

Programmazione e Algoritmi 1

A.A. 2023/24 — Soluzioni esercizi 1 e 2 del compito del 20 giugno 2024

prof. Gianluca Amato

Versione Python

Esercizio 1 (5 punti)

Un tipo di dato si chiama *immutabile* quando non è possibile modificare un oggetto di quel tipo una volta creato. Tipici esempi di dati immutabili sono interi (`int`), booleani (`bool`), stringhe (`str`), numeri in virgola mobile (`float`) e tuple. Se invece è possibile modificare il dato dopo la creazione, si parla di tipo di dato *mutabile*. Un esempio tipico di dato mutabile è la lista.

Bisogna stare attenti a non confondere la modifica di un oggetto di un certo tipo dalla modifica di una variabile. Ricordiamo infatti che una variabile è solo un puntatore ad un oggetto. Una istruzione come `a = a + 2` non modifica il numero puntato `a` ma crea un nuovo numero e modifica la variabile `a` in modo che punti ad esso. Di contro, se `l` è una lista, l'istruzione `l.append(2)` non crea una nuova lista, ma modifica la lista puntata da `l` aggiungendo un elemento. La variabile `l` continua a puntare allo stesso oggetto a cui puntava prima, solo che ora questo oggetto è cambiato.

Esercizio 2 (8 punti)

riga programma		valore variabili	note
9		<code>x=[1, 2, 3, 4]</code>	
10		<code>x=[1,2,3,4] y=[50, 60, 70, 80]</code>	
11		<code>x=[1,2,3,4] y=[50,60,70,80]</code>	call mistero
	1	<code>l1=[1,2,3,4] l2=[50,60,70,80]</code>	
	2	<code>l1=[1,2,3,4] l2=[50,60,70,80] i=0</code>	
	3	<code>l1=[51,2,3,4] l2=[50,60,70,80] i=0</code>	
	2	<code>l1=[51,2,3,4] l2=[50,60,70,80] i=1</code>	
	3	<code>l1=[51,62,3,4] l2=[50,60,70,80] i=1</code>	
	2	<code>l1=[51,62,3,4] l2=[50,60,70,80] i=2</code>	
	3	<code>l1=[51,62,73,4] l2=[50,60,70,80] i=2</code>	
	2	<code>l1=[51,62,73,4] l2=[50,60,70,80] i=3</code>	
	3	<code>l1=[51,62,73,84] l2=[50,60,70,80] i=3</code>	
	2	<code>l1=[51,62,73,84] l2=[50,60,70,80] i=3</code>	return
11		<code>x=[51,62,73,84] y=[50,60,70,80]</code>	<code>x</code> è un alias di <code>l1</code>
12		<code>x=[51,62,73,84] y=[50,60,70,80]</code>	call mistero2
	5	<code>l=[51,62,73,84]</code>	
	6	<code>l=[51,62,73,84] x=51</code>	
	7	<code>l=[51,62,73,84] x=51</code>	print 51
	6	<code>l=[51,62,73,84] x=62</code>	
	7	<code>l=[51,62,73,84] x=62</code>	print 62
	6	<code>l=[51,62,73,84] x=73</code>	
	7	<code>l=[51,62,73,84] x=73</code>	print 73
	6	<code>l=[51,62,73,84] x=84</code>	
	7	<code>l=[51,62,73,84] x=84</code>	print 84
	5	<code>l=[51,62,73,84] x=84</code>	return
12		<code>x=[51,62,73,84] y=[50,60,70,80]</code>	

Notare che quando la funzione `mistero` termina, le modifiche apportate ad `l1` sono visibili anche su `x`. Questo perché, dopo la chiamata di funzione, `x` è un alias di `l1`: le due variabili identificano esattamente la stessa lista.

Versione Java

Esercizio 1 (5 punti)

Lo svolgimento dell'esercizio è identico a quello della versione Python. L'unica differenza potrebbe essere negli esempi, visto che in Java non esistono le tuple e che il tipo lista è rimpiazzato dal tipo array.

Esercizio 2 (8 punti)

L'esercizio è molto simile a quello in Python, cambiano quasi esclusivamente i numeri di riga

riga programma				valore variabili	note
15				<code>x={1, 2, 3, 4}</code>	
16				<code>x={1,2,3,4} y={50, 60, 70, 80}</code>	
17				<code>x={1,2,3,4} y={50,60,70,80}</code>	call mistero
	2			<code>l1={1,2,3,4} l2={50,60,70,80}</code>	
	3			<code>l1={1,2,3,4} l2={50,60,70,80} i=0</code>	
	4			<code>l1={51,2,3,4} l2={50,60,70,80} i=0</code>	
	3			<code>l1={51,2,3,4} l2={50,60,70,80} i=1</code>	
	4			<code>l1={51,62,3,4} l2={50,60,70,80} i=1</code>	
	3			<code>l1={51,62,3,4} l2={50,60,70,80} i=2</code>	
	4			<code>l1={51,62,73,4} l2={50,60,70,80} i=2</code>	
	3			<code>l1={51,62,73,4} l2={50,60,70,80} i=3</code>	
	4			<code>l1={51,62,73,84} l2={50,60,70,80} i=3</code>	
	3			<code>l1={51,62,73,84} l2={50,60,70,80} i=4</code>	
	5			<code>l1={51,62,73,84} l2={50,60,70,80}</code>	return
17				<code>x={51,62,73,84} y={50,60,70,80}</code>	<code>x</code> è un alias di <code>l1</code>
18				<code>x={51,62,73,84} y={50,60,70,80}</code>	call mistero2
	8			<code>l={51,62,73,84}</code>	
	9			<code>l={51,62,73,84} x=51</code>	
	10			<code>l={51,62,73,84} x=51</code>	print 51
	9			<code>l={51,62,73,84} x=62</code>	
	10			<code>l={51,62,73,84} x=62</code>	print 62
	9			<code>l={51,62,73,84} x=73</code>	
	10			<code>l={51,62,73,84} x=73</code>	print 73
	9			<code>l={51,62,73,84} x=84</code>	
	10			<code>l={51,62,73,84} x=84</code>	print 84
	11			<code>l={51,62,73,84}</code>	return
18				<code>x={51,62,73,84} y={50,60,70,80}</code>	

Notare che quando il metodo `mistero` termina, le modifiche apportate ad `l1` sono visibili anche su `x`. Questo perché, dopo la chiamata di funzione, `x` è un alias di `l1`: le due variabili identificano esattamente lo stesso array.