

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4 import java.text.*;
5
6 /**
7  *
8  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
9  */
10 public class WriteTextFile1
11 {
12
13     public static void main(String[] args)
14     {
15         String fileName = "data1.txt";
16         File aFile = new File(fileName);
17         if ( ! aFile.exists() ) {
18             try {
19                 FileWriter aFileWriter = new FileWriter(aFile);
20                 BufferedWriter aBufferedWriter = new BufferedWriter(aFileWriter);
21                 PrintWriter aPrintWriter = new PrintWriter(aBufferedWriter);
22                 aPrintWriter.println("Dies ist die 1. Zeile.");
23                 aPrintWriter.println("Dies ist die 2. Zeile.");
24                 aPrintWriter.println("911");
25                 aPrintWriter.println("12.34");
26                 aPrintWriter.println("000.77");
27                 aPrintWriter.flush(); // Puffer proaktiv "spülen"
28                 aFileWriter.close();
29             } catch (IOException e) {
30                 // ...
31             }
32         }
33     }
34 }

```

```

1
2 package oop6_dat6_ss;
3
4 import java.io.*;
5 /**
6  *
7  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
8  */
9 public class WriteBinaryFile3 {
10     public static void main(String[] args)
11     {
12         String fileName = "data3.bin";
13         File aFile = new File(fileName);
14         if ( ! aFile.exists()) {
15             try {
16                 FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
17                 // DataOutputStream unterstützt elementare Datentypen
18                 DataOutputStream aDataOutputStream = new
19 DataOutputStream(aFileOutputStream);
20                 // 1. Anzahl intWerte, 2. intWerte...
21                 aDataOutputStream.writeInt(2);
22                 aDataOutputStream.writeInt(5);
23                 aDataOutputStream.writeInt(9);
24
25                 // 1. Anzahl floatWerte, 2. floatWerte ...
26                 aDataOutputStream.writeInt(3);
27                 aDataOutputStream.writeFloat(1.23f);
28                 aDataOutputStream.writeFloat(1.111f);
29                 aDataOutputStream.writeFloat(123.123f);
30
31                 // 1. Anzahl CharWerte, 2. CharWerte ...
32                 aDataOutputStream.writeInt(4);
33                 aDataOutputStream.writeChar('a');
34                 aDataOutputStream.writeChar('b');
35                 aDataOutputStream.writeChar('c');
36                 aDataOutputStream.writeChar('d');
37
38                 aFileOutputStream.close();
39             } catch (IOException e) {
40                 // ...
41             }
42         }
43 }

```

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4
5 /**
6  * Binäre Dateien: elementare Werte schreiben
7  *
8  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
9  */
10 public class WriteBinaryFile2
11 {
12
13     public static void main(String[] args)
14     {
15         String fileName = "data2.bin";
16         File aFile = new File(fileName);
17         if ( ! aFile.exists()) {
18             try {
19                 FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
20                 // DataOutputStream unterstützt elementare Datentypen
21                 DataOutputStream aDataOutputStream = new
22 DataOutputStream(aFileOutputStream);
23                 // 1. Anzahl intWerte, 2. intWerte...
24                 aDataOutputStream.writeInt(3);
25                 aDataOutputStream.writeInt(2);
26                 aDataOutputStream.writeInt(3);
27                 aDataOutputStream.writeInt(5);
28                 // 1. Anzahl doubleWerte, 2. doubleWerte ...
29                 aDataOutputStream.writeInt(2);
30                 aDataOutputStream.writeDouble(1.23456789);
31                 aDataOutputStream.writeDouble(1.E3);
32                 aFileOutputStream.close();
33             } catch (IOException e) {
34                 // ...
35             }
36         }
37 }

```

```
1 package oop6_dat6_ss;
2
3 import java.io.*;
4
5 /**
6  *
7  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
8  */
9 public class WriteBinaryFile1
10 {
11
12     public static void main(String[] args)
13     {
14         String fileName = "data1.bin";
15         File aFile = new File(fileName);
16         if ( ! aFile.exists()) {
17             try {
18                 FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
19                 for (int i = 0; i <= 10; i ++ ) {
20                     aFileOutputStream.write(i);
21                 }
22                 aFileOutputStream.close();
23             } catch (IOException e) {
24                 // ...
25             }
26         }
27     }
28 }
```

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4 import java.util.StringTokenizer;
5
6 /**
7  *
8  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
9  */
10 public class ReadTextFile1
11 {
12
13     public static void main(String[] args)
14     {
15         String fileName = "data1.txt";
16         File aFile = new File(fileName);
17         if (aFile.exists()) {
18             try {
19                 FileReader aFileReader = new FileReader(aFile);
20                 BufferedReader aBufferedReader = new BufferedReader(aFileReader);
21                 System.out.println(aBufferedReader.readLine());
22                 System.out.println(aBufferedReader.readLine());
23                 System.out.println();
24                 String line = aBufferedReader.readLine();
25                 while (line != null) { // while not EOF
26                     Double d = Double.valueOf(line);
27                     System.out.println(d);
28                     line = aBufferedReader.readLine();
29                 }
30                 aFileReader.close();
31             } catch (IOException e) {
32                 System.out.println("Exception: " + e.getMessage());
33                 return;
34             }
35         }
36     }
37 }

```

```

1
2 package oop6_dat6_ss;
3 import java.io.*;
4 /**
5  *
6  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
7  */
8 public class ReadBinaryFile3 {
9
10     public static void main(String[] args)
11     {
12         String fileName = "data3.bin";
13         File aFile = new File(fileName);
14         if (aFile.exists()) {
15             try {
16                 FileInputStream aFileInputStream = new FileInputStream(aFile);
17                 DataInputStream aDataInputStream = new DataInputStream(aFileInputStream);
18                 // Einlesen der intWerte
19                 int numberOfInt = aDataInputStream.readInt();
20                 for (int i = 0; i < numberOfInt; i++) {
21                     System.out.println(aDataInputStream.readInt());
22                 }
23                 // Einlesen der floatWerte
24                 int numberOfFloat = aDataInputStream.readInt();
25                 for (int i = 0; i < numberOfFloat; i++) {
26                     System.out.println(aDataInputStream.readFloat());
27                 }
28                 // Einlesen der charWerte
29                 int numberOfChar = aDataInputStream.readInt();
30                 for (int i = 0; i < numberOfChar; i++) {
31                     System.out.println(aDataInputStream.readChar());
32                 }
33                 aFileInputStream.close();
34             } catch (IOException e) {
35                 // ...
36             }
37         }
38     }
39 }

```

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4
5 /**
6  * Binäre Dateien: elementare Werte lesen
7  *
8  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
9  */
10 public class ReadBinaryFile2
11 {
12
13     public static void main(String[] args)
14     {
15         String fileName = "data2.bin";
16         File aFile = new File(fileName);
17         if (aFile.exists()) {
18             try {
19                 FileInputStream aFileInputStream = new FileInputStream(aFile);
20                 DataInputStream aDataInputStream = new DataInputStream(aFileInputStream);
21                 // Einlesen der intWerte
22                 int numberOfInt = aDataInputStream.readInt();
23                 for (int i = 0; i < numberOfInt; i++) {
24                     System.out.println(aDataInputStream.readInt());
25                 }
26                 // Einlesen der doubleWerte
27                 int numberOfDouble = aDataInputStream.readInt();
28                 for (int i = 0; i < numberOfDouble; i++) {
29                     System.out.println(aDataInputStream.readDouble());
30                 }
31                 aFileInputStream.close();
32             } catch (IOException e) {
33                 // ...
34             }
35         }
36     }
37 }

```

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4
5 /**
6  *
7  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
8  */
9 public class ReadBinaryFile1
10 {
11
12     public static void main(String[] args)
13     {
14         String fileName = "data1.bin";
15         File aFile = new File(fileName);
16         if (aFile.exists()) {
17             try {
18                 FileInputStream aFileInputStream = new FileInputStream(aFile);
19                 int i = aFileInputStream.read();
20                 while (i != 5) { // while not EOF
21                     System.out.println(i);
22                     i = aFileInputStream.read();
23                 }
24                 aFileInputStream.close();
25             } catch (IOException e) {
26                 // ...
27             }
28         }
29     }
30 }

```



```

1 package oop6_dat6_ss;
2
3 import java.io.Serializable;
4
5 /**
6  * Serialisierung: Klasse Matrix
7  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
8  */
9 public class Matrix implements Serializable
10 {
11
12     private int[][] m;
13     private transient int noOfAccess;
14
15     public Matrix(int row, int col)
16     {
17         m = new int[row][col];
18         noOfAccess = 0;
19     }
20
21     public void setElt(int r, int c, int value)
22     {
23         if ((r >= 0) && (r < m.length) && (c >= 0) && (c < m[0].length)) {
24             m[r][c] = value;
25             noOfAccess ++;
26         }
27     }
28
29     public void print()
30     {
31         for (int i = 0; i < m.length; i++) {
32             for (int j = 0; j < m[0].length; j++) {
33                 System.out.print(m[i][j] + " ");
34             }
35             System.out.println();
36         }
37         System.out.println("NoOfAccess: " + noOfAccess);
38     }
39 }

```

```

1 package oop6_dat6_ss;
2
3 import java.io.*;
4
5 /**
6  * Serialisierung: Objekte schreiben und lesen
7  * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
8  */
9 public class DemoSerialization
10 {
11
12     public static void main(String[] args)
13     {
14         try {
15             // Matrix definieren
16             Matrix m1 = new Matrix(3, 4);
17             m1.setElt(0, 0, 1); // row, col, value
18             m1.setElt(1, 1, 2);
19             m1.setElt(2, 2, 3);
20             m1.print();
21
22             Matrix m2 = new Matrix(2, 2);
23             m2.setElt(0, 1, 5);
24             m2.setElt(1, 0, 7);
25             m2.print();
26
27             // matrix in file schreiben - serialization
28             FileOutputStream aFileOutputStream = new FileOutputStream("matrix.dat");
29             ObjectOutputStream aObjectOutputStream = new
ObjectOutputStream(aFileOutputStream);
30             aObjectOutputStream.writeObject(m1);
31             aObjectOutputStream.writeObject(m2);
32             aFileOutputStream.close();
33
34             // matrix aus file lesen - (de)serialization
35             FileInputStream aFileInputStream = new FileInputStream("matrix.dat");
36             ObjectInputStream aObjectInputStream = new ObjectInputStream(aFileInputStream);
37             Object o = aObjectInputStream.readObject();
38             Object o2 = aObjectInputStream.readObject();
39             aFileInputStream.close();
40
41             // eingelesens Object ist vom Typ Object -> Cast auf Matrix
42             Matrix m3 = (Matrix) o;
43             Matrix m4 = (Matrix) o2;
44             m3.print();
45             m4.print();
46         } catch (Exception e) {
47             System.out.println("Exception: " + e.getMessage());
48         }
49     }
50 }

```