

Übung 1

Abgabe 24.4.2012

Aufgabe 1 – Einführung in Python

8 Punkte

- Installieren Sie die Programmiersprache Python auf Ihrem System (siehe www.python.org). Wenn Sie Linux verwenden, ist Python wahrscheinlich bereits installiert.
- Arbeiten Sie aus dem Python-Tutorium (siehe docs.python.org/tutorial/) die Abschnitte 1 bis 6 sowie 7.2 und 9.3 durch. Sie sollen dabei die angegebenen Beispiele auf Ihrer Maschine nachvollziehen. Die folgenden Unterabschnitte, die sich an fortgeschrittene Benutzer richten, können übersprungen werden: 2.2, 4.7, 5.1.3, 5.1.4, 5.6, 5.7, 5.8, 6.4.
- Implementieren Sie das Sieb des Eratosthenes zur Bestimmung von Primzahlen (den Algorithmus finden Sie im Internet). Geben Sie Ihre Lösung in einem File „sieve.py“ ab, das wie folgt benutzt werden kann:

```
>>> execfile("sieve.py") # load the function sieve()
>>> primes = sieve(1000) # return an array of all primes below 1000
```

Aufgabe 2 – Werte und Referenzen

8 Punkte

- Wenn Sie im Internet auf eine interessante Seite stoßen, die Sie sich merken wollen, können Sie entweder ein Lesezeichen setzen (bookmark), oder die Seite abspeichern (save page as). Dies entspricht genau der Unterscheidung zwischen Referenzen und Werten in der Programmierung. Beschreiben Sie Vor- und Nachteile beider Vorgehensweisen. Denken Sie dabei z.B. an Speicherverbrauch, Offline-Betrieb, Datenkonsistenz und die Möglichkeit, dass die Originalseite aktualisiert oder gelöscht werden könnte.
- In Python erfolgen Zuweisungen standardmäßig per Referenz.¹ Für alle Datentypen kann man eine Zuweisung per Wert erzwingen, wenn man das Python-Modul "copy" verwendet. Informieren Sie sich über dieses Modul (siehe docs.python.org/library/copy.html). Analog gilt für Vergleiche, dass der operator "==" (z.B. `a == b`) die Werte von zwei Variablen vergleicht, während der operator "is" (z.B. `a is b`) prüft, ob zwei Variablen dasselbe Objekt referenzieren. Führen Sie folgendes Experiment durch:

```
>>> import copy
>>>
>>> a = [1, 2, 3, 4]
>>> b = a
>>> c = copy.deepcopy(a)

>>> print a == b, a == c, b == c    # are they equal?
>>> print a is b, a is c, b is c    # are they identical?

>>> a[0] = 42                        # change array a
>>> print a == b, a == c, b == c    # are they still equal?
>>> print a is b, a is c, b is c    # are they identical?
>>> print a[0], b[0], c[0]          # which values do you get?
```

Beschreiben und begründen Sie, wie sich das Programm verhält.

¹ Ausnahmen bilden die Zahlen (Typen `bool`, `int` und `float`), die stets als Werte zugewiesen werden.

Aufgabe 3 – Dynamisches Array

8 Punkte

Wir haben in der Vorlesung das *dynamische Array* kennengelernt, das eine effiziente `append()`-Funktion implementiert: Immer wenn die Kapazität des internen Speichers erschöpft ist, wird ein neuer interner Speicherbereich der doppelten Größe angelegt. Die vorhandenen Daten aus dem alten Speicherbereich werden in den neuen kopiert, so dass der alte Speicher danach gelöscht werden kann (wir untersuchen später, im Kapitel „Effizienz“, warum dies ein geschicktes Vorgehen ist). Eine solche Array-Klasse kann folgendermaßen implementiert sein (vgl. die Vorlesung):

```
class DynamicArray:
    def __init__(self):
        self.capacity = 1      # we reserve memory for at least one item
        self.data = [None]*self.capacity  # the internal memory
        self.size = 0          # no item has been inserted yet
    def __getitem__(self, index):  # __getitem__ implements array[index]
        if index < 0 or index >= self.size:
            print "index out of range"
            return None
        return self.data[index]
    def append(self, item):
        if self.capacity == self.size:      # internal memory is full
            ... # your code to increase the capacity
        self.data[self.size] = item
        self.size += 1
```

Implementieren Sie den in `append()` fehlenden Code, um die Kapazität bei Bedarf zu verdoppeln. Erzeugen Sie ein solches Array und führen Sie einige Einfügungen durch. Geben Sie nach jeder Einfügung die aktuellen Werte von `size` und `capacity` aus und überprüfen Sie, dass sich Ihre Klasse wie erwartet verhält. Geben Sie Ihren Code im File „array.py“ ab.