

Elektronikschule Tettnang	Softwareentwicklung	Datum:	Klasse:
	Fehlerbehandlung in Java	Name:	

## Fehlerarten bei Software:

Beim Programmieren lassen sich grundsätzliche drei verschiedene Fehlerarten unterscheiden:

### 1. Compilerfehler (Syntaxfehler)

Der Compiler versucht den vom Menschen geschriebenen Programmcode in Maschinencode zu übersetzen. Dies kann er jedoch nur, wenn die Syntax der Programmiersprache eingehalten wird.

Beispiele:

### 2. Laufzeitfehler

Der Compiler übersetzt den Programmcode richtig. Nach dem Programmstart können aber Laufzeitfehler entstehen. Sie können vom Compiler nicht erkannt werden, da sie durch Ereignisse ausgelöst werden, die während dem Kompilieren noch nicht bekannt sind.

Beispiele:

### 3. Logische Fehler (Semantische Fehler)

Diese Fehler sind keine Laufzeitfehler, sondern Fehler im Algorithmus, d.h. im logischen Ablauf des Programms. Der Compiler übersetzt den Programmcode richtig und während der Laufzeit gibt es keine Laufzeitfehler. Jedoch ist das Ergebnis (die Semantik) des Programms trotzdem falsch.

Beispiele:

Elektronikschule Tettnang	Softwareentwicklung	Datum:	Klasse:
	Fehlerbehandlung in Java	Name:	

## Exceptions

Exceptions (Ausnahmen) sind Objekte vom Typ „java.lang.Exception“, die erzeugt werden, wenn ein Laufzeitfehler auftritt. Man sagt auch „eine Exception wird geworfen“.

Nun kann **an anderer Stelle** das erzeugte Exception-Objekt abgefangen werden und somit der Fehler behandelt werden. Der Fehler kann z.B. weitergereicht werden an die aufrufende Methode und dort entschieden werden, wie sich das Programm verhält.

Betrachten Sie folgende Fehlerbehandlung ohne Exceptions. Welche Nachteile ergeben sich?

```
public class Kreis {
    private double radius;

    public void setRadius(double radius) {
        if(radius <= 0) {
            System.out.println("FEHLER: Der Radius muss positiv sein.");
        }
        else {
            this.radius = radius;
        }
    }
}

public class Test {
    public static void main(String[] args) {
        Kreis k = new Kreis();
        k.setRadius(-1);
    }
}
```

Und nun die Fehlerbehandlung mit Hilfe einer Exception:

```
public class Kreis {
    private double radius;

    public void setRadius(double radius) throws Exception {
        if(radius <= 0) {
            throw new Exception(); // Hier wird die Exception geworfen
        }
        else {
            this.radius = radius;
        }
    }
}

public class Test {
    public static void main(String[] args) {
        Kreis k = new Kreis();

        try { // Hier beginnt der kritische Abschnitt
            k.setRadius(-1);
        }
        catch(Exception e) { // Hier wird die Exception gefangen
            System.out.println("FEHLER: Der Radius muss positiv sein.");
        }
    }
}
```

Elektronikschule Tett nang	Softwareentwicklung	Datum:	Klasse:
	Fehlerbehandlung in Java	Name:	

## Aufgabe 1

- Schreiben Sie ein Programm, das von der Tastatur eine Zahl in eine int-Variable einliest. Benutzen Sie zunächst keinen try-catch-Block.
  - Ein Laufzeitfehler kann nun auftreten, falls der Benutzer anstatt einer Zahl, einen Buchstaben eingibt. Es wird eine Exception geworfen. Wie lautet der Typ dieser Exception?
- 
- Fangen Sie nun die geworfene Exception mit einem try-catch-Block auf. Gibt der Benutzer keine Zahl ein, wird ein Hinweis „Bitte geben Sie eine Zahl ein“ am Bildschirm angezeigt.
  - Erweitern Sie Ihr Programm, in dem der Benutzer solange nach einer Zahl gefragt wird, bis die Eingabe einer Zahl erfolgt.

## Aufgabe 2 Fehler klassifizieren

Betrachten sie folgenden Programmcode.

```
import java.util.scanner;

public class Tore {
    public static void main(String[] args) {

        scanner kb = new scanner(System.in);
        int tore[] = new int[7];
        int anzSpiele, sum = 0;
        int mTore;

        System.out.println("Das Programm berechnet die mittlere Toranzahl.");
        System.out.print("Anzahl der Spiele eingeben: ");

        anzahl = kb.nextInt();
        System.out.println("Geben Sie zu jedem Spiel die Tore ein!");

        for(int i=0; i < anzahl; i++) {
            System.out.print(i+2+" . Spiel");
            tore[i] = kb.nextInt();
        }

        for(int i=1; i < anzahl; i++) {
            sum = sum - tore[i];
        }
        mTore = sum / anzSpiele;
        System.out.println("Durchschnittliche Tore: " + mTore);
    }
}
```

- Markieren Sie alle Syntaxfehler.
- Markieren Sie in einer anderen Farbe alle Zeilen, in denen Laufzeitfehler auftreten können.
- Beschreiben Sie gefundene logische Fehler.

Elektronikschule Tettnang	Softwareentwicklung	Datum:	Klasse:
	Fehlerbehandlung in Java	Name:	

## Abgeleitete Exceptions

Es ist auch möglich eigens definierte Exceptions zu erstellen, die nur dann geworfen werden wenn ein bestimmter zugehöriger Fehler auftritt. Hierzu erstellt man eine Klasse die von „java.lang.Exception“ erbt.

Beispiel

```
public class KreisException extends Exception {}

public class Kreis {
    private double radius;

    public void setRadius(double radius) throws KreisException {
        if(radius <= 0) {
            throw new KreisException();
        }
        else {
            this.radius = radius;
        }
    }
}

public class Test {
    public static void main(String[] args) {
        Kreis k = new Kreis();

        try {
            k.setRadius(-1);
        }
        catch(KreisException e) {
            System.out.println("FEHLER: Der Radius muss positiv sein.");
        }
    }
}
```

## Einige bekannte abgeleitete Exceptions der Java-Runtime-Klassen

Exception	Beschreibung
NullPointerException	Eine Methode wird auf einem Objekt aufgerufen, das nicht erzeugt wurde.
IndexOutOfBoundsException	Der Index für das Array oder den String ist nicht erlaubt.
NumberFormatException	Der String konnte nicht in eine Zahl konvertiert werden.
FileNotFoundException	Die Datei wurde nicht gefunden.
IOException	Eingabe-/Ausgabefehler.

## Aufgabe 3 PIN-Abfrage mit Exceptions

Erweitern Sie Ihr Programm aus Aufgabe 1 zu einer PIN-Abfrage. Der Benutzer hat drei Versuche, die hinterlegte PIN „2884“ korrekt einzugeben. Benutzen Sie für folgende zwei Fehlerfälle zusätzliche eigene unterschiedliche Exceptions.

- Die PIN wurde in einem Versuch falsch eingegeben.

Elektronikschule Tettnang	Softwareentwicklung	Datum:	Klasse:
	<b>Fehlerbehandlung in Java</b>	Name:	

- Die PIN wurde nach insgesamt drei Versuchen falsch eingegeben und die Eingabe der PUK ist nun erforderlich.