
Übungsblatt 2 zur Vorlesung Prog 1 / GdP

Übung 1

Rechnerarchitekturen und Turing-Vollständigkeit

- 1) Welche Vorteile haben festprogrammierte Rechner im Vergleich zu speicherprogrammierten Rechnern? Welche Beispiele von festprogrammierten Rechnern kennen Sie aus Ihrem täglichen Leben?
- 2) Wir haben gelernt, dass Turing-vollständige Programmiersprachen i.d.R. gegeneinander ausgetauscht werden können. Warum ist die Wahl der Programmiersprache dennoch ein so intensiv debattiertes Thema unter Entwicklern und ein so zentrales Entscheidungsthema unter Softwarearchitekten?
- 3) [Hoffentlich] zum Schmunzeln und Reflektieren:
<https://geek-and-poke.com/geekandpoke/2013/11/16/spoilsports>

Übung 2

Variablen und Werte

Benennen Sie in jedem Schritt die **Werte** der Variablen `umfang` und `flaeche`. Bitte immer erst im Kopf überlegen und dann in Python die Annahme prüfen:

<pre>pi=3.14159 radius=3 umfang = 2*pi*radius flaeche = pi*radius**2</pre>	<pre>umfang = flaeche =</pre>
<pre>pi=3.14</pre>	<pre>umfang = flaeche =</pre>
<pre>radius = 7</pre>	<pre>umfang = flaeche =</pre>
<pre>umfang = 2*pi*radius</pre>	<pre>umfang = flaeche =</pre>
<pre>radius = 5 umfang = 2*pi*radius</pre>	<pre>umfang = flaeche =</pre>
<pre>flaeche = pi*radius**2</pre>	<pre>umfang = flaeche =</pre>
<pre>umfang = flaeche flaeche = umfang</pre>	<pre>umfang = flaeche =</pre>
<pre>umfang = 2*pi*radius flaeche = pi*radius**2</pre>	<pre>umfang = flaeche =</pre>

Übung 3

Schwimmbadtarife – einfache Abfragen

Ein Schwimmbad hat (vereinfacht) folgende Tarife:

- Kleinkinder (bis einschließlich 2 Jahre): 0,00€
- Kinder (ab 3 bis einschl. 15 Jahre): 3,40€
- Jugendliche (ab 16 Jahre): 4,00€
- Erwachsene (ab 18 Jahre): 5,00€
- Rentner (ab 67 Jahre): 3,00€

Schreiben Sie ein simples Programm, welches bei Angabe des Alters des Schwimmbadbesuchers den korrekten Preis ausgibt.

Zur Überprüfung Ihres Programms folgende Testausgaben:

Test-Case 1:

```
Bitte geben Sie das Alter der Person ein: 1
Zu zahlender Eintrittspreis: 0,00€
```

Test-Case 2:

```
Bitte geben Sie das Alter der Person ein: 9
Zu zahlender Eintrittspreis: 3,40€
```

Übung 4

Zutrittskontrolle in der Schwimmbadrutsche

Das Schwimmbad hat eine sehr schnelle Rutsche. Aus Sicherheitsgründen müssen folgende Regeln eingehalten werden:

- Kleinkinder dürfen die Rutsche nicht nutzen
- Kinder dürfen die Rutsche nur in Begleitung eines Erwachsenen oder eines Rentners nutzen
- Falls Kinder jedoch mindestens 150cm groß sind, ist eine Begleitung nicht mehr erforderlich
- Jugendliche dürfen die Rutsche immer benutzen

Schreiben Sie ein Programm, welches bei Angabe

- des **Alters** der Person und
- der **Größe** der Person (aber nur wenn erforderlich!)

ausgibt, ob

- es **zulässig** ist die Rutsche zu nutzen, oder
- es **nicht zulässig** ist die Rutsche zu benutzen, oder
- es **nur zulässig** ist, die Rutsche in **Begleitung** eines Erwachsenen oder Rentners zu benutzen.

Zur Überprüfung Ihres Programms folgende Testausgaben:

Test-Case 1:

```
Bitte geben Sie das Alter der Person ein: 2
```

Rutschen ist nicht zulässig!

Test-Case 2:

Bitte geben Sie das Alter der Person ein: 9
Bitte geben Sie die Größe der Person ein: 155
Rutschen ist zulässig!

Test-Case 3:

Bitte geben Sie das Alter der Person ein: 9
Bitte geben Sie die Größe der Person ein: 145
Rutschen ist nur in Begleitung durch einen Erwachsenen oder
Rentner zulässig!

Test-Case 4:

Bitte geben Sie das Alter der Person ein: 99
Rutschen ist zulässig!

Test-Case 5:

Bitte geben Sie das Alter der Person ein: 18
Rutschen ist zulässig!

Übung 5

AND, OR, NOT

1) Was ist das Ergebnis der folgenden Ausdrücke:

1. True and True
2. True and False
3. True or False
4. not (True or False)
5. not True and not False
6. not (True and False)
7. not True or not False
8. not (False and False)
9. not False or not False
10. not (True and False) == ((not True) or (not False))
11. not (True or False) == ((not True) and (not False))
12. not (False and True) == ((not False) or (not True))
13. not (False or True) == ((not False) and (not True))

2) Die „de-morgansche Regeln“ sind zwei grundlegende Regeln in Booleschen Algebren und können in der Praxis sehr nützlich sein, um komplexere Prüfungen in Programmen zu vereinfachen. Es gelten im Allgemeinen folgende zwei Äquivalenzen:

nicht (a und b) ist äquivalent zu [(nicht a) oder (nicht b)]
nicht (a oder b) ist äquivalent zu [(nicht a) und (nicht b)]

Oder formaler notiert:

$$\neg(a \wedge b) \Leftrightarrow \neg a \vee \neg b$$
$$\neg(a \vee b) \Leftrightarrow \neg a \wedge \neg b$$

Hierbei wird `not` durch \neg , `and` durch \wedge , sowie `or` durch \vee dargestellt.

Eventuell ist Ihnen schon aufgefallen, dass Sie im letzten Übungsteil gleiche Ergebnisse für die Ausdrücke 4. und 5., 6. und 7. sowie 8. und 9. erhalten haben.

Erkennen Sie, warum die Ausdrücke die gleichen Ergebnisse liefern?

Bitte zeigen Sie unter Verwendung der de-morganschen Regeln, dass die Ausdrücke 4. und 5., 6. und 7. sowie 8. und 9. tatsächlich äquivalent sind.

- 3) `varA` und `varB` sind Bool'sche Variablen in Ihrem Programm. Welche der folgenden Aussagen sind immer wahr, welche sind immer falsch, welche sind vom Wert von `varA` und `varB` bzw. der Kombinationen von `varA` und `varB` abhängig?

1. `varA and not varA`
2. `varA or not varA`
3. `varA and varA`
4. `not (not varA)`
5. `varA and not varB`
6. `not varA and varB`

Übung 6

XOR

In manchen Situationen möchte man als Programmierer eine Kombination aus zwei Bool'schen Werten nur wahr werden lassen, wenn genau nur einer der beiden Werte wahr ist und der andere falsch. Diesen Operator bezeichnet man als XOR oder „Exklusiv-Oder“:

Wahr	XOR	Wahr	→ Falsch
Wahr	XOR	Falsch	→ Wahr
Falsch	XOR	Wahr	→ Wahr
Falsch	XOR	Falsch	→ Falsch

Wie würden Sie diesen Operator in Python unter Verwendung der Operatoren `and`, `not` und `or` darstellen? Nehmen Sie an, in Ihrem Programm gibt es zwei Bool'sche Variablen `varA` und `varB`, die Sie durch die Operation verknüpfen wollen.

Hinweise:

- Bedenken Sie, dass XOR eine „besondere“ bzw. erweiterte Form des `or` Operators darstellt – versuchen Sie daher auf Basis des `or` Operators gedanklich zu starten
- Durchdenken Sie bitte nochmals Übung 5.3, insbesondere die Ausdrücke 5. und 6.

Übung 7

Lügendetektor

Sie wollen die Aussagen von zwei Personen prüfen und stellen ihnen jeweils dieselbe Frage. Die Personen sollen einen Schalter auf „ein“ schalten, wenn sie die Frage mit Ja beantworten und auf „aus“, wenn sie die Frage mit nein beantworten. Eine Lampe leuchtet, sobald es widersprüchliche Aussagen zwischen den beiden Personen gibt. Die erste Person bedient Schalter A, die zweite Person bedient Schalter B.

Schalter A: ein, Schalter B: aus	→ Licht an / Widerspruch erkannt
Schalter A: aus, Schalter B: ein	→ Licht an / Widerspruch erkannt
Schalter A: aus, Schalter B: aus	→ Licht aus / Übereinstimmende Aussage
Schalter A: ein, Schalter B: ein	→ Licht aus / Übereinstimmende Aussage

Schreiben Sie ein kleines Programm, welches zwei Eingaben abfragt und die Rückmeldung gibt, ob die Leuchte – als Indikator für zwei widersprüchliche Antworten - an oder aus ist.

Zur Überprüfung Ihres Programms folgende Testausgaben:

Test-Case 1:

```
Person A / Schalter A (1:ein/2:aus): 1
Person B / Schalter B (1:ein/2:aus): 2
Die Leuchte ist an - Die Aussagen widersprechen sich!
```

Test-Case 2:

```
Person A / Schalter A (1:ein/2:aus): 2
Person B / Schalter B (1:ein/2:aus): 2
Die Leuchte ist aus - Die Aussagen stimmen überein!
```

Test-Case 3:

```
Person A / Schalter A (1:ein/2:aus): 1
Person B / Schalter B (1:ein/2:aus): 1
Die Leuchte ist aus - Die Aussagen stimmen überein!
```