

hyväksymispäivä arvosana

arvostelija

## **COBOL ja Python: Datan kapselointi**

Erkki Heino, Tero Huomo, Eeva Terkki

Helsinki 19.2.2013

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1</b>	<b>Datan kapselointi</b>	<b>1</b>
1.1	COBOL . . . . .	1
1.1.1	Olio-orientoitunut COBOL . . . . .	2
1.2	Python . . . . .	2
<b>2</b>	<b>Etuja ja haittoja</b>	<b>2</b>
	<b>Lähteet</b>	<b>3</b>

# 1 Datan kapselointi

## 1.1 COBOL

COBOLissa tietoa voidaan ryhmitellä muodostamalla tietueita (record) ja ryhmiä (group). DATA DIVISION -osiossa käytettävät muuttujien tasonumerot (level number) määrittävät, minkälaisia kokonaisuuksia muuttujat muodostavat. Seuraavassa esimerkissä on kuvattu opiskelijatietue.

```
DATA DIVISION.
WORKING-STORAGE SECTION.
01 OPISKELIJA.
    05 NIMI.
        10 ETUNIMI PIC A(10).
        10 SUKUNIMI PIC A(10).
    05 NUMERO PIC 9(10).
    05 SUKUPUOLI PIC A.
        88 MIES VALUE "M".
        88 NAINEN VALUE "N".
```

OPISKELIJA on tietue, joka koostuu nimestä, opiskelijanumerosta ja sukupuolesta. Nimi on jaettu kahteen osaan: etunimi ja sukunimi.

Tasonumerot kuvaavat tiedon hierarkian, mutta tietyillä numeroilla on erityismerkitys. Hierarkiaa kuvaavat tasonumerot 02-49. Tasonumero 01 tarkoittaa, että kyseessä on tietue. Merkitsemällä tasonumeroksi 66 voidaan edeltävälle muuttujalle antaa toinen nimi. 77 kuvaa muuttujaa, joka ei ole osa mitään rakennetta. Tason 88 avulla voidaan määrittää muuttujaan liittyviä ehtoja: yllä olevassa esimerkissä opiskelijan sukupuoli on mies, jos muuttujan SUKUPUOLI arvo on M, ja nainen, jos arvo on N [Ans74, s. I-84]. Esimerkiksi voitaisiin tarkistaa, onko opiskelija mies:

```
IF MIES OF OPISKELIJA
```

Todellisuudessa tietue ei ole juuri muuta kuin merkkijono, jonka eri osat kuvaavat tietueen kenttiä. Tietueelle voidaan antaa arvot kenttä kerrallaan:

```
MOVE "ARTO" TO ETUNIMI OF OPISKELIJA.
```

Toisaalta koko tietueelle voidaan antaa arvo kerralla:

```
MOVE "ARTO      WIKLA      1234567890M" TO OPISKELIJA.
```

Esimerkin tietue sisältää kerrallaan yhden opiskelijan tiedon — tällaisenaan koodissa ei siis voida käsitellä useampaa opiskelijaa samanaikaisesti. Rakenne voidaan määritellä taulukoksi (table), jolloin samaa rakennetta voidaan käyttää useamman samanrakenteisen tiedon käsittelemiseen ilman saman rakenteen uudelleenmäärittelyä.

Muuttuja määritellään taulukoksi `OCCURS`-lauseen avulla. `OPISKELIJA` voitaisiin muuttaa taulukoksi muokkaamalla määrittelyä seuraavasti:

```
01 OPISKELIJA OCCURS 5 TIMES.
```

Esimerkiksi taulukon toiseen opiskelijaan viitattaisiin indeksillä 2: `OPISKELIJA(2)`. Taulukon sisällä voidaan myös määritellä sisäkkäisiä taulukoita.

### 1.1.1 Olio-orientoitunut COBOL

COBOL ei alunperin ole olio-ohjelmointikielin, vaan tieto kapseloitiin tietueisiin ja ryhmiin. COBOLista on kuitenkin kehittynyt olio-ohjelmoinnin mahdollistava laajennos, Object-oriented COBOL, joka on lisännyt kieleen olio-ohjelmoinnin tärkeimmät piirteet.

Teron muistilistaa epämääräisiin lähteisiin:

- <http://supportline.microfocus.com/documentation/books/sx20books/opintr.htm>
- <http://www.objs.com/x3h7/oocobol.htm>
- <http://supportline.microfocus.com/documentation/books/nx60/opconc.htm>

(IBM:n Enterprise COBOL tukee myös olioita, mutta OO COBOLista eroten estää moniperinnän. - <http://pic.dhe.ibm.com/infocenter/pdthelp/v1r1/index.jsp?topic=>

## 1.2 Python

## 2 Etuja ja haittoja

## Lähteet

Ans74      *American national standard programming language COBOL.* ANSI,  
New York, USA, 1974.