

Hochschule -

Fakultät IV – Technische Informatik

Modul: Programmieren 1

Professor: -

Entwicklungsarbeit

von

Sebastian Schramm Matrikel-Nr. -

5. Dezember 2020

Inhaltsverzeichnis

1	Kapitel 1	3
1.1	Aufgabenstellung	3
1.2	Anforderungsdefinition	3
1.3	Entwurf	3
1.4	Quellcode	3
1.4.1	Main.java	3
1.5	Testdokumentation	3
1.6	Benutzungshinweise	3
1.7	Anwendungsbeispiel	4
2	Kapitel 3	4
2.1	Teilaufgabe 1	4
2.1.1	Aufgabenstellung	4
2.1.2	Anforderungsdefinition	4
2.1.3	Entwurf	4
2.1.4	Quelltext	4
2.1.4.1	Typkonvertierungen.java	4
2.1.5	Testdokumentation	7
2.1.6	Benutzungshinweise	7
2.1.7	Anwendungsbeispiel	7
2.2	Teilaufgabe 2	9
2.2.1	Aufgabenstellung	9
2.2.2	Anforderungsdefinition	9
2.2.3	Entwurf	9
2.2.4	Quelltext	9
2.2.4.1	Wertebereiche.java	9
2.2.5	Testdokumentation	10
2.2.6	Benutzungshinweise	10
2.2.7	Anwendungsbeispiel	10
3	Kapitel 4	10
3.1	Teilaufgabe 1	10
3.1.1	Aufgabenstellung	10
3.1.2	Anforderungsdefinition	10
3.1.3	Entwurf	10
3.1.4	Quellcode	10
3.1.4.1	Referenzen.java	10
3.1.4.2	Punkt.java	11
3.1.5	Testdokumentation	11
3.1.6	Benutzungshinweise	11
3.1.7	Anwendungsbeispiel	11
3.2	Teilaufgabe 2	12
3.2.1	Aufgabenstellung	12
3.2.2	Anforderungsdefinition	12
3.2.3	Entwurf	12
3.2.4	Quellcode	12
3.2.4.1	Matrizen.java	12
3.2.5	Testdokumentation	14
3.2.6	Benutzungshinweise	14
3.2.7	Anwendungsbeispiel	14

1 Kapitel 1

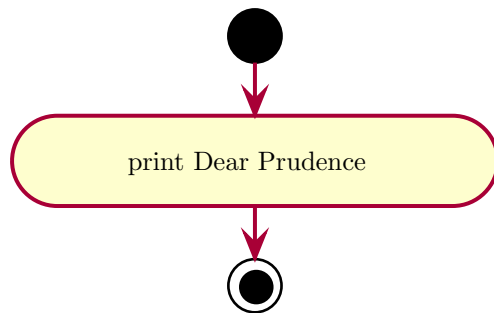
1.1 Aufgabenstellung

Wir sollen ein Programm schreiben welches den Text "Dear Prudence" in der Konsole ausgibt. Um uns mit Java vertraut zu machen, sollten wir das erste Programm in der Kommandozeile schreiben. Anschließend mit Javac Kompilieren und mit Java ausführen. Danach öffnen wir unsere IDE, erstellen ein neues Projekt und schreib das selbe Programm diesmal in der IDE.

1.2 Anforderungsdefinition

1. Das Programm soll "Dear Prudence" auf der Konsole ausgeben.

1.3 Entwurf



1.4 Quellcode

1.4.1 Main.java

```
1 package chapter_01;
2
3 /**
4  * Klasse mit der Main-Methode
5  * @author sebastian
6  *
7  */
8 public class Main {
9
10     /**
11      * Die Main Methode
12      * Gibt "Dear Prudence" aus
13      * @param args
14      */
15     public static void main(String[] args) {
16         System.out.println("Dear Prudence");
17     }
18 }
```

1.5 Testdokumentation

?

1.6 Benutzungshinweise

Navigieren Sie in der Kommandozeile zum dem Ordner, wo sich die Java Datei befindet. Danach führen sie "javac Main.java" auf. Jetzt können Sie das Programm mit "java main" starten. In der Konsole sollte nun "Dear Prudence" angezeigt werden.

1.7 Anwendungsbeispiel

Nach dem Aufruf von java Main, sollten wir folgendes sehen:

```
1 [sebastian@laptop bin]$ java Main
2 Dear Prudence
3 [sebastian@laptop bin]$
```

2 Kapitel 3

2.1 Teilaufgabe 1

2.1.1 Aufgabenstellung

In der ersten Teilaufgabe sollten wir uns mit der Typkonvertierung befassen. Welches alle primitive Datentypen erweiternd und einschränkend Konvertiert.

2.1.2 Anforderungsdefinition

1. Zu jedem Primitiven Datentypen eine erweiternde und einschränkende Konvertierung durchführen.

2.1.3 Entwurf

2.1.4 Quelltext

2.1.4.1 Typkonvertierungen.java

```
1 package chapter_03;
2
3 /**
4  * Klasse mit der Main-Methode
5  * und der einzelnen Typkonvertierungen
6  * @author Sebastian
7  */
8 public class Typkonvertierungen {
9
10     public static void main(String[] args) {
11         /*
12          * Rund die einzelnen Methoden auf, mit entsprechenden Werten
13          */
14         convertByte((byte) -128);
15         convertShort((short) 34);
16         convertInt(98987);
17         convertLong(987987987);
18
19         convertChar('a');
20
21         convertFloat(15.0f);
22         convertDouble(1.7976931348623157E308);
23     }
24
25     /**
26      * Eine erweiternde Konvertierung von Byte zu Double
27      * @param _byte
28      */
29     private static void convertByte(byte _byte) {
30         short newShort = _byte;
31         int newInt = _byte;
32         long newLong = _byte;
33         float newFloat = _byte;
34         double newDouble = _byte;
35
36         System.out.println("-----");
```

```

37     System.out.println("Byte erweiternd");
38     System.out.println("Byte   " + _byte);
39     System.out.println("Short  " + newShort);
40     System.out.println("Int    " + newInt);
41     System.out.println("Long   " + newLong);
42     System.out.println("Float  " + newFloat);
43     System.out.println("Bouble  " + newDouble);
44     System.out.println("\nChar   " + (char) newInt); //Char wird hier separat
        ausgegeben
45     System.out.println("-----");
46 }
47
48 /**
49  * Eine einschraenkende Konvertierung von Short zu Byte
50  * Eine erweiternde Konvertierung von Short zu Double
51  * @param _short
52  */
53 private static void convertShort(short _short) {
54     byte newByte = (byte) _short;
55     int newInt = _short;
56     long newLong = _short;
57     float newFloat = _short;
58     double newDouble = _short;
59
60     System.out.println("Short einschraenkend");
61     System.out.println("Short   " + _short);
62     System.out.println("Byte    " + newByte);
63
64     System.out.println("Short erweiternd");
65     System.out.println("Short   " + _short);
66     System.out.println("Int     " + newInt);
67     System.out.println("Long    " + newLong);
68     System.out.println("Float   " + newFloat);
69     System.out.println("Bouble  " + newDouble);
70     System.out.println("\nChar    " + (char) newInt); //Char wird hier separat
        ausgegeben
71     System.out.println("-----");
72 }
73
74 /**
75  * Eine einschraenkende Konvertierung von Int zu Byte
76  * Eine erweiternde Konvertierung von Int zu Double
77  * @param _int
78  */
79 private static void convertInt(int _int) {
80     short newShort = (short) _int;
81     byte newByte = (byte) _int ;
82
83     long newLong = _int;
84     float newFloat = _int;
85     double newDouble = _int;
86
87     System.out.println("Int einschraenkend");
88     System.out.println("Int      " + _int);
89     System.out.println("Short    " + newShort);
90     System.out.println("Byte     " + newByte);
91
92     System.out.println("Int erweiternd");
93     System.out.println("Int      " + _int);
94     System.out.println("Long     " + newLong);
95     System.out.println("Float    " + newFloat);
96     System.out.println("Bouble   " + newDouble);
97     System.out.println("\nChar     " + (char) _int); //Char wird hier separat

```

```

    ausgegeben
98     System.out.println("-----");
99 }
100
101 /**
102  * Eine einschaenkende Konvertierung von Long zu Byte
103  * Eine erweiternde Konvertierung von Long zu Double
104  * @param _long
105  */
106 private static void convertLong(long _long) {
107     int newInt = (int) _long;
108     short newShort = (short) _long;
109     byte newByte = (byte) _long;
110
111     float newFloat = _long;
112     double newDouble = _long;
113
114     System.out.println("Long einschaenkend");
115     System.out.println("Long    " + _long);
116     System.out.println("Int      " + newInt);
117     System.out.println("Short   " + newShort);
118     System.out.println("Byte    " + newByte);
119
120     System.out.println("Long erweiternd");
121     System.out.println("Long    " + _long);
122     System.out.println("Float    " + newFloat);
123     System.out.println("Bouble   " + newDouble);
124     System.out.println("\nChar    " + (char) newInt); //Char wird hier separat
    ausgegeben
125     System.out.println("-----");
126 }
127
128 /**
129  * Eine einschaenkende Konvertierung von Char zu Byte
130  * Eine erweiternde Konvertierung von Char zu Double
131  * @param _char
132  */
133 private static void convertChar(char _char) {
134     int newInt = _char;
135     short newShort = (short) _char;
136     byte newByte = (byte) _char;
137
138     long newLong = _char;
139     float newFloat = _char;
140     double newDouble = _char;
141
142     System.out.println("Char einschaenkend");
143     System.out.println("Char    " + _char);
144     System.out.println("Long    " + newLong);
145     System.out.println("Int      " + newInt);
146     System.out.println("Short   " + newShort);
147     System.out.println("Byte    " + newByte);
148
149     System.out.println("Char erweiternd");
150     System.out.println("Char    " + _char);
151     System.out.println("Long    " + newLong);
152     System.out.println("Float    " + newFloat);
153     System.out.println("Bouble   " + newDouble);
154     System.out.println("-----");
155 }
156
157 /**
158  * Eine einschaenkende Konvertierung von FLoat zu Byte

```

```

159  * Eine erweiternde Konvertierung von Float zu Double
160  * @param _float
161  */
162  private static void convertFloat(float _float) {
163      long newLong = (long) _float;
164      int newInt = (int) _float;
165      short newShort = (short) _float;
166      byte newByte = (byte) _float;
167
168      double newDouble = _float;
169
170      System.out.println("Float einschraenkend");
171      System.out.println("Float " + _float);
172      System.out.println("Long " + newLong);
173      System.out.println("Int " + newInt);
174      System.out.println("Short " + newShort);
175      System.out.println("Byte " + newByte);
176
177      System.out.println("Float erweiternd");
178      System.out.println("Float " + _float);
179      System.out.println("Bouble " + newDouble);
180      System.out.println("\nChar " + (char) newInt); //Char wird hier separat
181      ausgegeben
182      System.out.println("-----");
183  }
184  /**
185   * Eine einschraenkende Konvertierung von Double zu Byte
186   * @param _double
187   */
188   private static void convertDouble(double _double) {
189       float newFloat = (float) _double;
190       long newLong = (long) _double;
191       int newInt = (int) _double;
192       short newShort = (short) _double;
193       byte newByte = (byte) _double;
194
195       System.out.println("Double einschraenkend");
196       System.out.println("Bouble " + _double);
197       System.out.println("Float " + newFloat);
198       System.out.println("Long " + newLong);
199       System.out.println("Int " + newInt);
200       System.out.println("Short " + newShort);
201       System.out.println("Byte " + newByte);
202       System.out.println("\nChar " + (char) newInt); //Char wird hier separat
203       ausgegeben
204       System.out.println("-----");
205   }
206 }

```

2.1.5 Testdokumentation

2.1.6 Benutzungshinweise

Keine Besonderen Benutzungshinweise. Man navigiere zu dem Ordner von sich die Compilierte Datei mit dem Namen "Typkonvertierungen.class" befindet und führt anschließend `java Typkonvertierungen` aus.

2.1.7 Anwendungsbeispiel

Nach dem man das Programm gestartet hat, sollte folgende Ausgabe erscheinen:

```

1 | [sebastian@laptop bin]$ java Typkonvertierungen

```

```

2  -----
3  Byte erweiternd
4  Byte   -128
5  Short  -128
6  Int    -128
7  Long   -128
8  FLoat  -128.0
9  Bouble -128.0
10
11 Char
12 -----
13 Short einschraenkend
14 Short   34
15 Byte    34
16 Short erweiternd
17 Short   34
18 Int     34
19 Long    34
20 FLoat   34.0
21 Bouble  34.0
22
23 Char    "
24 -----
25 Int einschraenkend
26 Int     98987
27 Short  -32085
28 Byte   -85
29 Int erweiternd
30 Int     98987
31 Long    98987
32 FLoat   98987.0
33 Bouble  98987.0
34
35 Char
36 -----
37 Long einschraenkend
38 Long   987987987
39 Int    987987987
40 Short -32749
41 Byte   19
42 Long erweiternd
43 Long   987987987
44 FLoat  9.8798797E8
45 Bouble 9.87987987E8
46
47 Char
48 -----
49 Char einschraenkend
50 Char   a
51 Long   97
52 Int    97
53 Short  97
54 Byte   97
55 Char erweiternd
56 Char   a
57 Long   97
58 FLoat  97.0
59 Bouble 97.0
60 -----
61 Float einschraenkend
62 FLoat  15.0
63 Long   15
64 Int    15

```



```

65 Short 15
66 Byte 15
67 Float erweiternd
68 FLoat 15.0
69 Bouble 15.0
70
71 Char
72 -----
73 Double einschraenkend
74 Bouble 1.7976931348623157E308
75 FLoat Infinity
76 Long 9223372036854775807
77 Int 2147483647
78 Short -1
79 Byte -1
80
81 Char
82 -----
83 [sebastian@laptop bin]$

```

2.2 Teilaufgabe 2

2.2.1 Aufgabenstellung

In dieser Teilaufgabe sollen wir ein Programm schreiben welle die Wertebereiche der primitieven Datentypen ausgibt.

2.2.2 Anforderungsdefinition

1. Zu jedem primitieven Datentypen den Max und Min-Wert ausgeben.

2.2.3 Entwurf

2.2.4 Quelltext

2.2.4.1 Wertebereiche.java

```

1 package chapter_03;
2
3 /**
4  * Klasse mit der Main-Methode
5  * und gibt die Wertebereiche der primitieven Datentypen aus
6  * @author Sebastian
7  */
8 public class Wertebereiche {
9
10     public static void main(String[] args) {
11         //Min und Max Value von Byte
12         System.out.println("Byte min " + Byte.MIN_VALUE + " | Byte max " + Byte.
13             MAX_VALUE);
14         //Min und Max Value von Short
15         System.out.println("Short min " + Short.MIN_VALUE + " | Short max " + Short.
16             MAX_VALUE);
17         //Min und Max Value von Integer
18         System.out.println("Integer min " + Integer.MIN_VALUE + " | Integer max " +
19             Integer.MAX_VALUE);
20         //Min und Max Value von Long
21         System.out.println("Long min " + Long.MIN_VALUE + " | Byte Long " + Long.
22             MAX_VALUE);
23
24         //Min und Max Value von Char
25         System.out.println("Char min " + Character.MIN_VALUE + " | Char max " +
26             Character.MAX_VALUE);
27     }
28 }

```

```

22 //System.out.println("\u0000 | \uffff");
23
24 //Min und Max Value von Float
25 System.out.println("Float min " + Float.MIN_VALUE + " | Float max " + Float.
    MAX_VALUE);
26 //Min und Max Value von Double
27 System.out.println("Double min " + Double.MIN_VALUE + " | Double max " + Double
    .MAX_VALUE);
28 }
29
30 }

```

2.2.5 Testdokumentation

2.2.6 Benutzungshinweise

Keine Besonderen Benutzungshinweise. Man navigiere zu dem Ordner von sich die Compilierte Datei mit dem Namen "Wertebereiche.class" befindet und führt anschließend java Wertebereiche aus.

2.2.7 Anwendungsbeispiel

Nach dem man das Programm gestartet hat, sollte folgende Ausgabe erscheinen:

```

31 [sebastian@laptop bin]$ java Wertebereiche
32 Byte min -128 | Byte max 127
33 Short min -32768 | Short max 32767
34 Integer min -2147483648 | Integer max 2147483647
35 Long min -9223372036854775808 | Byte Long 9223372036854775807
36 Char min | Char max
37 Float min 1.4E-45 | Float max 3.4028235E38
38 Double min 4.9E-324 | Double max 1.7976931348623157E308
39 [sebastian@laptop bin]$

```

3 Kapitel 4

3.1 Teilaufgabe 1

3.1.1 Aufgabenstellung

Wir sollen ein Programm schreiben welches Prüft ob zwei Referenzen gleich sind.

3.1.2 Anforderungsdefinition

1. Prüfe ob zwei Referenzen gleich sind.

3.1.3 Entwurf

3.1.4 Quellcode

3.1.4.1 Referenzen.java

```

1 package chapter_04;
2
3 /**
4  * Classe mit der Main-Methode
5  * und prüft ob Zwei Referenzen gleich sind
6  * @author Sebastian
7  *
8  */
9 public class Referenzen {
10
11     public static void main(String[] args) {

```

```

12  /*
13   * Es werden zwei identische Objekte erzeugt
14   * mit den selben Werten.
15   * Zuletzt wird noch ein drittes erzeugt mit einer
16   * Referenz auf das erste
17   */
18  Punkt p1 = new Punkt(10, 20);
19  Punkt p2 = new Punkt(10, 20);
20  Punkt p3 = p1;
21
22  //Hier wird gerüft ob p1 und p2 die selbe Adresse hat.
23  if (p1 == p2)
24      System.out.println("Ist gleich");
25  else
26      System.out.println("Ist ungleich");
27
28  //Hier wird geprüft ob der Inhalt der selbe ist
29  if (p1.equals(p2))
30      System.out.println("Ist gleich");
31  else
32      System.out.println("Ist ungleich");
33
34  //Hier wird geprüft ob p3 und p1 gleich sind
35  if (p3 == p1)
36      System.out.println("Ist gleich");
37  else
38      System.out.println("Ist ungleich");
39  }
40
41  }

```

3.1.4.2 Punkt.java

```

1  package chapter_04;
2
3  /**
4   * Punkt Classe
5   * Hier werden nur Zwei Punkte gespeichert
6   * @author Sebastian
7   *
8   */
9  @SuppressWarnings("unused")
10 public class Punkt {
11     private int x = 0;
12     private int y = 0;
13
14     public Punkt(int x, int y) {
15         this.x = x;
16         this.y = y;
17     }
18 }

```

3.1.5 Testdokumentation

3.1.6 Benutzungshinweise

Navigieren Sie in der Kommandozeile zum dem Ordner, wo sich die Java Datei befindet. Danach führen sie "javac Referenzen.java" auf. Jetzt können Sie das Programm mit "java Referenzen" starten.

3.1.7 Anwendungsbeispiel

Nach dem Aufruf von java Referenzen, sollten wir nun folgendes sehen:

```

1 [sebastian@laptop bin]$ java Referenzen
2 Ist ungleich
3 Ist ungleich
4 Ist gleich
5 [sebastian@laptop bin]$

```

3.2 Teilaufgabe 2

3.2.1 Aufgabenstellung

Wir sollen ein Programm schreiben welches 2 nxn Matrizen miteinander Addieren und Multiplizieren kann.

3.2.2 Anforderungsdefinition

1. Addiere zwei nxn Matrizen.
2. Multipliziere zwei nxn Matrizen.

3.2.3 Entwurf

3.2.4 Quellcode

3.2.4.1 Matrizen.java

```

1 package chapter_04;
2
3 import java.util.Scanner;
4 /**
5  * Classe mit der Main-Methode
6  * Addiert und Multipliziert Matrizen
7  * @author Sebastian
8  *
9  */
10 public class Matrizen {
11
12     public static void main(String[] args) {
13         int matrixA[][];
14         int matrixB[][];
15
16         //Hier können sie die GrösSe definieren, z.B. 2,3 oder 5
17         System.out.println("Dieses Programm berechnet eine zufällig erstellte nxn
18             Matrix");
19         System.out.print("Geben sie n an: ");
20         Scanner sc = new Scanner(System.in);
21         int n = sc.nextInt();
22
23         /*
24          * Inizialisierungsmethode wird mit dem Wert n aufgerufen.
25          * Anschliessend wird diese Matrix erzeugt und mit
26          * zufällig generierten Zahlen befüllt.
27          */
28         matrixA = initialize(n);
29         matrixB = initialize(n);
30
31         /*
32          * Zuerst werden die Beiden Matrizen A und B jeweils ausgegeben
33          */
34         System.out.println("Matrix A:");
35         printMatrix(matrixA);
36         System.out.println("Matrix B:");
37         printMatrix(matrixB);
38     }
39
40     /*
41      * Methode zur Initialisierung einer Matrix
42      * @param n Die Grösse der Matrix
43      * @return Eine nxn Matrix mit zufällig generierten Zahlen
44      */
45     private static int[][] initialize(int n) {
46         int[][] matrix = new int[n][n];
47         for (int i = 0; i < n; i++) {
48             for (int j = 0; j < n; j++) {
49                 matrix[i][j] = (int) (Math.random() * 100);
50             }
51         }
52         return matrix;
53     }
54
55     /*
56      * Methode zum Ausgeben einer Matrix
57      * @param matrix Die Matrix die ausgegeben werden soll
58      */
59     private static void printMatrix(int[][] matrix) {
60         for (int i = 0; i < matrix.length; i++) {
61             for (int j = 0; j < matrix[i].length; j++) {
62                 System.out.print(matrix[i][j] + " ");
63             }
64             System.out.println();
65         }
66     }
67 }

```

```

38     * AnschlieSSend werden die Matrizen hier Addiert
39     */
40     System.out.println("Addition von A und B:");
41     printMatrix(addition(matrixA, matrixB));
42     /*
43     * Und hier Multipliziert
44     */
45     System.out.println("Multiplikation von A und B:");
46     printMatrix(multiplikation(matrixA, matrixB));
47
48     sc.close();
49 }
50
51 /**
52  * Initialisierung des Arrays
53  * @param n
54  * @return matrix
55  */
56 private static int[][] initialize(int n) {
57     int matrix[][] = new int[n][n];
58     /*
59     * Bei der Initialisierung wird einmal durch das gesamt Array durch itteriert.
60     * Dabei werden dann mit Math.random() zufällige Zahlen rein geschrieben.
61     */
62     for (int i = 0; i < matrix.length; ++i)
63         for (int l = 0; l < matrix[i].length; ++l)
64             matrix[i][l] = (int) (Math.random() * 100);
65
66     return matrix;
67 }
68
69 /**
70  * Addition der beiden Matrizen A und B
71  * @param matrixA
72  * @param matrixB
73  * @return
74  */
75 private static int[][] addition(int matrixA[][], int matrixB[][]) {
76     int matrixAd[][] = new int[matrixA.length][matrixA[0].length]; //Es wird ein
77     //neues Temporäres Array angelegt
78
79     for (int i = 0; i < matrixA.length; ++i) {
80         for (int n = 0; n < matrixA[i].length; ++n) {
81             matrixAd[i][n] = matrixA[i][n] + matrixB[i][n];
82         }
83     }
84
85     return matrixAd;
86 }
87
88 /**
89  * Multiplikation der beiden Matrizen A und B
90  * @param matrixA
91  * @param matrixB
92  * @return
93  */
94 private static int[][] multiplikation(int matrixA[][], int matrixB[][]) {
95     int matrixMult[][] = new int[matrixB.length][matrixB[0].length];
96
97     for (int HmatrixB = 0; HmatrixB < matrixB.length; ++HmatrixB)
98         for (int WmatrixB = 0; WmatrixB < matrixB[HmatrixB].length; ++WmatrixB)
99             for (int WmatrixA = 0; WmatrixA < matrixB.length; ++WmatrixA)
100                 matrixMult[HmatrixB][WmatrixB] += matrixA[HmatrixB][WmatrixA] * matrixB[

```

```

100         WmatrixA][WmatrixB];
101     return matrixMult;
102 }
103
104 /**
105  * Hier wird die Matrix ausgegeben
106  * @param matrix
107  */
108 private static void printMatrix(int matrix[][] ) {
109     for (int y[]: matrix) {
110         for (int x: y)
111             System.out.print(x + "\t");
112         System.out.println();
113     }
114     System.out.println();
115 }
116
117 }

```

3.2.5 Testdokumentation

3.2.6 Benutzungshinweise

Navigieren Sie in der Kommandozeile zum dem Ordner, wo sich die Java Datei befindet. Danach führen sie "javac Matrizen.java" auf. Jetzt können Sie das Programm mit "java Matrizen" starten. Nach dem das Programm gestartet ist, können Sie die größe der Matrix angeben.

3.2.7 Anwendungsbeispiel

Nach dem Aufruf von java Matrizen, sollten wir nun folgendes sehen:

```

1  [sebastian@laptop bin]$ java Matrizen
2  Dieses Programm berechnet eine zufällig erstellte nxn Matrix
3  Geben sie n an: 5
4  Matrix A:
5  22  54  23  36  35
6  49  98  67  82  23
7  78  62  36  60  2
8  54  2  74  48  95
9  58  2  86  3  75
10
11 Matrix B:
12  84  7  63  44  84
13  69  55  76  83  97
14  60  98  66  41  4
15  27  6  71  96  24
16  40  8  21  66  38
17
18 Addition von A und B:
19  106 61  86  80  119
20  118 153 143 165 120
21  138 160 102 101 6
22  81  8 145 144 119
23  98  10 107 69  113
24
25 Multiplikation von A und B:
26  9326 5874 10299 12159 9372
27  18032 12975 21262 22427 16732
28  14690 7860 16304 15946 14226
29  14210 8788 13841 16454 9788
30  13251 9562 11270 11482 8332
31 [sebastian@laptop bin]$

```