

Hochschule -

Fakultät IV – Technische Informatik

Modul: Programmieren 1

Professor: -

Entwicklungsarbeit

von

Sebastian Schramm Matrikel-Nr. -

5. Dezember 2020

Inhaltsverzeichnis

1	Kapitel 3	3
1.1	Teilaufgabe 1	3
1.1.1	Aufgabenstellung	3
1.1.2	Anforderungsdefinition	3
1.1.3	Entwurf	3
1.1.4	Quelltext	3
1.1.4.1	Typkonvertierungen.java	3
1.1.5	Testdokumentation	6
1.1.6	Benutzungshinweise	6
1.1.7	Anwendungsbeispiel	6
1.2	Teilaufgabe 2	8
1.2.1	Aufgabenstellung	8
1.2.2	Anforderungsdefinition	8
1.2.3	Entwurf	8
1.2.4	Quelltext	8
1.2.4.1	Wertebereiche.java	8
1.2.5	Testdokumentation	9
1.2.6	Benutzungshinweise	9
1.2.7	Anwendungsbeispiel	9

1 Kapitel 3

1.1 Teilaufgabe 1

1.1.1 Aufgabenstellung

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

1.1.2 Anforderungsdefinition

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

1.1.3 Entwurf

1.1.4 Quelltext

1.1.4.1 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 einschränkende 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 einschränkend");
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("Double " + newDouble);
70     System.out.println("\nChar " + (char) newInt); //Char wird hier separat
        ausgegeben
71     System.out.println("-----");
72 }
73
74 /**
75  * Eine einschränkende 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 einschränkend");
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("Double " + newDouble);
97     System.out.println("\nChar " + (char) _int); //Char wird hier separat
        ausgegeben
98     System.out.println("-----");
99 }
100
101 /**
102  * Eine einschränkende 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 einschraenkend");
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
125     //ausgegeben
126     System.out.println("-----");
127 }
128 /**
129  * Eine einschraenkende 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 einschraenkend");
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 einschraenkende 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
        ausgegeben
181     System.out.println("-----");
182 }
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
        ausgegeben
203     System.out.println("-----");
204 }
205
206 }

```

1.1.5 Testdokumentation

1.1.6 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.

1.1.7 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]$

```

1.2 Teilaufgabe 2

1.2.1 Aufgabenstellung

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

1.2.2 Anforderungsdefinition

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

1.2.3 Entwurf

1.2.4 Quelltext

1.2.4.1 Wertebereiche.java

```

84 package chapter_03;
85
86 /**
87  * Klasse mit der Main-Methode
88  * und gibt die Wertebereiche der primitieven Datentypen aus
89  * @author Sebastian
90  */
91 public class Wertebereiche {
92
93     public static void main(String[] args) {
94         //Min und Max Value von Byte
95         System.out.println("Byte min " + Byte.MIN_VALUE + " | Byte max " + Byte.
96             MAX_VALUE);
97         //Min und Max Value von Short
98         System.out.println("Short min " + Short.MIN_VALUE + " | Short max " + Short.
99             MAX_VALUE);
100         //Min und Max Value von Integer
101         System.out.println("Integer min " + Integer.MIN_VALUE + " | Integer max " +
102             Integer.MAX_VALUE);
103         //Min und Max Value von Long
104         System.out.println("Long min " + Long.MIN_VALUE + " | Byte Long " + Long.
105             MAX_VALUE);
106
107         //Min und Max Value von Char
108         System.out.println("Char min " + Character.MIN_VALUE + " | Char max " +
109             Character.MAX_VALUE);
110         //System.out.println("\u0000 | \uffff");
111
112         //Min und Max Value von Float
113         System.out.println("Float min " + Float.MIN_VALUE + " | Float max " + Float.
114             MAX_VALUE);
115         //Min und Max Value von Double
116         System.out.println("Double min " + Double.MIN_VALUE + " | Double max " + Double.
117             MAX_VALUE);
118     }

```



```
112 |  
113 }
```

1.2.5 Testdokumentation

1.2.6 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.

1.2.7 Anwendungsbeispiel

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

```
114 [sebastian@laptop bin]$ java Wertebereiche  
115 Byte min -128 | Byte max 127  
116 Short min -32768 | Short max 32767  
117 Integer min -2147483648 | Integer max 2147483647  
118 Long min -9223372036854775808 | Byte Long 9223372036854775807  
119 Char min | Char max  
120 Float min 1.4E-45 | Float max 3.4028235E38  
121 Double min 4.9E-324 | Double max 1.7976931348623157E308  
122 [sebastian@laptop bin]$
```