

Bereich: Input/Output (3)**Zugriff auf eine Textdatei****Musterlösung****Package:** de.dhbwka.java.exercise.io.textfile**Klasse:** TextFile

```
package de.dhbwka.java.exercise.io.textfile;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 */
@SuppressWarnings("serial")
public class LineNumberOutOfBoundsException extends Exception {

    public LineNumberOutOfBoundsException() {
        super();
    }

    public LineNumberOutOfBoundsException(String message) {
        super(message);
    }
}
```

```
package de.dhbwka.java.exercise.io.textfile;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 */
public class TextFile {

    /** File zum Lesen und Schreiben */
    private File f;
    private String[] buffer;
```

```
/** Konstruktor mit File */
public TextFile(File f) {
    this.f = f;
    this.read();
}

/** Konstruktor mit String als Pfadname */
public TextFile(String filename) {
    this(new File(filename));
}

/** Datei (erneut) einlesen. Wirft keine Exception! */
public void read() {
    int zeilen = 0;
    try (BufferedReader in = new BufferedReader(new FileReader(f))) {
        for (; in.ready(); zeilen++)
            in.readLine();
    } catch (IOException e) {
        System.err.println("Fehler beim Lesen der Datei.");
    }
    buffer = new String[zeilen];
    try (BufferedReader in = new BufferedReader(new FileReader(f))) {
        for (int i = 0; i < zeilen; i++)
            buffer[i] = in.readLine();
    } catch (IOException e) {
        System.err.println("Fehler beim Lesen der Datei.");
    }
}

/** Datei schreiben. Wirft Exception! */
public void write() {
    if (buffer != null)
        try (PrintWriter out = new PrintWriter(new FileWriter(f))) {
            for (int i = 0; i < buffer.length; i++)
                out.println(buffer[i]);
        } catch (IOException e) {
            System.err.println("Fehler beim write()");
        }
}

/** Liefert die Anzahl der Zeilen. */
public int availableLines() {
    if (buffer == null)
        return -1;
    return buffer.length;
}

/** Liefert alle Zeilen als Array. */
public String[] getLines() {
    return buffer;
}

/** Liefert die angegebene Zeile. Zählung ab 1. */
public String getLine(int i) throws LineNumberOutOfBoundsException {
    if (buffer == null)
        return null;
    if (i > 0 && i <= buffer.length)
```

```
        return buffer[i - 1];
    else
        throw new LineNumberOutOfBoundsException(
            "Falsche Eingabe bei getLine");
}

/** Ersetzt die angegeben Zeile. Zählung ab 1. */
public void setLine(int i, String s)
    throws LineNumberOutOfBoundsException {
    if (buffer != null && i > 0 && i <= buffer.length)
        buffer[i - 1] = s;
    else
        throw new LineNumberOutOfBoundsException(
            "Falsche Eingabe bei setLine");
}

/** Ersetzt alle Vorkommen von regexp in allen Zeilen durch ersatz */
public void replaceAll(String regexp, String ersatz) {
    if (buffer != null && regexp != null && ersatz != null)
        for (int i = 0; i < buffer.length; i++)
            buffer[i] = buffer[i].replaceAll(regexp, ersatz);
}

/** Löscht den Puffer und das Filehandle */
public void close() {
    buffer = null;
    this.f = null;
}
}
```

```
package de.dhbwka.java.exercise.io.textfile;

import java.io.IOException;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 */
public class TextFileTest {

    public static void main(String[] args) throws IOException,
        LineNumberOutOfBoundsException {
        TextFile a = new TextFile("bla.txt");
        // Anzahl der Zeilen
        System.out.println("Anzahl der Zeilen: " + a.availableLines());
        // Erste Zeile
        System.out.println("Zeile 1: " + a.getLine(1));
        // Fünfte Zeile
        System.out.println("Zeile 5: " + a.getLine(5));
    }
}
```

```
// Alle Zeilen
System.out.println("Alle Zeilen: ");
for(String line : a.getLines())
    System.out.println(line);
// Ersetzen von "meine" durch "unsre"
a.replaceAll("meine", "unsre");
// Datei schreiben
// a.write();
System.out.println("Alle Zeilen nach dem Ersetzen: ");
for(String line : a.getLines())
    System.out.println(line);
a.close();
}
}
```

Bereich: Input/Output (3)**Primzahlen speichern und lesen*****Musterlösung****Package:** de.dhbwka.java.exercise.io**Klasse:** PrimesFile

```
package de.dhbwka.java.exercise.io;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'. Baden-Wuerttemberg
 * Cooperative State University.
 *
 * (C) 2016 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 */
public class PrimesFile {

    public static final int MAX = 100000;
    public static final String PRIM_FILENAME = "primes.txt";

    public static void main(String[] args) {
        savePrimes(getPrimes(MAX), PRIM_FILENAME);
        System.out.println("prime numbers up to " + MAX + " saved to " +
            PRIM_FILENAME);
    }

    public static void savePrimes(boolean[] prim, String filename) {
        try {
            PrintWriter pw = new PrintWriter(new FileWriter(
                new File(filename)));
            for (int i = 0; i < prim.length; i++) {
                if (prim[i])
                    pw.println(i);
            }
            pw.close();
        } catch (IOException e) {
            System.err.println("Error writing numbers to " + filename);
        }
    }

    public static boolean[] getPrimes(int max) {
        boolean[] prim = new boolean[max];
        // initialize array with true
        for (int i = 2; i < prim.length; i++)
            prim[i] = true;
        // eratostenes' sieve
        for (int i = 2; i < prim.length; i++)
            if (prim[i])
                for (int j = i * 2; j < prim.length; j += i)
```

```
        prim[j] = false;

    return prim;
}

package de.dhbwka.java.exercise.io;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Random;

/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2016 by W. Geiger, T. Schlachter, C. Schmitt, W. Suess
 */
public class PrimesTest {

    public static void main(String[] args) {
        boolean[] prim = LoadPrimes(PrimesFile.PRIM_FILENAME);
        if (prim!=null && prim.length>0) {
            Random rnd = new Random();
            // Test for 10 random numbers if they are prime
            for (int i=0; i<10; i++) {
                int number = rnd.nextInt(prim.length);
                System.out.println(number + " is " +
                    (prim[number]?"":"not ") + "prime.");
            }
        }

        public static boolean[] loadPrimes(String filename) {
            boolean[] prim = new boolean[PrimesFile.MAX];
            // read lines using try-with-resources statement
            try (BufferedReader br2 = new BufferedReader(new FileReader(
                new File(filename)))) {
                while (br2.ready()) {
                    prim[Integer.parseInt(br2.readLine())] = true;
                }
            } catch (IOException e) {
                System.err.println("Error reading from "+filename);
            }
            return prim;
        }
    }
}
```