

hyväksymispäivä arvosana

arvostelija

COBOL ja Python: Tyypitys ja laskennan kapselointi

Erkki Heino, Tero Huomo, Eeva Terkki

Helsinki 12.2.2013

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1	Tyypijärjestelmät	1
1.1	COBOL	1
1.2	Python	1
1.2.1	Alkeis- ja perustyytit	1
1.2.2	Tyypitys	2
2	Laskennan kapselointi	2
2.1	COBOL	2
2.2	Python	3
3	Etuja ja haittoja	3
	Lähteet	4

1 Tyyppijärjestelmät

1.1 COBOL

COBOL on vahvasti ja staattisesti tyypitetty kieli [Wik13]. Olio-orientointunut COBOL (Object Oriented COBOL) sisältää sekä vahvan että heikon tyypityksen piirteitä. [AC96]

"In COBOL, there are really only three data types -

- numeric
- alphanumeric (text/string)
- alphabetic

The distinction between these data types is a little blurred and only weakly enforced by the compiler. For instance, it is perfectly possible to assign a non-numeric value to a data item that has been declared to be numeric.

The problem with this lax approach to data typing is that, since COBOL programs crash (halt unexpectedly) if they attempt to do computations on items that contain non-numeric data, it is up to the programmer to make sure this never happens."

- <http://www.csis.ul.ie/cobol/course/DataDeclaration.htm>

Eli kääntäjä ei tee tyypitarkistuksia (ei olisi staattinen)? Vai puhuiko teksti käyttäjän syötteistä? (Mikä COBOLin versio?)

1.2 Python

1.2.1 Alkeis- ja perustyytit

Pythonissa ei ole erikseen alkeistyypppejä, vaan Pythonissa kaikki on ilmaistu olioina tai olioiden välisinä suhteina [Pyt13a]. Jokaisella oliolla on *identiteetti* (identity), tyyppi ja arvo. Identiteetti on olion luonnin jälkeen muuttumaton. Identiteettiä voidaan ajatella olion osoitteena muistissa. Olion tyyppi on muuttumaton. Tietyissä kontrolloiduissa tilanteissa tyyppiä on mahdollista muuttaa, mutta usein tyyppimuunnoksilla voi olla odottamattomia seurauksia.

Pythonissa siis myös esimerkiksi None on olio. None on vastaavuus Java-kielen null-arvolle. Toisin kuin Pythonin None, Javan null ei kuitenkaan ole olio, eikä sillä ole tyyppiä.

Kielessä on valmiina useita kymmeniä sisäänrakennettuja tyyppejä, jotka on lueteltu Pythonin tyyppihierarkiassa [Pyt13a]. Valmiita numeerisia tyyppejä ovat esimerkiksi totuusarvot, kompleksiluvut, kokonaisluvut ja reaaliluvut. Tavalliset kokonaisluvut ovat Javan tapaan 32-bittisiä. Pythonin versiossa 3 ei ole kiinteän pituisia kokonaislukuja [Pyt13b]. Reaalilukujen ja kompleksilukujen arvoalueet voivat riippua käytettävästä konearkkitehtuurista, virtuaalimuistin määrästä sekä kääntäjän asetuksista. Pythonissa ei lisäksi ole valmiina olemassa erillistä Character-tyyppiä, vaan yksittäiset merkit ovat merkkijonoja.

Erilaiset laajennusmoduulit lisäävät valmiita luokkia Javan kirjastojen tapaan.

1.2.2 Tyypitys

Python on vahvasti tyyplitetty kieli [Stat09]. Tietyn tyyppiselle muuttujalle ei voida tehdä toisen tyyppin operaatioita ennen eksplisiittistä tyyppimuunnosta.

```
a = 5
b = "9"
c = a + int(b)
```

Esimerkissä `b` sisältää merkkijonon `"9"`, mutta yhteenlaskussa merkkijonosta jäsenetään kokonaisluku. Jos kokonaislukujäsennyksen jättää tekemättä, antaa ohjelma kyseisellä rivillä poikkeuksen.

Suoritusaikana muuttujan tyyppi ei ole sidottu, vaan muuttujaan voi dynaamisesti sitoa eri vaiheessa eri tyyppisiä olioita. Seuraavassa esimerkissä muuttuja `a` saa ensin kokonaislukuarvon 5. Sen jälkeen muuttujan `a` arvoksi muutetaan merkkijono `"hei"`.

```
a = 5
a = "Hei"
```

2 Laskennan kapselointi

2.1 COBOL

Muistilistaa:

3 Etuja ja haittoja

Lähteet

- AC96 Arranga, E. C. ja Coyle, F. P., Object-oriented cobol.
- Pyt13a Python v2.7.3 documentation language reference - data model, <http://docs.python.org/2/reference/datamodel.html>. [12.2.2013].
- Pyt13b Python v3.3 documentation language reference - data model, <http://docs.python.org/3.3/reference/datamodel.html>. [12.2.2013].
- Stat09 Static vs. dynamic typing of programming languages, <http://pythonconquerstheuniverse.wordpress.com/2009/10/03/static-vs-dynamic-typing-of-programming-languages/>, 2009. [11.2.2013].
- Wik13 Cobol, <http://en.wikipedia.org/wiki/COBOL>. [11.2.2013].