



Abgabe für Rückmeldungen bis zum 25.01.2021 (08:00 Uhr), Besprechung ab dem 25.01.2021.

Aufgabe 09-0 (Exceptions - Typen)

Betrachten Sie folgendes Listing.

```
1 int[] a = { 1, 2, 3 };  
2 a[0] = Integer.parseInt("hallo");
```

Dieser Code wirft bei seiner Ausführung eine `NumberFormatException` (siehe [Handbuch](#)). Entscheiden und begründen Sie, ob folgende `catch`-Anweisungen diese Exception abfangen.

0. `catch (Exception e)`
1. `catch (IndexOutOfBoundsException e)`
2. `catch (IllegalArgumentException e)`

Hinweis: Eine `catch`-Anweisung fängt nicht nur die angegebene Exception-Klasse, sondern auch alle ihre vererbten Klassen ab.

Aufgabe 09-1 (Exceptions - try...catch)

Betrachten Sie folgendes Listing.

```
1 public class Main {  
2     static int[] a = { 1, 2, 3, 4 };  
3  
4     public static void main(String[] args) {  
5         try {  
6             for (int i = 0; i < a.length + 1; i++) {  
7                 int z = a[i];  
8             }  
9         } catch (IndexOutOfBoundsException e) {  
10            System.out.println("catch \"IndexOutOfBoundsException\");  
11        } catch (Exception e) {  
12            System.out.println("catch \"Exception\");  
13        } finally {  
14            System.out.println("finally");  
15        }  
16    }  
17 }
```

Geben Sie an und begründen Sie, was bei jeder der folgenden Aktionen auf der Konsole ausgegeben wird. Fangen Sie jedes Mal mit dem ursprünglichen Quelltext an.

0. Kompilieren und führen Sie das Programm aus.
1. Kommentieren Sie Zeile 9 und 10 aus, kompilieren und führen Sie das Programm aus.
2. Kommentieren Sie Zeile 13 und 14 aus, kompilieren und führen Sie das Programm aus.

Aufgabe 09-2 (Zeichenübersetzer)

Schreiben Sie ein Programm, welches eine Datei `datei` einliest, alle Leer- und Tabulatorzeichen entfernt und das Ergebnis in eine Ausgabedatei `datei.stripped` schreibt.

Präsenzaufgabe 09-3 (Zufallszahlen)

Schreiben Sie eine statische Funktion mit der Signatur `int[] rand_million()`, die einen Array der Länge 10^6 erzeugt, diesen mit Zufallswerten aus $\{0, \dots, 99\}$ befüllt und zurückgibt. Nutzen Sie hierzu die Funktion `Math.random()` aus dem Package `java.lang` (siehe Handbuch).

Präsenzaufgabe 09-4 (Wrapperklassen)

Das Package `java.lang` enthält sogenannte „Wrapperklassen“ für jeden elementaren Datentyp (i.e. `Byte`, `Short`, `Integer`, `Long`, `Float`, `Double`, `Boolean`, `Character`), die auch entsprechend initialisiert werden können und eine Reihe von Konstanten und Methoden für ihren jeweiligen Datentyp enthalten.

Nutzen Sie die Wrapperklassen, um für jeden elementaren Datentyp (bis auf `boolean`) den kleinsten und größten Wert und die Anzahl der Bits in der Kommandozeile auszugeben.

Bemerkung: Wenn Sie die Klassen im Handbuch (z.B. `Integer`) nachschlagen, werden Sie feststellen, daß die Konstruktoren der Wrapperklassen deprekiert sind (im Quelltext mit der `@Deprecated`-Annotation).

Diese Klassen werden nicht wie normale Klassen, sondern über eine Sprachfunktion namens „Autoboxing“ initialisiert. Kurzum gesagt muß die Klasse nicht explizit mit `new` initialisiert werden und die implizite Typumwandlung zwischen elementarem Datentyp und Wrapperklasse ist möglich. Betrachten Sie hierzu als Beispiel das folgende Listing.

```
1 Character a = 'Q';
2 char b = a;
3 Character c = b;
```