

Informatik I: Einführung in die Programmierung

Prof. Dr. Peter Thiemann
Dr. Daniel Büscher, Hannes Saffrich
Wintersemester 2019

Universität Freiburg
Institut für Informatik

Übungsblatt 2 – Lösungen

Abgabe: Montag, 04.11.2019, 9:00 Uhr morgens

Legen Sie in Ihrer Arbeitskopie ein Verzeichnis `sheet02` an und bearbeiten Sie dort die folgenden Aufgaben. Vergessen Sie nicht das Verzeichnis und die enthaltenen Dateien zur Arbeitskopie hinzuzufügen und mit `svn commit` zu publizieren, sonst können Ihre Abgaben nicht bearbeitet werden.

Aufgabe 2.1 (Arithmetische Ausdrücke; Datei: `arithmetik.txt`; Punkte: 4)

Bestimmen Sie nach jeder der folgenden Wertzuweisungen an die Variable `res` den Typ von `res`. Geben Sie jeweils eine kurze Erläuterung, warum das so ist. Konsultieren Sie dazu in der Python-Shell auch die Hilfe zu den jeweiligen arithmetischen Funktionen.

- (a)

```
>>> from math import log2
>>> res = 6 * int(log2(128))
```
- (b)

```
>>> from math import floor, sin, pi
>>> res = floor(2.3 * 7) * (10 * sin(pi/4) // 3)
```
- (c)

```
>>> from math import tan, radians
>>> res = tan(radians(60))*2j
```
- (d)

```
>>> res = 6 * round(2.1, 1) // 1 + 1.0
```

Lösung:

- (a) `res` hat den Typ `int`, da

```
log2()  -> float
int()   -> int
int*int -> int
```
- (b) `res` hat den Typ `float`, da

```
floor()  -> int
sin()    -> float
int*float -> float
float//int -> float
int*float -> float
```
- (c) `res` hat den Typ `complex`, da

```
radians() -> float
tan()     -> float
```

```
float**complex -> complex
```

(d) `res` hat den Typ `float`, da

```
round()      -> float
int*float    -> float
float//int   -> float
float+float  -> float
```

Hinweis zu Aufgabe 2.2 und 2.3 (Text-Eingabe mit `input`)

In der Vorlesung wurde die Funktion `print` vorgestellt, die es einem Python-Script ermöglicht dem Benutzer eine Text-Ausgabe zu präsentieren.

Die Funktion `input` stellt das Gegenstück zu `print` dar und ermöglicht es den Benutzer nach einer Text-Eingabe zu fragen:

```
>>> s = input("Geben Sie etwas ein: ")
Geben Sie etwas ein: foo123
>>> s
'foo123'
```

Der Aufruf von `input` erzeugt dabei zunächst die Ausgabe

`Geben Sie etwas ein:`

und wartet dann bis der Benutzer eine beliebige Tasteneingabe tätigt (hier `foo123`) und mit einem Zeilenumbruch (Enter) die Eingabe beendet. Die vom Benutzer eingegebenen Zeichen werden dann als String zurückgegeben (hier `'foo123'`).

Aufgabe 2.2 (Fahrenheit nach Celsius; Datei: `fahrenheit.py`; Punkte: 7)

Schreiben Sie ein Python-Script `fahrenheit.py`, welches den Benutzer dazu auffordert einen Fahrenheit-Wert (Fließkommazahl) einzugeben und anschließend den entsprechenden, auf zwei Nachkommastellen gerundeten, Celsius-Wert ausgibt. Der Aufruf des Scripts, z.B. durch `python3 fahrenheit.py`, bei dem der Benutzer den Fahrenheit-Wert 9.11 eingibt, soll dabei folgende Ausgabe erzeugen:

```
Fahrenheit: 9.11
Celsius: -12.72
```

Hinweis. Zum Runden können Sie die Funktion `round` verwenden. Konsultieren Sie hierzu `help(round)` im Python Interpreter.

Hinweis. Strings können wie folgt zu Fließkommazahlen konvertiert werden:

```
>>> float('2.340')
2.34
```

Versucht man einen String, der keiner Fließkommazahl entspricht, zu konvertieren, wird die Ausführung des Programms durch eine Ausnahme zum Absturz gebracht:

```
>>> float('not a number')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: could not convert string to float: 'not a number'
```

In Ihrem Python-Script dürfen Sie dies ignorieren. Sie können also annehmen, dass der Benutzer stets eine Fließkommazahl eingibt.

Lösung:

```
fahrenheit = float(input("Fahrenheit: "))
celsius = (fahrenheit - 32) * (5 / 9)
print("Celsius:", round(celsius, 2))
```

Aufgabe 2.3 (Binärdarstellung; Datei: `binary.py`; Punkte: 7)

Schreiben Sie ein Python-Script `binary.py`, welches den Benutzer dazu auffordert eine positive ganze Dezimalzahl einzugeben und die letzten acht Stellen ihrer Binärdarstellung ausgibt.

Sie können dabei wie folgt vorgehen:

- (a) Verwenden Sie `input`, um den Benutzer nach einer Zahl zu fragen, und weisen Sie den resultierenden Eingabe-Text der Variable `s` zu.
- (b) Verwenden Sie `int`, um den Eingabe-Text `s` zu einer ganzen Zahl zu konvertieren, und weisen Sie das Ergebnis der Variable `n` zu.
- (c) Zerlegen Sie `n` in die einzelnen Ziffern ihrer Binärdarstellung und weisen Sie diese jeweils den Variablen `b0` bis `b7` zu.
- (d) Geben Sie die Variablen `b0` bis `b7` in geeigneter Reihenfolge in einer Zeile aus.

Nutzen Sie für Ihre Implementierung weder Imports, Schleifen noch eigene Funktionen. Für die Dezimalzahl 33 sollte sich beispielsweise folgende Ausgabe ergeben:

```
Dezimalzahl: 33  
Binärdarstellung: 0 0 1 0 0 0 0 1
```

Lösung:

```
s = input("Dezimalzahl: ")  
n = int(s)  
b0 = (n // 2 ** 0) % 2  
b1 = (n // 2 ** 1) % 2  
b2 = (n // 2 ** 2) % 2  
b3 = (n // 2 ** 3) % 2  
b4 = (n // 2 ** 4) % 2  
b5 = (n // 2 ** 5) % 2  
b6 = (n // 2 ** 6) % 2  
b7 = (n // 2 ** 7) % 2  
print("Binärdarstellung:", b7, b6, b5, b4, b3, b2, b1, b0)
```

Aufgabe 2.4 (Erfahrungen; Datei: `erfahrungen.txt`; Punkte: 2)

Legen Sie im Unterverzeichnis `sheet02` eine Textdatei `erfahrungen.txt` an. Notieren Sie in dieser Datei kurz Ihre Erfahrungen beim Bearbeiten der Übungsaufgaben (Probleme, Bezug zur Vorlesung, Interessantes, benötigter Zeitaufwand, etc.).