```java
package oop6_dat6_ss;

import java.io.*;
import java.text.*;

/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class WriteTextFile1
{

    public static void main(String[] args)
    {
        String fileName = "data1.txt";
        File aFile = new File(fileName);
        if ( ! aFile.exists()) {
            try {
                FileWriter aFileWriter = new FileWriter(aFile);
                BufferedWriter aBufferedWriter = new BufferedWriter(aFileWriter);
                PrintWriter aPrintWriter = new PrintWriter(aBufferedWriter);
                aPrintWriter.println("Dies ist die 1. Zeile.");
                aPrintWriter.println("Dies ist die 2. Zeile.");
                aPrintWriter.println("911");
                aPrintWriter.println("12.34");
                aPrintWriter.println("000.77");
                aPrintWriter.flush(); // Puffer proaktiv "spülen"
                aFileWriter.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java

package oop6_dat6_ss;

import java.io.*;
/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class WriteBinaryFile3 {
    public static void main(String[] args)
    {
        String fileName = "data3.bin";
        File aFile = new File(fileName);
        if ( ! aFile.exists()) {
            try {
                FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
                // DataOutputStream unterstützt elementare Datentypen
                DataOutputStream aDataOutputStream = new
DataOutputStream(aFileOutputStream);
                // 1. Anzahl intWerte, 2. intWerte...
                aDataOutputStream.writeInt(2);
                aDataOutputStream.writeInt(5);
                aDataOutputStream.writeInt(9);

                // 1. Anzahl floatWerte, 2. floatWerte ...
                aDataOutputStream.writeInt(3);
                aDataOutputStream.writeFloat(1.23f);
                aDataOutputStream.writeFloat(1.111f);
                aDataOutputStream.writeFloat(123.123f);

                // 1. Anzahl CharWerte, 2. CharWerte ...
                aDataOutputStream.writeInt(4);
                aDataOutputStream.writeChar('a');
                aDataOutputStream.writeChar('b');
                aDataOutputStream.writeChar('c');
                aDataOutputStream.writeChar('d');

                aFileOutputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java
package oop6_dat6_ss;

import java.io.*;

/**
 * Binäre Dateien: elementare Werte schreiben
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class WriteBinaryFile2
{

    public static void main(String[] args)
    {
        String fileName = "data2.bin";
        File aFile = new File(fileName);
        if ( ! aFile.exists()) {
            try {
                FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
                // DataOutputStream unterstützt elementare Datentypen
                DataOutputStream aDataOutputStream = new
DataOutputStream(aFileOutputStream);
                // 1. Anzahl intWerte, 2. intWerte...
                aDataOutputStream.writeInt(3);
                aDataOutputStream.writeInt(2);
                aDataOutputStream.writeInt(3);
                aDataOutputStream.writeInt(5);
                // 1. Anzahl doubleWerte, 2. doubleWerte ...
                aDataOutputStream.writeInt(2);
                aDataOutputStream.writeDouble(1.23456789);
                aDataOutputStream.writeDouble(1.E3);
                aFileOutputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```
package oop6_dat6_ss;

import java.io.*;

/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class WriteBinaryFile1
{

    public static void main(String[] args)
    {
        String fileName = "data1.bin";
        File aFile = new File(fileName);
        if ( ! aFile.exists()) {
            try {
                FileOutputStream aFileOutputStream = new FileOutputStream(aFile);
                for (int i = 0; i <= 10; i ++) {
                    aFileOutputStream.write(i);
                }
                aFileOutputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java
package oop6_dat6_ss;

import java.io.*;
import java.util.StringTokenizer;

/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class ReadTextFile1
{

    public static void main(String[] args)
    {
        String fileName = "data1.txt";
        File aFile = new File(fileName);
        if (aFile.exists()) {
            try {
                FileReader aFileReader = new FileReader(aFile);
                BufferedReader aBufferedReader = new BufferedReader(aFileReader);
                System.out.println(aBufferedReader.readLine());
                System.out.println(aBufferedReader.readLine());
                System.out.println();
                String line = aBufferedReader.readLine();
                while (line != null) { // while not EOF
                    Double d = Double.valueOf(line);
                    System.out.println(d);
                    line = aBufferedReader.readLine();
                }
                aFileReader.close();
            } catch (IOException e) {
                System.out.println("Exception: " + e.getMessage());
                return;
            }
        }
    }
}
```

```java

package oop6_dat6_ss;
import java.io.*;
/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class ReadBinaryFile3 {

    public static void main(String[] args)
    {
        String fileName = "data3.bin";
        File aFile = new File(fileName);
        if (aFile.exists()) {
            try {
                FileInputStream aFileInputStream = new FileInputStream(aFile);
                DataInputStream aDataInputStream = new DataInputStream(aFileInputStream);
                // Einlesen der intWerte
                int numberOfInt = aDataInputStream.readInt();
                for (int i = 0; i < numberOfInt; i ++) {
                    System.out.println(aDataInputStream.readInt());
                }
                // Einlesen der floatWerte
                int numberOfFloat = aDataInputStream.readInt();
                for (int i = 0; i < numberOfFloat; i ++) {
                    System.out.println(aDataInputStream.readFloat());
                }
                // Einlesen der charWerte
                int numberOfChar = aDataInputStream.readInt();
                for (int i = 0; i < numberOfChar; i ++) {
                    System.out.println(aDataInputStream.readChar());
                }
                aFileInputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java
package oop6_dat6_ss;

import java.io.*;

/**
 * Binäre Dateien: elementare Werte lesen
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class ReadBinaryFile2
{

    public static void main(String[] args)
    {
        String fileName = "data2.bin";
        File aFile = new File(fileName);
        if (aFile.exists()) {
            try {
                FileInputStream aFileInputStream = new FileInputStream(aFile);
                DataInputStream aDataInputStream = new DataInputStream(aFileInputStream);
                // Einlesen der intWerte
                int numberOfInt = aDataInputStream.readInt();
                for (int i = 0; i < numberOfInt; i ++) {
                    System.out.println(aDataInputStream.readInt());
                }
                // Einlesen der doubleWerte
                int numberOfDouble = aDataInputStream.readInt();
                for (int i = 0; i < numberOfDouble; i ++) {
                    System.out.println(aDataInputStream.readDouble());
                }
                aFileInputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java
package oop6_dat6_ss;

import java.io.*;

/**
 *
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class ReadBinaryFile1
{

    public static void main(String[] args)
    {
        String fileName = "data1.bin";
        File aFile = new File(fileName);
        if (aFile.exists()) {
            try {
                FileInputStream aFileInputStream = new FileInputStream(aFile);
                int i = aFileInputStream.read();
                while (i != 5) { // while not EOF
                    System.out.println(i);
                    i = aFileInputStream.read();
                }
                aFileInputStream.close();
            } catch (IOException e) {
                // ...
            }
        }
    }
}
```

```java
package oop6_dat6_ss;

import java.io.Serializable;

/**
 * Serialisierung: Klasse Matrix
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class Matrix implements Serializable
{

    private int[][] m;
    private transient int noOfAccess;

    public Matrix(int row, int col)
    {
        m = new int[row][col];
        noOfAccess = 0;
    }

    public void setElt(int r, int c, int value)
    {
        if ((r >= 0) && (r < m.length) && (c >= 0) && (c < m[0].length)) {
            m[r][c] = value;
            noOfAccess ++;
        }
    }

    public void print()
    {
        for (int i = 0; i < m.length; i ++) {
            for (int j = 0; j < m[0].length; j ++) {
                System.out.print(m[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println("NoOfAcccess: " + noOfAccess);
    }
}
```

```java
package oop6_dat6_ss;

import java.io.*;

/**
 * Serialisierung: Objekte schreiben und lesen
 * @author Felix Rohrer <felix.rohrer@stud.hslu.ch>
 */
public class DemoSerialization
{

    public static void main(String[] args)
    {
        try {
            // Matrix definieren
            Matrix m1 = new Matrix(3, 4);
            m1.setElt(0, 0, 1); // row, col, value
            m1.setElt(1, 1, 2);
            m1.setElt(2, 2, 3);
            m1.print();

            Matrix m2 = new Matrix(2, 2);
            m2.setElt(0, 1, 5);
            m2.setElt(1, 0, 7);
            m2.print();

            // matrix in file schreiben - serialization
            FileOutputStream aFileOutputStream = new FileOutputStream("matrix.dat");
            ObjectOutputStream aObjectOutputStream = new ObjectOutputStream(aFileOutputStream);
            aObjectOutputStream.writeObject(m1);
            aObjectOutputStream.writeObject(m2);
            aFileOutputStream.close();

            // matrix aus file lesen - (de)serialization
            FileInputStream aFileInputStream = new FileInputStream("matrix.dat");
            ObjectInputStream aObjectInputStream = new ObjectInputStream(aFileInputStream);
            Object o = aObjectInputStream.readObject();
            Object o2 = aObjectInputStream.readObject();
            aFileInputStream.close();

            // eingelesens Object ist vom Typ Object -> Cast auf Matrix
            Matrix m3 = (Matrix) o;
            Matrix m4 = (Matrix) o2;
            m3.print();
            m4.print();
        } catch (Exception e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```