahl = Integer.parseInt(s, 16);
    System.out.println(zahl);
}
catch (NumberFormatException ex) {
    System.out.println("Der Wert " + s + " ist KEINE gültige HEX-Zahl!");
}
```
4. Welche Nachteile haben die normalen Kontrollstrukturen bei der Fehlerbehandlung?
unübersichtliche, viele/tief verschachtelte if's, Verwendung eines Rückgabewerts
5. Nennen Sie die fünf Java Schlüsselwörter für die Ausnahmebehandlung?
throws, throw, try, catch, finally
6. Beschreiben Sie jedes der fünf Schlüsselwörter aus Frage 5 in ein bis zwei Sätzen.
throws: Exception signalisieren und weiterleiten
throw: Exception auslösen
try: Operation „versuchen“, aufrufen
catch: Fehler abfangen und behandeln
finally: „Aufräumaktionen“ ausführen (fakultativ)

Kontrollfragen B

1. Worin unterscheiden sich "checked" und "unchecked" Exception?
Checked Exceptions: müssen behandelt, d.h. abgefangen werden
Compiler überprüft ob sie abgefangen werden)
Unchecked Exceptions: können freiwillig behandelt werden, müssen aber nicht behandelt werden
2. Was geschieht, wenn eine "unchecked" Exception auftritt, aber in der Methode nicht behandelt wird?
Nicht abgefangene Exceptions propagieren bis zum zuständigen Exceptionhandler, im Extremfall bis zur JVM.
Ggf. App-Crash...
3. Wie finden ProgrammierInnen heraus, ob eine Methode eine Ausnahmebehandlung auslösen kann?
In der API Dokumentation (rtfm ;-))