

hyväksymispäivä arvosana

arvostelija

## Otsikko

Erkki Heino

Tero Huomo

Eeva Terkki

Helsinki 4.2.2013

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1</b>	<b>Näkyvyysalueet</b>	<b>1</b>
1.1	COBOL . . . . .	1
1.2	Python . . . . .	1
<b>2</b>	<b>Sidonta</b>	<b>2</b>
2.1	Cobol . . . . .	2
2.2	Python . . . . .	2
	<b>Lähteet</b>	<b>4</b>

# 1 Näkyvyysalueet

## 1.1 COBOL

## 1.2 Python

Pythonin lohkorakenne on syvä, ja ohjelman suoritusaikana käytössä on ainakin kolme sisäkkäistä näkyvyysaluetta [Pyt13b]. Näkyvyysalueita käytetään dynaamisesti. Sisimmällä näkyvyysalueella ovat paikalliset nimet. Mahdollisilla funktioita ympäröivillä funktioilla on omat näkyvyysalueensa, joiden sisältämät nimet eivät ole paikallisia eivätkä globaaleja. Toiseksi uloimmalla näkyvyysalueella ovat moduulin globaalit nimet ja kaikkein uloimmalla kieleen rakennetut nimet.

Pythonissa luokan näkyvyysalueella määritellyt nimet eivät näy luokan metodeille [Pyt13a]. Metodin ensimmäinen argumentti, jolle on tapana antaa nimi `self`, edustaa luokan ilmentymää. Sen kautta metodi voi käyttää luokan ilmentymän muita metodeja ja attribuutteja.

Seuraavassa esimerkissä on kaksi sisäkkäistä funktiota:

```
def f1():  
    a = 1  
    def f2():  
        b = 2  
        print a + b  
    print a  
    f2()
```

`f1()`

Funktio `f1` määrittelee muuttujan `a` ja funktion `f2`, tulostaa `a`:n arvon ja kutsuu määrittelemäänsä funktiota. Funktio `f2` määrittelee muuttujan `b` ja tulostaa muuttujien `a` ja `b` arvojen summan. Ohjelma tulostaa luvut 1 ja 3. Muuttuja `a` on näkyvissä funktion `f1` ja sen sisäisten funktioiden sisällä. Muuttuja `b` puolestaan on paikallinen muuttuja, joka on näkyvissä vain `f2`-funktion sisällä. Siihen viittaaminen `f2`-funktion ulkopuolella johtaisi virhetilanteeseen.

Seuraavassa esimerkissä käytetään globaalia muuttujaa:

```
g = 1

def f3():
    global g
    g = 2
    print g

f3()
print g
```

Esimerkissä globaalin muuttujan `g` arvoksi alustetaan ensin luku 1. Funktio `f3` asettaa `g:n` arvoksi luvun 2 ja myös tulostaa muuttujan arvon. Kun funktiota `f3` kutsutaan ja sen jälkeen vielä tulostetaan `g:n` arvo, ohjelma tulostaa kaksi kertaa luvun 2. Funktio `f3` siis käsittelee globaalia muuttujaa. Avainsana `global` on tärkeä, sillä se ilmaisee, että kyseinen tunnus tulkitaan globaalin muuttujan tunnukseksi. Ilman koodiriviä `global g` funktion määritelmän sisällä oleva muuttuja olisi paikallinen muuttuja, ja esimerkkiohjelma tulostaisi luvut 2 ja 1.

Pythonissa kaikki asiat, jotka voidaan nimetä, ovat ensimmäisen luokan arvoja – myös funktiot, metodit ja moduulit [Gui09].

## 2 Sidonta

### 2.1 Cobol

### 2.2 Python

Python on dynaamisesti sidottu kieli [Ste12]. Jokainen muuttujanimi sidotaan objektiin. Jos muuttujaa ei ole sidottu objektiin, muuttujalla on `Null`-arvo. Suoritusaikana muuttujan tyyppi ei ole sidottu, vaan muuttujaan voi dynaamisesti sitoa eri

vaiheessa eri tyyppisiä objekteja.

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

Esimerkissä muuttuja `a` saa ensin kokonaislukuarvon 5. Sen jälkeen muuttujan `a` arvoksi muutetaan merkkijono `"hei"`.

Python on myös vahvasti tyyplitetty kieli [Ste12]. 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 kokonaislukujäsenennyksen jättää tekemättä, antaa ohjelma kyseisellä rivillä poikkeuksen.

## Lähteet

- Pyt13a     The python language reference – execution model, <http://docs.python.org/2/reference/executionmodel.html#naming-and-binding>. [4.2.2013].
- Pyt13b     The python tutorial – classes, <http://docs.python.org/2/tutorial/classes.html>. [2.2.2013].
- Ste12     Static vs. dynamic typing of programming languages, <http://pythonconquerstheuniverse.wordpress.com/2009/10/03/static-vs-dynamic-typing-of-programming-languages/>. [4.2.2013].
- Gui09     van Rossum, G., First-class everything, <http://python-history.blogspot.fi/2009/02/first-class-everything.html>, 2009. [2.2.2013].