

Programmieren 1 (PRG1)

Übung 5

5.1

a)

Funktion nimmt Parameter entgegen und gibt *return value* aus. (Bsp.: `print()`)

Prozedur führt Reihe von Operationen an spezifischem Objekt aus und gibt keinen Return Wert zurück (Bsp.: `list.reverse()`)

b)

Prozeduren können Werte im aufrufenden Programm ändern (siehe `list.reverse()`). Da Prozeduren keinen Return Wert ausgeben, hätten sie keine Möglichkeit Einfluss auf ein Programm zu nehmen, dass sie aufruft, wenn dies nicht möglich wäre.

c)

*call by reference:*

Aktuelle Parameter werden nicht als fester Wert, sondern als Speicherort der entsprechenden Variable übergeben. Eine Veränderung dieses Parameters, verändert auch den Speicherort und kann somit gleichzeitig das aufrufende Programm beeinflussen.

*Call by value:*

Aktueller Parameter wird in Form einer Kopie übergeben und kann verändert werden, ohne die originale Variable zu beeinflussen. Daraus folgt, dass Änderungen der Originalen Variable nicht möglich sind und somit das aufrufende Programm nicht beeinflusst werden kann (Stichwort „Kapselung“).

(Quelle: Vorlesungsskript „07- Funktionen\_und\_Prozeduren“; 7.2 Parameterübergabe – 7.3 Funktionen versus Prozeduren; Seiten 4 – 6)

5.2

a)

Als Modul wird ein Programmcode (Script) bezeichnet, das Statements und Funktionen enthält. Der Code muss außerdem als korrekter Dateityp (bei Python als .py Datei) gespeichert werden.

b)

Ein Modul kann in ein anderes Programm (oder einen Interpreter) importiert werden. Man erhält so Zugriff auf alle Funktionen und Statements des importierten Moduls. Dies ermöglicht eine Zentralisierung (Funktionen müssen nicht in jedes neue Script, neu kopiert werden). Außerdem werden Veränderungen/Verbesserungen/Reparaturen an der Funktion, direkt von allen Programmen übernommen, die das modifizierte Modul importieren.

c)

In Python werden Module über das abspeichern von Dateien mit der Endung „.py“ realisiert. Diese Module können über „import *filename*“ in ein Programm importiert werden (alternativ können einzelne Funktionen über „from *filename* import *funktion*“ importiert werden).

(Quelle: <https://docs.python.org/2/tutorial/modules.html>; Datum: 23.11.17; Uhrzeit: 18:40 Uhr)

### 5.3

a)

Unter Namensraum versteht man die Gesamtheit aller in einem Programm enthaltenen Funktionen, Parameter und Variablen. Hierbei ist zwischen globalen, lokalen und eingebauten Namensräumen zu unterscheiden.

b)

Der Python Interpreter löst Namensräume von innen nach außen auf. Hierbei wird zuerst (falls vorhanden) der Lokale Namensraum nach einer Variablen durchsucht (der Namensraum einer Funktion), dann die Umliegenden lokalen Namensräume (andere Funktionen, auch von innen nach außen), dann der globale Namensraum (das Modul) und schließlich der eingebaute Namensraum (`__builtin__` des Moduls).

c)

Eine Variable kann mithilfe von „global *variable*“ definiert werden. Hierbei beginnt der Interpreter seine „Suche“ nach der Variablen nicht im Lokalen, sondern im globalen Raum. Die Nutzung dieser Methode ist im Rahmen dieser Veranstaltung verboten, da auf diese Weise globale Variablen durch das aufrufen einer Funktion verändert werden können (keine Kapselung; sehr Anfällig für schwer auffindbare Fehler).

(Quelle: Vorlesungsskript „07- Funktionen\_und\_Prozeduren“; 8.1 Namensräume; Seiten 11 – 13)