

15.2. Java.nio

Beispiel: Filename, eventuell mit Path, zusammenbauen.

```
var fileName = java.nio.file.Paths.get("zahlen.txt");
```

```
String current = System.getProperty("user.dir");
String home    = System.getProperty("user.home");
String temp    = System.getProperty("java.io.tmpdir");

var fileName   = java.nio.file.Paths.get(home, "zahlen.txt");

var path       = java.nio.file.Paths.get(home);
var fileName   = path.resolve("zahlen.txt"); // anhängen
```

```
java.nio.file.Files.delete(fileName);
java.nio.file.Files.move(fileNameAlt, fileNameNeu);
...
```

ACHTUNG: File darf nicht geöffnet sein!

Beispiel: lesen

```
// File lesend öffnen
var datei = java.nio.file.Files.newBufferedReader(fileName);
...
datei.close(); // schließen des Files
```

Beispiel: schreiben

```
// File schreibend öffnen
var datei = java.nio.file.Files.newBufferedWriter(fileName);
...
datei.close(); // schließen des Files
```

```
datei.write(string)
datei.readLine()

java.nio.file.Files.readAllLines(fileName)
```

Schreibt einen String aufs File.
Liest eine Zeile ein.

Liest alle Zeilen in ein List<String>.

Beispiel: Text-File am Bildschirm ausgeben.

```

public static void main(String[] args) throws java.io.IOException {
    var eingabe = new java.util.Scanner(System.in);
    System.out.print("File Name: ");
    String name = eingabe.nextLine();

    var fileName =
        java.nio.file.Paths.get(System.getProperty("user.dir"), name);

    for(final String line:java.nio.file.Files.readAllLines(fileName)) {
        System.out.println(line);
    }
    // Achtung: KEIN close() bei readAllLines (da es auch kein open gibt)
}

```

Beispiel: Zahlen-File in Array einlesen (1).

```

public static void main(String[] args) throws java.io.IOException {
    var fileName = java.nio.file.Paths.get("zahlen.txt");

    float[] zahlen = new float[100];
    int anz=0;

    for(final String line:java.nio.file.Files.readAllLines(fileName)) {
        zahlen[anz++] = Float.parseFloat(line);
        if(anz >= zahlen.length) {
            break;
        }
    }

    for(int i=0; i<anz; i++) {
        System.out.println(i+1 + ": " + zahlen[i]);
    }
}

```

Beispiel: Zahlen-File in Array einlesen (2).

```

public static void main(String[] args) throws java.io.IOException {
    var fileName = java.nio.file.Paths.get("zahlen.txt");
    var readZahlen = java.nio.file.Files.newBufferedReader(fileName);

    float[] zahlen = new float[100];
    int anz = 0;
    String line;

    while(((line=readZahlen.readLine())!=null) && anz<zahlen.length) {
        zahlen[anz++] = Float.parseFloat(line);
    }
    readZahlen.close();

    for(int i=0; i<anz; i++) {
        System.out.println(i+1 + ": " + zahlen[i]);
    }
}

```

Beispiel: Zahlen-File in Array mit Hilfe von Scanner einlesen (3). Es werden die Vorteile von Scanner genutzt.

```
public static void main(String[] args) throws java.io.IOException {
    var fileName = java.nio.file.Paths.get("zahlen.txt");
    var file = java.nio.file.Files.newBufferedReader(fileName);
    var readZahlen = new java.util.Scanner(file);

    float[] zahlen = new float[100];
    int anz = 0;

    while(readZahlen.hasNext() && anz<zahlen.length) {
        zahlen[anz++] = readZahlen.nextFloat();
    }
    file.close();
    readZahlen.close();

    for(int i=0; i<anz; i++) {
        System.out.println(i+1 + ": " + zahlen[i]);
    }
}
```

Beispiel: Array auf Zahlen-File schreiben.

```
public static void main(String[] args) throws java.io.IOException {
    float[] zahlen = {1,3,5,7};

    var fileName = java.nio.file.Paths.get("zahlen.txt");
    var writeZahlen = java.nio.file.Files.newBufferedWriter(fileName);

    for(final float zahl : zahlen) {
        writeZahlen.write(zahl + "\n");
    }

    writeZahlen.close();
}
```