

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

Aufgabe T8.1: Syntaxdiagramm (2 + 4 = 6 Punkte)

Sie haben auf den Folien 324 ff. die Regeln und Vorgaben einer erlaubten Syntax kennengelernt. Nun sei folgender Code gegeben:

```
1  int a = 12
2  int b = 0;
3
4  while (a < 24) {
5      switch (a) {
6          case : 0
7              b = b + a;
8          case 1:
9              if a = 12 {{return;}}
10         default:
11             a = a - 1;
12     }
13 }
```

(a) Untersuchen Sie den Code auf Syntaxfehler und begründen Sie ihre Antwort.

```
1  int a = 12                                //Semikolon fehlt
2  int b = 0;
3
4  while (a < 24) {
5      switch (a) {
6          case : 0                            //case 0: wäre richtig
7              b = b + a;                      //Semantischer Schwachsinn wenn a = 0 ist ist b+a=b... b
              ↳ würde also reichen.
8          case 1:                             //In allen cases fehlt break; außer das ist so gewünscht
9              if a = 12 {{return;}}           //Klammern um den Booleschen Ausdruck fehlen und a = 12
              ↳ ist eine Variablenzuweisung, kein boolescher Ausdruck und wenn a == 1 ist kann a
              ↳ nicht == 12 sein. Wozu das return und was soll es wohin returnen.
10         default:
11             a = a - 1;
12     }
13 }
```

Wenn wir mal ignorieren das alles drumherum fehlt, also Klasse, Methoden-Signatur etc. dann würde immer a immerweiter runtergezählt und die Abbruchbedingung $a > 24$ nie erreicht. Da Int in Java jedoch die minimale Größe von -2147483648 hat, wird a bis zu dieser Zahl berechnet und das Programm ordnungsgemäß terminiert. So wird ein scheinbar valides Ergebnis erzeugt.

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

Der ausführbare Code würde wie folgt aussehen:

```
1  public class Main {
2      public static void main(String[] args) {
3
4          int a = 12;
5          int b = 0;
6
7          while (a < 24) {
8              switch (a) {
9                  case 0:
10                     b = b + a;
11                 case 1:
12                     if (a == 12) {{return;}}
13                 default:
14                     a = a - 1;
15             }
16
17         }
18     }
19 }
```

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

- (b) Gegeben sei eine Sprache, die eine Syntax definiert. In diesem Fall beschreibt sie das Erstellen eines Übungsblattes. Geben Sie ein Syntaxdiagramm an, das folgende Regeln (R1-R8) definiert:

R1 Ein Aufgabenblatt besitzt zu Beginn einen Titel.

R2 Ein Titel ist ein Freitext.

R3 Ein Aufgabenblatt besteht danach aus beliebig vielen, mindestens jedoch einer Aufgabe.

R4 Eine Aufgabe ist durch das Zeichen **A** gekennzeichnet, gefolgt von der maximal zu erreichenden Punktzahl, gefolgt vom Zeichen **P**. Hierauf folgen beliebig viele, jedoch mindestens eine, Teilaufgaben.

R5 Jede Teilaufgabe wird durch das Zeichen Punkt '•' eingeleitet, auf das ein Aufgabentext folgt.

R6 Ein Aufgabentext ist ein Freitext.

R7 Ein Freitext ist eine beliebige Kombination der Zeichen 'a', 'b', 'c', 'd', 'e' und '_' (Leerzeichen).

R8 Die maximal zu erreichende Punktzahl ist eine ein- bzw. zweistellige Zahl zwischen 1 und 99.

Ermitteln Sie zunächst anhand der Regeln, welche Terminal- und Nichtterminalsymbole existieren. Geben Sie ein Syntaxdiagramm an, das diese Symbole entsprechend der genannten Regeln zusammenführt. Kennezeichnen Sie in Ihrem Diagramm, welche Regel dargestellt werden soll (mehrere Regeln in einem Diagramm sind möglich). Ein Syntaxdiagramm kann aus mehreren Teildiagrammen bestehen.

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

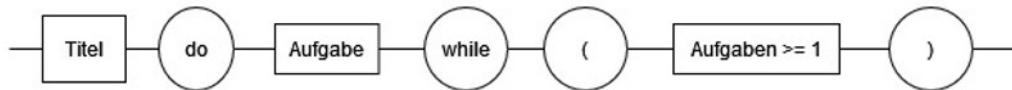
Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

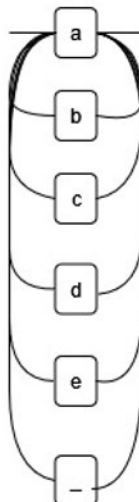
Aufgabenblatt:



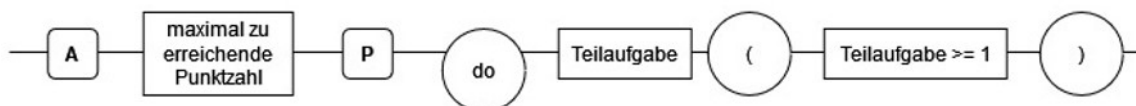
Titel:



Freitext:



Aufgabe:



Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

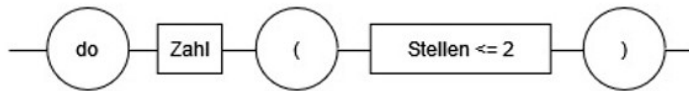
Tutor: Tim Völker

Ali Kurt 528961

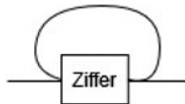
Thomas Kujawa 463620

Felix Hoff 374689

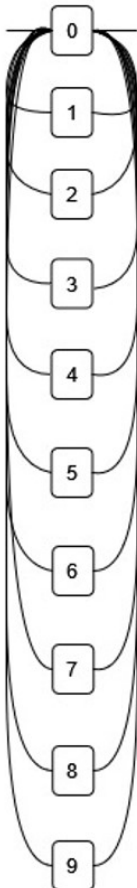
maximal zu erreichende Punktzahl:



Zahl:



Ziffer:



Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

Teilaufgabe:



Aufgabentext:



Aufgabe T8.2: BNF (6 + 2 = 8 Punkte)

Nachfolgend sei eine Grammatik in Form der BNF gegeben:

```
<Wort> ::= <VKV> | <KVK> | <Wort><KVK>
<VKV> ::= <Vok><Kons><Vok>
<KVK> ::= <KVK><VV> | <Kons><Vok><Kons>
<VV>  ::= <Vok><Vok>
<Vok> ::= A | E | I | O | U | Ä | Ö | Ü
<Kons> ::= B | C | D | F | G | H | J | K | L | M | N | P | Q |
          R | S | T | V | W | X | Y | Z
```

- (a) Geben Sie für die folgenden Worte an, ob diese durch die gegebene BNF abgebildet werden und begründen Sie ihre Antwort:

i) LILIE

Das Wort wird durch die Grammatik abgebildet, da LIL ein $\langle\text{KVK}\rangle$ ist und IE ein $\langle\text{VV}\rangle$. Daraus lässt sich das zulässige $\langle\text{KVK}\rangle ::= \langle\text{KVK}\rangle\langle\text{VV}\rangle$ bilden. Als $\langle\text{Wort}\rangle$ ist ein $\langle\text{KVK}\rangle$ zulässig.

ii) ZUCKER

Das Wort wird durch die Grammatik abgebildet, da ZUC ein $\langle\text{KVK}\rangle$ ist. Ein $\langle\text{KVK}\rangle$ ist ein zulässiges $\langle\text{Wort}\rangle$. KER ist ebenfalls ein $\langle\text{KVK}\rangle$ die Konkatenation von ZUC und KER bildet ein Zulässiges $\langle\text{Wort}\rangle$ wegen $\langle\text{Wort}\rangle ::= \langle\text{Wort}\rangle \langle\text{KVK}\rangle$.

iii) AUBERGINE

Das Wort wird nicht durch die Grammatik abgebildet, da AU ein $\langle\text{VV}\rangle$ ist und BER ein $\langle\text{KVK}\rangle$. Es gibt keine Regel laut der auf ein $\langle\text{VV}\rangle$ ein $\langle\text{KVK}\rangle$ folgen darf.

iv) BITTERMANDEL

Das Wort wird durch die Grammatik abgebildet, da BIT, TER, MAN, DEL alles $\langle\text{KVK}\rangle$'s sind und nach $\langle\text{Wort}\rangle ::= \langle\text{Wort}\rangle \langle\text{KVK}\rangle$ ergeben konkatenierte $\langle\text{KVK}\rangle$'s ein zulässiges $\langle\text{Wort}\rangle$.

v) BASILIKUM

Das Wort wird nicht durch die Grammatik abgebildet, da BAS ein $\langle\text{KVK}\rangle$ ist und ILI ein $\langle\text{VKV}\rangle$. In der Grammatik existiert keine gültige Regel in der ein $\langle\text{VKV}\rangle$ auf ein $\langle\text{KVK}\rangle$ folgen darf.

vi) BLAUBEERE

Das Wort wird nicht durch die Grammatik abgebildet, da B ein $\langle\text{Kons}\rangle$ und L ein $\langle\text{Kons}\rangle$ ist. In der Grammatik existiert keine gültige Regel in der ein $\langle\text{Kons}\rangle$ auf ein $\langle\text{Kons}\rangle$ folgen darf.

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

-
- (b) Nennen Sie zwei deutsche Wörter, jeweils bestehend aus 5 oder mehr Buchstaben, die durch die gegebene BNF gebildet werden können. Dabei müssen die beiden Wörter in gängigen Wörterbüchern zu finden sein. Zusammengesetzte Worte sind dabei auch erlaubt.

▸ KACHEL

▸ TACKER

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

Aufgabe T8.3: Code-Verständnis (1 Punkte)

Ihnen wird die Datei *HelloJava.java* bereitgestellt. Der Code beinhaltet ein vollständiges Programm, das allerdings einige Funktionen nutzt, die Sie bisher noch nicht kennengelernt haben.

Kopieren Sie den Code oder nutzen Sie die gegebene Datei und führen Sie diese aus. (Sie können beispielsweise die in der Vorlesung vorgestellte Software Eclipse nutzen.) Beschreiben Sie in knappen Sätzen, was die Aufgabe des Programms ist und welche Eingaben wie verwertet werden.

Das Programm berechnet alle Primzahlen bis zum eingegebenen Wert n . Als Eingaben werden ganzzahlige, positive und negative Integer akzeptiert. Characters und Floatingpoints führen zum Programabsturz. Bei $n \leq 1$ werden keine Primzahlen zurückgegeben, da die kleinstmögliche Primzahl 2 ist und Primzahlen nur im positiven definiert sind. Das Ergebnis ist folglich korrekt und der Programm wird vollständig ausgeführt.

Aufgabe T8.4: While / Do While / For Schleife (2 + 2 + 2 = 6 Punkte)

In der Vorlesung haben Sie verschiedene Schleifen kennengelernt. Nun geben Sie ein Programm in Auftrag, das den Satz "Hallo Welt" n mal ausgibt und dafür eine Schleife verwendet. Dabei kann n Werte im Intervall $I = [0, \infty)$ annehmen. Ihnen wurden nun drei verschiedene Programme geschickt, die sich jeweils einer Art von Schleife bedienen. Da Sie sich noch nicht sicher sind, welche Art von Schleife Ihnen am besten gefällt, wollen Sie die Möglichkeit haben, zwischen den verschiedenen Formen zu wechseln. Beschreiben Sie daher für Schleifen mit n Iterationen,

Jede der drei Schleifen benötigt einen Parameter n , der der Methode übergeben wird. Der Parameter gibt an, wie oft die Schleife durchlaufen werden soll. Desweiteren ist es sinnvoll eine Laufvariable zu definieren, welche auf 0 initialisiert wird. Diese Variable ist zwar nicht für alle Schleifentypen erforderlich, erleichtert jedoch die Umwandlung von einen zum anderen Typ.

- (a) wie eine `while` in eine `do while` Schleife umgewandelt werden kann und umgekehrt,

Für ein `while`-Programm würde zu Beginn der Methode eine Variable `i` vom Typen `int` auf 0 initialisiert werden. Im Schleifenkopf steht die Bedingung, wie lange der Schleifenrumpf ausgeführt werden soll. In diesem Falle wäre `while (n>i)` eine sinnvolle Möglichkeit. Darauf folgt der Schleifenrumpf, in dem der Befehl `System.out.println("Hello World!");` zur Ausgabe des String "Hello World!" gefolgt von der Inkrementierung der Laufvariable `i` durch `i++`; folgt.

Um das oben beschriebene `while`-Programm in ein `do while`-Programm zu überführen, würde der Schleifenkopf mit einem `do` beginnen. Der Schleifenrumpf ist identisch mit dem des `while`-Programms. Auf den Schleifenrumpf folgt die Abbruchbedingung, welche identisch ist mit der Bedingung des `while`-Programms, also `while (n>i)`.

Der Unterschied zwischen einer `while` in einer `do while` Schleifen besteht darin, dass bei einer `while` Schleife der Schleifenrumpf solange ausgeführt wird, wie die Schleifenbedingung wahr ist. Bei einer `do while` Schleife wird der Schleifenrumpf mindestens einmal ausgeführt und danach solange, wie die Schleifenbedingung wahr ist.

Die Umkehr von einer `do while`-Schleife in eine `while`-Schleife erfolgt analog zu dem oben beschriebenen Weg, da die Umwandlungsschritte äquivalent sind. Dazu wird das `do` ersetzt durch die Schleifenbedingung `while (n>1)`; und das `while (n>1)`; am Ende der `do while`-Schleife entfernt.

Zur Veranschaulichung folgt der entsprechende Java-Code:

```
1 public static void HelloWorldWhile(int n){
2     int i = 0;
3     while (n > i) {
4         System.out.println("Hello World!");
5         i++;
6     }
7 }
8 public static void HelloWorldDoWhile(int n){
9     int i = 0;
10    do{
11        System.out.println("Hello World!");
12        i++;
13    }while (n > i);
14 }
```

(b) wie eine `while` in eine `for` Schleife umgewandelt werden kann und umgekehrt und

Das in Aufgabenteil a) beschriebene `while`-Programm, lässt sich wie folgt in ein `for`-Programm umwandeln. Die Laufvariable `int i = 0;` wird nun im Schleifenkopf der `for`-Schleife initialisiert, darauf folgt die Schleifenbedingung mit `i < n;` und dann die Inkrementierung von `i` mit `i++`. Bei der `for`-Schleife befindet sich im Schleifenrumpf nur der Befehl `System.out.println("Hello World!");`, da die Inkrementierung bereits im Kopf stattfindet.

Zur Umwandlung der `for`-Schleife in eine `while`-Schleife wird die Laufvariable `int i = 0;` aus dem Schleifenkopf entfernt und vor der Schleife initialisiert. Die Inkrementierung von `i` durch `i++;` wird aus dem Schleifenkopf entfernt und in den Schleifenrumpf eingefügt.

Zur Veranschaulichung folgt der entsprechende Java-Code:

```
1 public static void HelloWorldWhile(int n){
2     int i = 0;
3     while (n > i) {
4         System.out.println("Hello World!");
5         i++;
6     }
7 }
8 public static void HelloWorldFor(int n){
9     for(int i = 0; i < n; i++){
10        System.out.println("Hello World!");
11    }
12 }
```

- (c) wie eine `do while` in eine `for` Schleife umgewandelt werden kann und umgekehrt.

Zur Umwandlung des in Aufgabenteil a) beschriebenen `do while`-Programms in ein `for`-Programm wird das `do` ersetzt durch ein `for` gefolgt von einer runden Klammer. Die Initialisierung der Laufvariable `int i = 0;` wird vor der `do while`-Schleife entfernt und nach der runden Klammer eingefügt. Die Schleifenbedingung am Ende der `do while`-Schleife `while (n>i);` wird entfernt und nach der Laufvariable im Schleifenkopf als `i<n;` eingefügt. Die Inkrementierung `i++;` wird aus dem Schleifenrumpf entfernt und nach der Schleifenbedingung im Schleifenkopf der `for`-Schleife eingefügt. Der Schleifenkopf wird durch eine runde Klammer geschlossen. In geschweiften Klammern folgt der Schleifenrumpf mit `System.out.println("Hello World!");`

Die Umwandlung der `for`-Schleife in eine `do while`-Schleife erfolgt wieder analog. Der Schleifenkopf der `for`-Schleife, also `for(int i = 0; i < n; i++)` wird durch `do` ersetzt. Die Laufvariable `int i = 0;` wird vor der Schleife initialisiert. Im Schleifenrumpf wird die Inkrementierung `i++;` eingefügt. am Ende der Schleife wird die Schleifenbedingung `while(n>i)` eingefügt.

Zur Veranschaulichung folgt der entsprechende Java-Code:

```
1 public static void HelloWorldDoWhile(int n){
2     int i = 0;
3     do{
4         System.out.println("Hello World!");
5         i++;
6     }while (n > i);
7 }
8 public static void HelloWorldFor(int n){
9     for(int i = 0; i < n; i++){
10        System.out.println("Hello World!");
11    }
12 }
```

Bitte beachten Sie, dass Ihre Beschreibung allgemein gültig und für alle möglichen $n \in I$ gelten soll. Sie dürfen allerdings für Ihre Erklärungen das oben genannte Beispiel nutzen.

Aufgabe T8.5: Nikolaus 1 (3 Bonuspunkte)

Der Nikolaus ist auf der Suche nach Cartoons, die die Themen Informatik und Weihnachten verbinden. Reichen Sie einen solchen Cartoon (entweder selbst gezeichnet oder mit Quellenangabe!) ein und erklären Sie kurz, d.h. in maximal 3 Sätzen, warum dieser Cartoon einen Bezug zur Informatik hat und wie ein Bezug zu den bisher behandelten Themen in der Vorlesung Informatik 1 besteht. Wenn keine Erklärung angegeben wird, gibt es nur 1 Punkt und falls kein Bezug zur Vorlesung erklärt wird, gibt es nur 2 Punkte.



101010 binär ist 42 dezimal. Binärumrechnungen wurden in der Vorlesung behandelt und sind ein grundlegendes Element in der Informatik. 101010 war das Passwort zum Learnweb-Kurs. Warum das für Prof. Linsens Humor spricht muss ich

Aufgabe P8.6: Nikolaus 2 (4 Punkte +3 Bonuspunkte)

Für die diesjährige Nikolausaufgabe hat Ihr Übungsleiter einen Nikolaus gemalt. Leider gab es vorher auf dem Weihnachtsmarkt ein oder zwei Gläser Eierlikör zu viel, sodass ihm dabei ein Fehler unterlaufen ist. Gegeben ist das Programm `MakeNikolaus.java`, das das ebenfalls gegebene Bild `Nikolaus.jpg` einlädt und als multidimensionales Array zur Verfügung stellt. Im Array sind die RGB-Farbwerte der Pixel gespeichert und die Dimensionen des Arrays entsprechen jeweils der Höhe und der Breite des Bildes. Eine genaue Beschreibung finden Sie im Sourcecode. Außerdem wandelt das Programm das Array wieder in ein Bild um und speichert es als `NewNikolaus.jpg`.

- (a) Passen Sie die Variable `String path` an, sodass das Ihnen bereitgestellte Bild `Nikolaus.jpg` geladen wird und testen Sie, ob es unverändert als `NewNikolaus.jpg` im selben Ordner gespeichert wird. Fügen Sie nun im Quelltext an angegebener Stelle Ihren Code hinzu, der das multidimensionale Array durchläuft und somit den Wert jedes Pixels ausliest. Dabei handelt es sich um einen einzelnen Integer Wert. Nutzen Sie die bereitgestellte Funktion `isBlue`, um zu überprüfen, ob es sich um ein blaues Pixel handelt. Sollte dieser Fall eintreffen, ersetzen Sie den Wert im multidimensionalen Array mit einem roten Pixel. Nutzen Sie hierfür die bereitgestellte Funktion `makeColor(int r , int g , int b)` mit den Werten $r = 255, g = 0, b = 0$. Das fertige Ergebnis sollte etwa so aussehen:



- (b) (Bonus) Passen Sie den Code in a) so an, dass die Mütze mehrfarbig ist, z.B. mit einen Farbverlauf oder einem Muster. Dabei können Sie so kreativ sein, wie Sie möchten. Die Lösung für Aufgabenteil a) dürfen Sie dafür auskommentieren.



Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

```
1 import java.io.File;
2 import java.io.IOException;
3 import java.awt.image.BufferedImage;
4 import javax.imageio.ImageIO;
5
6 public class MakeNikolaus{
7
8     public static void main(String args[])throws IOException{
9
10         // //////////////////////////////////////
11         // Aufgabe a) Pfad zur Bilddatei auf IHREM Rechner anpassen.
12         // //////////////////////////////////////
13
14         // Pfad zur Bilddatei:
15         String path = "C:\\Users\\felix\\OneDrive\\Studium\\Uni Münster\\Informatik\\1.
16         ↳ Semester\\Informatik 1\\Übungszettel\\Übungszettel LaTeX\\Hausaufgaben\\Blatt
17         ↳ 8\\images";
18         String filename = path + "\\Nikolaus.jpg";
19
20
21         // //////////////////////////////////////
22         // Folgenden Code bitte VOR den weiteren Aufgaben stehen lassen
23         // //////////////////////////////////////
24
25         // Lade Bilddatei:
26         MyFile f = new MyFile(null);
27         BufferedImage img = loadImage(f, filename);
28         // Konvertiere in 2D Array
29         int[] [] imageArray = BufferedImageToArray(img);
30
31         // Lies Bildgröße aus:
32         int width = img.getWidth(); // 300px
33         int height = img.getHeight(); // 600px
34
35
36
37
38         // //////////////////////////////////////
39         //
40         // //////////////////////////////////////
41         // Aufgabe a)
42         // Das Bild liegt nun im 2D Array "imageArray" vor.
```

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

```
43      // Dabei gibt die erste Dimension die Pixelposition der Breite vor,
44      // die zweite Dimension die Pixelposition der Höhe.
45      // imageArray[0][0] speichert daher den Wert des ersten Pixels oben links.
46      // imageArray[1][0] speichert den Wert des Pixels rechts davon.
47      // imageArray[0][1] speichert den Wert unterhalb des ersten Pixels.
48      // Verallgemeinert gilt also: imageArray[y][x]
49      //
50      // Schreiben Sie in den folgenden Zeilen einen Code, der jedes Pixel besucht
51      // und daraufhin den Farbwert per zurückbekommt.
52      // Beispiel: int color = imageArray[0][0];
53      // Nutzen Sie dafür die Bilddimensionen width und height
54      //
55      // Überprüfen Sie den Farbwert jedes Pixels darauf,
56      // ob er blau ist. Dies geschieht durch die Funktion isBlue
57      // isBlue(int color) liefert also true zurück, wenn das Pixel blau ist.
58      // Wenn Sie ein blaues Pixel gefunden haben, ändern Sie die Farbe des Pixels in rot.
59      // Dies geschieht mit Hilfe der Funktion makeColor(int r, int g, int b).
60      // Ein Pixel kann also wie folgt verändert werden:
61      // imageArray[y][x] = makeColor(r,g,b)
62      // Dabei soll r = 255, g = 0 und b = 0 sein.
63      // //////////////////////////////////////
64
65      // Aufgabe 8.6a
66      // for(int cols = 0; cols < height; cols++){
67      //     for(int rows = 0; rows < width; rows++){
68      //         int color = imageArray[cols][rows];
69      //         if (isBlue(color)){
70      //             imageArray[cols][rows] = makeColor(255, 0, 0);
71      //         }
72      //     }
73      // }
74
75      // Aufgabe 8.6b
76
77      int green = 0;
78      int blue = 100;
79      int red = 200;
80
81      for(int cols = 0; cols < height; cols++){
82          for(int rows = 0; rows < width; rows++){
83              int color = imageArray[cols][rows];
84              if (isBlue(color)){
85                  green = (green >= 255) ? 0 : green;
86                  blue = (blue >= 255) ? 0 : blue;
87                  red = (red >= 255) ? 0 : red;
```

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

```
88         imageData[cols][rows] = makeColor(red, green, blue);
89         green+= 30;
90         blue+= 30;
91         red+= 30;
92     }
93 }
94 }
95
96
97 // //////////////////////////////////////
98 // Den folgenden Code bitte immer NACH Ihrem Code stehen lassen:
99 // //////////////////////////////////////
100
101 // Verwandle Array zu Bilddatei
102 ArrayToBufferedImage(imageArray, img);
103 // Schreibe Bild an den selben Ort wie das Eingangsbild
104 writeImage(f, img);
105 }
106
107
108
109
110 // //////////////////////////////////////
111 //
112 //
113 // Bitte Funktionen unterhalb dieses Kommentars NICHT verändern
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 // Oh, Sie haben doch entschieden, bis hier hin weiterzulesen?
132 // Na dann hoffe ich, Sie lernen noch was schönes dabei :)
```

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

```
133      //
134      // //////////////////////////////////////
135
136
137      // Funktion die den Farbwert in die Bestandteile r,g,b aufteilt
138      // und überprüft, ob ein sehr hoher Blauanteil und sehr niedriger Rotanteil vorliegt:
139      private static boolean isBlue(int color) {
140          int r = (color>>16) & 0xff;
141          int g = (color>>8) & 0xff;
142          int b = color & 0xff;
143
144          if (b > 200 && r < 50) {
145              return true;
146          } else {
147              return false;
148          }
149      }
150
151
152      // Funktion, die r,g,b Intensitätswerte nimmt und in einen Pixelwert umwandelt.
153      public static int makeColor(int r, int g, int b) {
154          return (255<<24) | (r<<16) | (g<<8) | b;
155      }
156
157      // Konvertiert ein BufferedImage Objekt in ein 2D Array
158      public static int[] [] BufferedImageToArray(BufferedImage img) {
159          int width = img.getWidth();
160          int height = img.getHeight();
161
162          int[] [] twoDArray = new int[height][width];
163
164          for(int x = 0; x < width; x++) {
165              for(int y = 0; y < height; y++) {
166                  twoDArray[y][x] = img.getRGB(x, y);
167              }
168          }
169
170          return twoDArray;
171      }
172
173      // Konvertiert ein 2D Array in ein BufferedImage Objekt
174      public static void ArrayToBufferedImage(int[] [] twoDArray, BufferedImage img) {
175
176          int width = img.getWidth();
177          int height = img.getHeight();
```

Informatik I - Gruppe 1 - Übungsblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

```
178
179     for(int i = 0; i < width; i++) {
180         for(int j = 0; j < height; j++) {
181             img.setRGB(i, j, twoDarray[j][i]);
182         }
183     }
184 }
185
186 // Lädt eine Bilddatei am angegebenen Pfad
187 public static BufferedImage loadImage(MyFile f, String filename) {
188     BufferedImage img = null;
189     System.out.println("Lade Bild:");
190     // Try to read the image
191     try{
192         f.myFile = new File(filename);
193         System.out.println("Prüfe Pfad: " + f.myFile.getPath());
194         img = ImageIO.read(f.myFile);
195         System.out.println("Bild erfolgreich geladen");
196     } catch(IOException e){
197         System.out.println(e);
198     }
199     return img;
200 }
201
202 // Speichert das bearbeitete Bild an der selben Stelle wie das originale Bild
203 public static void writeImage(MyFile f, BufferedImage img) {
204     System.out.println("Speichere Bild:");
205     try{
206         File g = new File(f.myFile.getParent()+"\\NewNikolaus.jpg");
207         ImageIO.write(img, "jpg", g);
208     }catch(IOException e){
209         System.out.println(e);
210     }
211     System.out.println("Bild erfolgreich gespeichert:");
212     System.out.println(f.myFile.getParent()+"\\NewNikolaus.jpg");
213 }
214 }
215
216 // Wrapperklasse, um File Objekt weiterzugeben
217 class MyFile {
218     public File myFile;
219
220     MyFile(File myFile) {
221         this.myFile = myFile;
222     }
}
```

Informatik I - Gruppe 1 - Übungblatt 8

Ausgabe: 05.12.2022

Abgabe: 12.12.2022

Tutor: Tim Völker

Ali Kurt 528961

Thomas Kujawa 463620

Felix Hoff 374689

223 }
