

Informatik I: Einführung in die Programmierung

Prof. Dr. Bernhard Nebel
Tim Schulte, Thorsten Engesser
Wintersemester 2017/2018

Universität Freiburg
Institut für Informatik

Übungsblatt 2

Abgabe: Freitag, 3. November 2017, 20:00 Uhr

WICHTIGE HINWEISE: Zur Bearbeitung der Übungsaufgaben legen Sie bitte ein neues Unterverzeichnis `sheet02` im Wurzelverzeichnis Ihrer Arbeitskopie des SVN-Repositories an. Ihre Lösungen werden dann in Dateien in diesem Unterverzeichnis erwartet. Beachten Sie bitte bei allen Aufgaben die *Hinweise zur Bearbeitung der Übungsaufgaben* unter der folgenden URL:

<http://gki.informatik.uni-freiburg.de/teaching/ws1718/info1/guide/hinweise.html>

Bewertet wird bei allen Aufgaben die letzte Version, die zur Deadline des Übungsblattes auf dem SVN-Server eingereicht ist.

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 = 2 ** log2(256)
```
- (b)

```
>>> from math import pi, sin, cos, radians
>>> res = cos(pi/4)**2 + sin(radians(45))**2
```
- (c)

```
>>> from math import sqrt, floor, ceil
>>> res = sqrt(floor(3.7)) // 2 or ("42" and ceil(1.1))
```
- (d)

```
>>> res = 3 * round(0.1, 1) == round(.3, 1)
```

Aufgabe 2.2 (Variablen und ihre Werte; Datei: `variablen.txt`; Punkte: 4)

Angenommen, Sie geben in der Python-Shell die folgende Befehlssequenz ein. Welche der darin verwendeten Namen `x`, `y`, `z`, `foo` sind an den mit (1), (2) und (3) markierten Stellen sichtbar (d.h. definiert)? Welche Werte haben die an diesen Stellen jeweils sichtbaren Variablen?

Bearbeiten Sie die Aufgabe zunächst auf Papier und überprüfen Sie Ihre Antworten anschließend mit der Python-Shell oder dem Python-Tutor.

```

>>> x = 12
>>> y = "spam"
>>> def foo(z):
...     if z % 2 == 0:
...         global y
...         y *= 2
...     return z // 2
...
>>> # (1)

>>> z = foo
>>> x = z(x)
>>> # (2)

>>> while x > 1:
...     y += "ham"
...     x = foo(x)
...
>>> # (3)

```

Aufgabe 2.3 (Stellenzahl in verschiedenen Stellenwertsystemen; Dateien: `lennum.py`, `test_lennum.txt`; Punkte: 5)

Definieren Sie eine Funktion `baselen`, die bei Eingabe einer nicht-negativen Zahl `n` und einer Zahl `base` ≥ 2 (beide vom Typ `int`) die Anzahl der Stellen von `n` im Stellenwertsystem mit der Basis `base` zurückgibt. Für ungültige Eingaben von `n` oder `base` soll `None` zurückgegeben werden.

Hinweis: Zu Stellenwertsystemen siehe:

<https://de.wikipedia.org/wiki/Stellenwertsystem>.

Definieren Sie die Funktion ausgehend von folgendem Template:

```

def baselen(n, base):
    """Calculate the length of a natural number in positional notation.

    Args:
        n (int): a natural number.
        base (int): base for the positional notation.

    Returns:
        int or None: int if `n` >= 0 and `base` >= 2, None otherwise.

    """
    # Anstelle dieses Kommentars folgt hier Ihre Definition

```

Testen Sie Ihre Funktion an geeigneten Beispielen, also z.B. mit:

```

>>> baselen(0, 7) == 1
True

```

```

>>> baselen(3, 2) == 2
True
>>> baselen(100, 8) == 3
True
>>> baselen(10000, 10) == 5
True
>>> baselen(0xABCDEF, 16) == 6
True
>>> baselen(-0b1010, 2) is None
True
>>> baselen(10, 1) is None
True

```

Bearbeiten Sie diese Aufgabe in der Python-IDE IDLE. Öffnen Sie in IDLE eine neue Datei mit dem Namen `lennum.py` und definieren Sie in dieser Datei Ihre Funktion `baselen`. Benutzen Sie dann die IDLE-Funktion `Run module (F5)`, um Ihre Funktion in der Python-Shell zu testen (mindestens vier Test-Beispiele wie oben). Sichern Sie die Shell, in der Sie Ihre Tests durchgeführt haben in der Datei `test_lennum.txt` und löschen Sie aus der Datei *alle* Zeilen mit Ausnahme Ihrer Test-Beispiele und deren Ausgaben in der Python-Shell (wie in obigem Beispiel). Committen Sie beide Dateien zum SVN-Repository.

Aufgabe 2.4 (Funktionen: Celsius nach Fahrenheit; Datei: `ctof.py`; Punkte: 2 + 3)

Bearbeiten Sie die folgenden beiden Teilaufgaben in IDLE. Öffnen Sie in IDLE eine neue Datei mit dem Namen `ctof.py` und definieren Sie in dieser Datei die folgenden zwei Funktionen.

- (a) Definieren Sie eine Funktion `cels_to_fahrenh(x)`, die bei Eingabe eines numerischen Wertes, gelesen als Temperaturwert in Grad Celsius, den entsprechenden Temperaturwert in Grad Fahrenheit als Fließkommazahl zurückgibt (sofern dies für den eingegeben Wert möglich ist).
- (b) Definieren Sie eine Funktion `print_ctof_table(x, y, z)`, die eine Tabelle von Temperaturwerten in Grad Celsius (in aufsteigender Reihenfolge) und entsprechenden Werten in Grad Fahrenheit in der Ausgabe erzeugt. Um die Tabelle variabel zu gestalten, soll die Funktion mit drei Argumenten definiert werden, wobei an den Parameter `x` der erste (= kleinste) Celsius-Wert in der Tabelle, an `y` eine obere Schranke für die dargestellten Celsius-Werte, und an `z` die Schrittweite in den Celsius-Werten übergeben wird (alle Werte als *floats*). Falls die Schrittweite nicht positiv oder die obere Schranke kleiner als der erste Wert ist, soll in der Tabelle nur die Umrechnung für den kleinsten Celsius-Wert ausgegeben werden. Alle Zahlenwerte in der ausgegebenen Tabelle sollen auf eine Nachkommastelle gerundet werden. Die ausgegebene Tabelle muss für alle Celsius-Werte $0 \leq x \leq 99$ wohl-formatiert sein (siehe Beispiel).

In Ihrer Lösung dürfen als einzige Funktionen die in 2.2(a) definierte Funktion `cels_to_fahrenh`, die Funktionen `print`, `float` und `round`, sowie arithmetische, boolsche und Vergleichs-Operatoren verwendet werden. Das Gradzeichen kann durch den String (Zeichenkette) `\u00B0` ausgegeben werden. Prüfen Sie Ihre Lösung an geeigneten Beispielen.

```
>>> print_ctof_table(0.0, 99.0, 17.0)
```

```
-----  
Celsius   Fahrenheit  
-----  
  0.0°C    32.0°F  
 17.0°C    62.6°F  
 34.0°C    93.2°F  
 51.0°C   123.8°F  
 68.0°C   154.4°F  
 85.0°C   185.0°F  
-----
```

```
>>> print_ctof_table(0.0, 99.0, -13.0)
```

```
-----  
Celsius   Fahrenheit  
-----  
  0.0°C    32.0°F  
-----
```

```
now exiting Console...
```

Aufgabe 2.5 (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, benötigter Zeitaufwand nach Teilaufgabe, Bezug zur Vorlesung, Interessantes, etc.).