

Einführung in die Programmierung mit Skriptsprachen

V09: Stack, Objektreferenzen vom 19.12.2022

Aufgabenblatt 8

1. Spickzettel

Der Spickzettel 8 befindet sich im Ordner unter 8. Spickzettel.pdf.

2. Gültigkeitsbereiche

Programm befindet sich in der Python Datei „Gueltingkeitsbereiche“ Datei.py“.

3. Parameterübergabe in Python

Programm befindet sich in der Python Datei „Parameteruebergabe_in_Python“ Datei.py“.

Stack		Heap		
Variable	Referenz	Objektreferenz	Datentyp	Inhalt
intZahl	100 ,200	100 ,150	int	21
stringName	200 ,300	200	int	100
liste	500 , 700	300	str	Pascal
liste2	600	400	str	Daniel
		500	list	[->300, -> 150 ,->200]
		600	list	[-> 150 ,->400]
		700	list	[->150,->400]

Programmablauf:

Zuerst zeigt die Variable „intZahl“ auf Objektreferenz „100“, was im Heap den Wert „21“ hat. Als nächstes zeigt die Variable „stringName“ auf Objektreferenz „200“, im Heap. Der Inhalt ist „21“.

Danach wird eine Liste erzeugt mit der Variable „liste“, die auf die Objektreferenz „500“ schaut. Im Inhalt sind dann nochmal zwei weitere Objektreferenzen die auf „100“ und die „300“ zeigt. Danach ändere ich den Wert der Variable „intZahl“ und dadurch haben wir eine neue Objektreferenz. Die Objektreferenz „100“, wird im Stack und im Heap gelöscht. Bei der „stringName“ Variable ändere ich als nächstes den Inhalt und habe somit eine neue Objektreferenz, die alte wird gelöscht. Die Zeile mit der Objektreferenz „200“ wird im Heap gelöscht.

Bei der Änderung des Inhalts, kann man genau sehen das es sich um Call by Value handelt, da es sich immer auf eine neue Objektreferenz bezieht, wenn man den Wert ändert.

Das nächste was ich Prüfe ist die Liste. Ich ersetze eine Stelle in der Liste und ändere somit nur den Inhalt im Heap. Im Stack gab es keine Änderung. Da im Stack die Objektreferenz „150“ nicht mehr verwendet wird, wird sie im Heap gelöscht.

Hier kann man schon sehen das es sich um Call by Reference handelt, da sich die Objektreferenz im Stack bei der Variable „liste“ nicht ändert.

Zuletzt erstelle ich eine neue Liste mit der Variable „liste2“. Die einen anderen Inhalt als „liste“ hat.

Sobald „liste“ gleich „liste2“ gesetzt wurde. Wird die Objektreferenz im Stack geändert und im Heap gelöscht.

Danach hat die „liste“ eine neue Objektreferenz. Somit kann man sehen das es sich hier um Call by Value handelt.