Musterlösung Übung 11

Wiederholungen

1. Aus welchen Komponenten besteht Ihre Stereoanlage[1]? Unterteilen Sie diese in die Kategorien Eingabe, Ausgabe und Verarbeitung von Information.

   Eingabe: *Radio/Tuner, CD/DVD-Laufwerk, Kassettendeck (Abspielteil), ggf. Plattenspieler, EIngabeschnittstellen (Audio-IN, USB, MIDI).*

   Verarbeitung: *Mixer, Equalizer, Verstärker (Lautstärkeregelung), ggf. Entzerrer oder Effektfilter.*

   Ausgabe: *Lautsprecher, Ausgabeschnittstellen (Audio-OUT, USB, MIDI, Digitalausgang), Kassettendeck (Aufnahmeteil), ggf. CD/DVD-Brenner.*

2. In welchem Jahr wurde vermutlich der erste funktionsfähige Computer gebaut, und in welcher Programmiersprache wurden die Programme vom Erfinder entworfen?

   Siehe Vorlesungsfolie 9, Konrad Zuse baute den vermutlich ersten wirklich funktionsfähigen Computer mit elektromechanischer Steuerung im Jahre 1936. Zur Programmierung dieser Maschine mit einer „universellen" formalen Algorithmenbeschreibung entwarf er gleichzeitig das „Plankalkül", mit dem sich Programme für diesen Computer konstruieren ließen.

3. Ist „Windows" eigentlich ein Betriebssystem im Sinne der Wikipedia-Definition? Ist „Linux" ein Betriebssytem?

   Als reines Betriebssystem (= Schnittstelle zwischen Hardware und Anwendersoftware) kann man Windows wohl nicht bezeichnen, da in seiner Architektur teilweise Anwenderprogramme direkt auf die Hardware zugreifen, und andererseits nicht zwischen der graphischen Oberfläche mit Applikationen (=Anwendersoftware) und reiner Systemsteuerung (=Betriebssystem) unterschieden wird.

   Linux hingegen ist definitionsgemäß tatsächlich als reines Betriebssystem konzipiert, enthält also im Kern (s.a. „Kernel") lediglich die zur Steuerung des Betriebes und insbesondere zum Zugriff auf Hardware notwendigen Mechanismen. Die Anwendersoftware, die auf den Linux-Kernel aufsetzt, entstammt hingegen größtenteils dem GNU-Projekt.

   Um eine Linux-basierte Computerinstallation mit „Windows" vergleichen zu können, also eine komplette Suite aus Betriebssystem und Anwendersoftware, sollte daher korrekterweise der Begriff „GNU/Linux" statt nur „Linux" verwendet werden.

4. Nenne Sie die drei wichtigsten Eigenschaften von Information in der Informatik.

   Syntax = formale Korrektheit, Semantik = Bedeutung/Interpretation, Pragmatik = Nützlichkeit (schwer zu definieren).

[1] auch dies ist ein datenverarbeitendes elektronisches Gerät

5. Bestimmen Sie den Informationsgehalt der Menge aller natürlichen Zahlen von 1...49.

   Um aus einer Menge von in diesem Fall genau 49 Zahlen eine Zahl durch „Ja/Nein"-Fragen zu erraten, sind im schlechtesten Fall 6 Fragen erforderlich (durch Ausschließen von möglichst jeweils der Hälfte der möglichen verbleibenden Zahlen bei jeder Frage), also ist der Informationsgehalt gleich 6. Formal sind zur Darstellung von 49 verschiedenen Zahlen 6 Bit erforderlich (die nächsthöhere Dualzahl zu 49 ist nämlich $2^6 = 64$).

6. Stellen Sie eine Interpretationsfunktion auf, die die Zahlen 1...9 auf einer Fernbedienung als Fernsehsender (mit beliebigen Namen) darstellt.

$$
\begin{aligned}
W &= \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \\
O &= \{\text{„}\textit{Sender1}\text{“}, \text{„}\textit{Sender2}\text{“}, \text{„}\textit{Sender3}\text{“}, \text{„}\textit{Sender4}\text{“}, \text{„}\textit{Sender5}\text{“}, \text{„}\textit{Sender6}\text{“}, \\
&\quad \text{„}\textit{Sender7}\text{“}, \text{„}\textit{Sender8}\text{“}, \text{„}\textit{Sender9}\text{“}\} \\
I &: W \to O \\
I(1) &= \text{„}\textit{Sender1}\text{“} \\
I(2) &= \text{„}\textit{Sender2}\text{“} \\
I(3) &= \text{„}\textit{Sender3}\text{“} \\
I(4) &= \text{„}\textit{Sender4}\text{“} \\
I(5) &= \text{„}\textit{Sender5}\text{“} \\
I(6) &= \text{„}\textit{Sender6}\text{“} \\
I(7) &= \text{„}\textit{Sender7}\text{“} \\
I(8) &= \text{„}\textit{Sender8}\text{“} \\
I(9) &= \text{„}\textit{Sender9}\text{“}
\end{aligned}
$$

7. Bilden Sie die Potenzmenge P aus einer Menge M mit $M = \{a, b, c\}$.

   *Die **Potenzmenge** enthält sämtliche Teilmengen der Ursprungsmenge, auch die leere Menge, als Elemente:*

$$P = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$$

8. Geben Sie einen Werte- und Definitionsbereich an, in dem die Wurzelfunktion $f(x) = \sqrt{x}$ bijektiv ist.

   Es gibt unendlich viele Lösungen, man muss hier lediglich einen Bereich finden, in dem für alle $f(x)$ im Wertebereich genau ein x im Definitionsbereich existiert. Dies trifft beispielsweise für

$$
\begin{aligned}
D &= \{x \mid x \geq 0 \quad \wedge \quad x \in \mathbb{R}\} \\
W &= \{y \mid y = f(x) \quad \wedge \quad y \in \mathbb{R}\}
\end{aligned}
$$

   zu.

9. Wenn A = „wahr“ und B = „falsch“ ist, welchen Wert hat dann $(A \wedge B) \vee ((A \vee B) \wedge (A \vee B))$?

$$\begin{aligned}(\text{„}wahr\text{“} \wedge \text{„}falsch\text{“}) \vee ((\text{„}wahr\text{“} \vee \text{„}falsch\text{“}) \wedge (\text{„}wahr\text{“} \vee \text{„}falsch\text{“})) &= \\ \text{„}falsch\text{“} \vee (\text{„}wahr\text{“} \wedge \text{„}wahr\text{“}) &= \\ \text{„}falsch\text{“} \vee \text{„}wahr\text{“} &= \textbf{„wahr“}\end{aligned}$$

10. Schreiben Sie ein C-Programm, mit dem Sie Ihre Behauptung aus der vorigen Aufgabe überprüfen können.

    Siehe **`Logik.c`**.

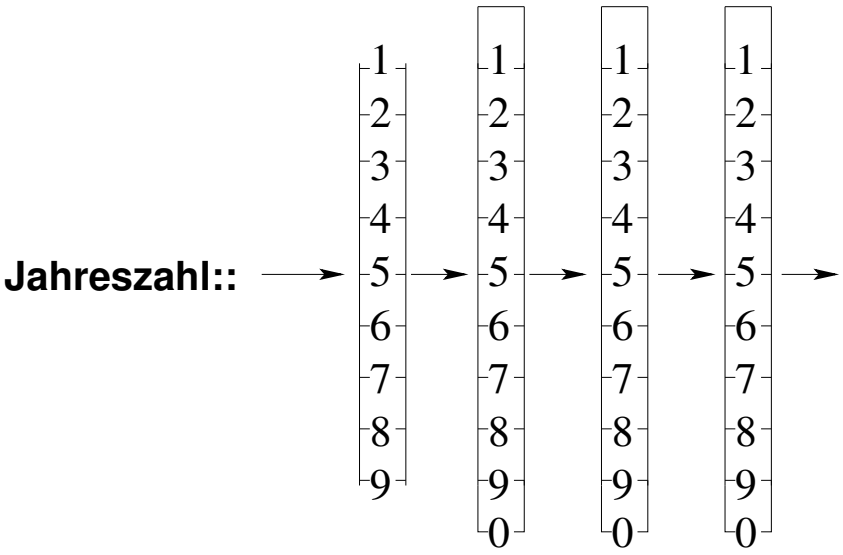
11. Stellen Sie die Dezimalzahl 123 in binärer, oktaler und in hexadezimaler Schreibweise dar.

$$\begin{aligned}123 &= \boxed{1} \cdot 64 + \boxed{1} \cdot 32 + \boxed{1} \cdot 16 + \boxed{1} \cdot 8 + \boxed{0} \cdot 4 + \boxed{1} \cdot 2 + \boxed{1} \\ &= 1111011_2 \\ 123 &= \boxed{1} \cdot (8 \cdot 8) + \boxed{7} \cdot 8 + \boxed{3} \\ &= 173_8 \\ 123 &= \boxed{7} \cdot 16 + \boxed{11} \\ &= 7B_{16}\end{aligned}$$

12. Schreiben Sie ein C-Programm, mit dem Sie zumindest die Hexadezimal- und Oktaldarstellung von 123 ausgeben können. (Hinweis: **`%o`** = Oktalschreibweise, **`%x`** = Hexadezimalschreibweise in der Formatangabe von **`printf()`**.)

    Siehe **`einszweidrei.c`**.

13. Erstellen Sie eine Grammatik für maximal 4-stellige, positive Jahreszahlen, wobei keine führenden Nullen auftreten dürfen. Verwenden Sie die Darstellung als Syntaxdiagramm sowie die formale Darstellung als Chomsky Typ 2 Grammatik.



Formale Grammatik:
Man fängt mit der Terminalmenge T (Menge erlaubter Zeichen) an, und findet beim Aufstellen der Produktionsregeln P „automatisch“ eine Anzahl von Ersetzungsvariablen, die nach und nach in die Nichterminalmenge N (Menge der Variablen) eingesetzt werden. Eine davon ist das Startwort, mit dem die Ersetzung und Bildung der Wörter beginnt.

Beispiel:

$$
\begin{aligned}
G_{jahreszahl} &= \{N, T, P, S\} \\
N &= \{Z, W, A\} \\
T &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\
P &= \{Z \rightarrow 1|2|3|4|5|6|7|8|9,\ \ W \rightarrow Z|0|\varepsilon,\ \ A \rightarrow ZWWW\} \\
S &= A
\end{aligned}
$$

14. Entwerfen Sie einen Algorithmus zur Bestimmung des Median-Wertes von zwei Messwerten in C. Schreiben Sie hierzu eine Fuktion **`median`**, die ein **`double`** zurückliefert und zwei **`double`**-Werte als Übergabeparameter erwartet.

    Mit nur 2 Eingabewerten ist der Median-Wert gleich dem Mittelwert aus zwei Zahlen!

```
double median(double a, double b)
{

 /* Unbedingt richtig klammern, da sonst
    Punkt- vor Strichrechnung gilt! */

 return (a + b) / 2.0;
}
```

15. Welcher Komplexitätsklasse gehört der folgende Algorithmus an? Was berechnet er?

```
int mal4(int a)
 {
  int i;
  for(i=0; i<2; i++) a += a;
  return a;
 }
```

Obwohl er eine ***for()****-Schleife enthält, gehört der Algorithmus der Komplexitätsklasse* $O(1)$ *an, da die Schleife nicht von irgendeiner Eingabevariablen abhängt, sondern immer genau und konstant 2 Durchläufe hat. Man könnte die Schleife also genausogut durch zwei Einzelanweisungen ersetzen.*

Man kann den kompletten Anweisungsblock der Funktion durch ***return a * 4;*** *ersetzen, was aber an der Komplexitätsklasse nichts ändert.*

16. Implementieren Sie in C eine Funktion **modulo** mit zwei **int**-Übergabeparametern **x** und **y**, die als **int**-Rückgabewert den Teilungsrest von **x/y** zurückliefert, OHNE dass der Modulo-Operator **%** verwendet wird!

```
int modulo(int x, int y)
{
 /* Berechnet werden soll eigentlich x % y
    -> Teilungsrest von x/y */

 int z = x / y;  /* Ganzzahlige Teilung ohne Nachkommastellen! */
 return x - (z * y); /* Liefert Teilungsrest */
}
```

17. Schreiben Sie eine rekursive Funktion (wahlweise in C oder JAVA), die von einer gegebenen Zahl x die Summe der Reziprokwerte angefangen mit $\frac{1}{2} + \frac{1}{3} + ... + \frac{1}{x}$ berechnet.

```
double rezi_sum(double x)
{
 if(x < 2.0) return 0.0;
 return 1.0/x + rezi_sum(x - 1.0);
}
```

Kleine Zusatzaufgabe: Ist der obige Algorithmus nun ein C- oder ein JAVA-Programmfragment?

18. Setzen Sie die Verbund-Definitionen aus Übung 9 als JAVA-Klassen um, und zwar mit **ItalienischesGericht** als Basisklasse, und den Klassen **Pasta**, **Lasagne** etc. als abgeleiteten (**extends**) Klassen. Schreiben Sie eine Klasse **u11**(**.java**) zum Testen, erzeugen Sie in **main()** ein **Lasagne**-Objekt und setzen Sie dessen Preis auf 4,50 €.

Siehe ***u11.java****.*

19. Implementieren Sie für jede Klasse aus der vorigen Aufgabe einen Konstruktor, der beim Erzeugen eines Objektes einen Text (z.B. **`"Hole Lasagne"`**) ausgibt.

    Siehe ebenfalls ***`u11.java`***.

20. Versuchen Sie, mit Hilfe der in den **`javakurs.pdf`**-Folien gezeigten Grafik-Funktionen ein Programm zu schreiben, das ein „lachendes Gesicht“ zeichnet. Sie können dies je nach beabsichtigtem Aufwand als alleine lauffähiges Java-Programm, oder als Applet zur Einbindung in Webseiten implementieren. Eine Vorlage für eine Webseite mit Java-Applet ist auf der nächsten Seite angegeben, der zur Verfügung stehende Grafikbereich ist in diesem Beispiel auf 400 x 400 Pixel beschränkt.

Datei ***LachendesGesicht.html*** *(Inklusive Quelltext für LachendesGesicht.java):*

```
<HEAD><TITLE>Lachendes Gesicht in Java</TITLE></HEAD>
<BODY>

Beispiel-Applet (Quelltext) von <B>LachendesGesicht.java</B>:
<PRE>
import java.awt.Graphics;
public class LachendesGesicht extends java.applet.Applet {
 public void paint(Graphics g) {
   // Zeichne einen Kreis
   g.drawOval(1, 1, 398, 398);
   // Auge links
   g.fillOval(200-100, 180-40, 40, 20);
   // Auge rechts
   g.fillOval(200+ 60, 180-40, 40, 20);
   // Nase
   g.drawLine(200,160,200,260);
   // Mund
   g.drawArc(80, 220, 240, 90, 360-90-60, 120);
 }
}
</PRE>

Hier ist der Bereich, in dem das Applet läuft:<BR>

<CENTER>
<APPLET CODE="LachendesGesicht.class" CODEBASE="."
        width=400 height=400> </APPLET>
</CENTER>

</BODY>
```