



Ausnahmebehandlung

Lösungen zu den
Übungsaufgaben

```
Funktion funktion = new Funktion();  
try{  
    funktion.auswerten(23.0);  
} catch (InvalidNumberException exception){  
    System.out.println("Ungültige Zahl!");  
}
```

Übung: Eigene Exception



```
/**
 * Gibt an, dass ein Spannungsschwellwert unterschritten wurde.
 */
public class SpannungsUnterversorgungException
    extends Exception {
    /**
     *
     */
    private static final long serialVersionUID =
        -4338933573569787558L;

    /**
     * Konstruktor.
     */
    public SpannungsUnterversorgungException(
        double spannungsSchwellwert,
        double aktuellerSpannungswert) {
        super(
            String
                .format(
                    "Aktueller Spannungswert %.2f unter Schwellwert %.2f.",
                    aktuellerSpannungswert,
                    spannungsSchwellwert));
    }

    /**
     * Programmeinstieg.
     */
    public static void main(String[] args) {
        try {
            throw new SpannungsUnterversorgungException(
                3.3, 2.9);
        } catch (SpannungsUnterversorgungException exception) {
            System.out.println("Exception: "
                + exception.getMessage());
        }
    }
}
```

```
TextDatei textDatei =  
    new TextDatei();  
try {  
    textDatei.oeffnen();  
    textDatei.zeileLesen();  
  
} catch (FileNotFoundException fileNotFoundException) {  
    System.out  
        .println("Datei nicht gefunden.");  
} catch (IOException ioException) {  
    System.out  
        .println("Fehler beim Dateizugriff.");  
} finally {  
    textDatei.schliessen();  
}
```

Übung: Exception werfen



```
/**
 * Berechnet das Divisionsergebnis aus zaehler und nenner. Wirft eine
 * Exception, wenn der nenner 0 ist.
 */
public double division(
    double zaehler, double nenner)
    throws DivisionDurchNullException {
    if (Math.abs(nenner) < 1e-5) {
        throw new DivisionDurchNullException();
    }
    return zaehler / nenner;
}

/**
 * Programmeinstieg.
 */
public static void main(String[] args) {
    Division division = new Division();
    try {
        System.out.println(division
            .division(23, 43));
        System.out.println(division
            .division(23, 0));
    } catch (DivisionDurchNullException e) {
        System.out
            .println("Fehler: Division durch 0");
    }
}
```

```
/**
 * Eigene Exception, die angibt, dass eine Division durch 0 versucht wurde.
 */
public class DivisionDurchNullException
    extends Exception {

    /**
     * Konstruktor.
     */
    public DivisionDurchNullException() {
        super("Division durch 0");
    }

    /**
     *
     */
    private static final long serialVersionUID =
        -5227986757424771222L;
}
```

```
/**
 * Dummy-Klasse für die Übung des Zugriffs auf eine Datei.
 */
public class TextDatei {

    /**
     * Logger-Objekt.
     */
    private final Logger logger;

    public TextDatei() {
        logger = Logger.getLogger(TextDatei.class.getName());
        logger.setLevel(Level.ALL);
        FileHandler fileHandler = null;
        try {
            fileHandler = new FileHandler("logdatei.txt");
        } catch (SecurityException e) {
            logger.log(Level.SEVERE,
                "Fehler beim Anlegen der Log-Datei", e);
        } catch (IOException e) {
            logger.log(Level.SEVERE,
                "Fehler beim Anlegen der Log-Datei", e);
        }
        logger.addHandler(fileHandler);
    }

    /**
     * Öffnen der Datei.
     */
    public void oeffnen() throws FileNotFoundException {
        logger.log(Level.INFO, "Datei geöffnet.");
    }

    /**
     * Lesen einer Zeile.
     */
    public void zeileLesen() throws IOException {
        if (Math.random() < 0.4) {
            // Per Zufall wird hier und da die Exception ausgelöst.
            IOException exception = new IOException();
            logger.log(Level.SEVERE, "zeileLesen", exception);
            throw exception;
        }
        logger.log(Level.INFO, "Zeile gelesen.");
    }
}
```

```
    /**
     * Schliessen der Datei.
     */
    public void schliessen() {
        logger.log(Level.INFO, "Datei geschlossen.");
    }

    public static void main(String[] args) {
        TextDatei textDatei = new TextDatei();
        try {
            textDatei.oeffnen();
            textDatei.zeileLesen();

        } catch (FileNotFoundException fileNotFoundException) {
            System.out.println("Datei nicht gefunden.");
        } catch (IOException ioException) {
            System.out.println("Fehler beim Dateizugriff.");
        } finally {
            textDatei.schliessen();
        }
    }
}
```