

hyväksymispäivä arvosana

arvostelija

## **COBOL ja Python: Tyypitys ja laskennan kapselointi**

Erkki Heino, Tero Huomo, Eeva Terkki

Helsinki 11.2.2013

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

|          |                              |          |
|----------|------------------------------|----------|
| <b>1</b> | <b>Tyypijärjestelmät</b>     | <b>1</b> |
| 1.1      | COBOL . . . . .              | 1        |
| 1.2      | Python . . . . .             | 1        |
| <b>2</b> | <b>Laskennan kapselointi</b> | <b>2</b> |
| 2.1      | COBOL . . . . .              | 2        |
| 2.2      | Python . . . . .             | 2        |
| <b>3</b> | <b>Etuja ja haittoja</b>     | <b>2</b> |
|          | <b>Lähteet</b>               | <b>3</b> |

# 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

Python on vahvasti tyypitetty 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:

- "Support for complexity management is minimal in COBOL, with no language features which support any structures larger than subprogram modules." ( <http://archive.adaic.com/docs/reports/lawlis/> )
- Funktioista COBOL ANSI-85 ( <http://public.support.unisys.com/aseries/docs/clearpath-mcp-13.0/pdf/86001518-310.pdf> )
- Object Oriented COBOL (2002) tukee nimettyjä funktioita ja funktioprototyypppejä, sekä tietysti olioita/perintää/polyformismia (<http://en.wikipedia.org/wiki/COBOL>)
- Kielessä ei luontaista tukea rinnakkaislaskennalle ( <http://archive.adaic.com/docs/reports/lawlis/> ) mutta osa vendoreista tarjoaa apuvälineitä rinnakkaisuuteen (esim. <http://supportline.microfocus.com/> )

### 2.2 Python

## 3 Etuja ja haittoja

## Lähteet

- AC96        Arranga, E. C. ja Coyle, F. P., Object-oriented cobol.
- 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>. [5.2.2013].