Ohjelmointikielten periaatteet 2013

Jari Koskinen, Hansi Keijonen, Eero Laine

29. tammikuuta 2013

Sisältö

1 D-ohjelmointikieli

D-ohjelmointikieli on C/C++ kielten pohjalta kehitetty ja sen pääasiallisena kehittäjänä on alusta lähtien toiminut ohjelmointikielten kääntäjiin erikoistunut Walter Bright. Kehitystyö on alkanut vuonna 1999 ja ensimmäinen julkaisu kääntäjästä on tehty vuonna 2001 Brightin omistaman yhtiön Digital Marsin toimesta. Ensimmäinen vakiintunut versio 1.0 julkaistiin kuitenkin vasta vuonna 2007. Nykyinen versio on 2.0 ja kielen kehitys jatkuu edelleen, kun mukaan on liittynyt useita kehittäjiä Brightin lisäksi. Version 2.0 pohjalta on julkaistu vuonna 2010 Andrei Alexandrescun kirjoittama kirja: The D Programming Language, jonka myötä version voidaan katsoa vakiintuneen. Versio 1.0 kattoi kolme ohjelmointiparadigmaa: imperatiivinen, oliopohjainen ja metaohjelmointi. 2.0 versio toi mukanaan tuen funktionaaliselle ohjelmoinnille ja aktorimallin rinnakkaisohjelmointiin. Tässä kirjoituksessa käsitellään kielen versiota 2.0. Kieli sisältää paljon uudistuksia, jotka ovat parannuksia C/C++ kieliin tai muiden, kuten C#:n ja Javan hyödylliseksi havaittuja ominaisuuksia. Samalla kielestä on pyritty jättämään pois C/C++ kielten taaksepäin yhteensopivuuden asettamia rajoitteita.

D on moniparadigmainen, oliopohjainen ja imperatiivinen ohjelmointikieli. D tarjoaa tuen myös funktionaaliselle ja metaohjelmoinnille. D:llä kirjoitetut ohjelmat käännetään käännös- ja linkitysvaiheen kautta kohdeympäristön konekielelle. Ohjelmoijalla on täysi pääsy laitetasolle, joten se soveltuu myös käyttöjärjestelmien ohjelmointiin. Muistinhallinta voidaan jättää automatiikan huoleksi, koska D sisältää C#:n ja Javan tapaan roskienkeruun, mutta sallii muistinkäsittelyn myös C/C++:lle tyypillisen tapaan. Perintä on toteutettu yksinkertaisella perinnällä, rajapinnoilla ja mixineillä. Kielen syntaksi muistuttaa hyvin paljon C++ kieltä.

Alla on esimerkki ohjelmasta, joka palauttaa annettua lukua suuremman fibonaccin luvun.

```
1.  import std.stdio;
2.  import std.conv;
3.
4.  void main(string[] args) {
5.  uint f;
6.   try {
7.    f = parse!uint(args[1]);
8.   }
9.   catch {
10.   writeln("Error in arguments!");
11.   return;
12.  }
13.  writeln(args[1]);
14.  writeln(fib(f));
```

```
15. }
16.
17. ulong fib(uint n) {
18.  ulong iter(ulong fib_1, ulong fib_2) {
19.   return fib_2 > n ? fib_2 : iter(fib_1 + fib_2, fib_1);
20.  }
21.  return iter(1, 0);
22. }
```

Ohjelmakoodin toiminta on seuraavanlainen: riveillä 1 ja 2 ladataan käyttöön kirjastot syötön ja tulostuksen käsittelyä (std.stdio) ja tyyppimuunnoksia (std.conv) varten. Rivillä 7 muunnetaan saatu tekstimuotoinen parametri etumerkittömäksi kokonaisluvuksi. Rivit 9 - 11 ovat virheenkäsittelyä varten ja virheen sattuessa ohjelmasta poistutaan. Rivillä 14 kutsutaan fib-funktiota. Fib-funktiossa on riveillä 18 ja 19 oma sisäfunktio, joka suorittaa varsinaisen laskennan. Sisäfunktiota kutsutaan ulomman funktion viimeisellä rivillä 21. Kyseessä on tällöin häntärekursio, johon D-kielessä on optimoija.

2 Erlang ohjelmointikieli

Erlang on funktionaalinen kieli, joka on tarkoitettu hajautettujen järjestelmien ohjelmointiin. Erlang soveltuu tosiaikajärjestelmien ja laajasti skaalautuvien järjestelmien kehittämiseen, kuten puhelinkeskukset, pankkijärjestelmät ja sähköinen liiketoiminta. Kieli sisältää laajan tuen rinnakkaisuudelle ja mahdollistaa erittäin vikasietoisten järjestelmien kehittämisen. Kielen kehittämisen aloitetti Joe Armstrong vuonna 1986 Ericsson-yhtiössä. Alunperin se on ollut suljetussa käytössä, mutta julkaistiin avoimena vuonna 1998. Erlangin tietotyypit ovat dynaamisesti ja vahvasti tyypitettyjä.

Ohjelma, joka tulostaa ensimmäisen syöttölukua suuremman fibonaccin luvun, näyttää Erlangilla seuraavalta. Kommentit alkavat %-merkillä. Numeroidut kommentit, jotka selventävät ohjelman toimintaa, on selitetty koodin alla.

```
1.  -module(fibo).
2.  -export([fibo/1, compfibo/2, fiboplus/1]).
3.       fiboplus(N) -> compfibo(N,0).
4.       compfibo(N,S) -> Curr = fibo(S),
5.            if Curr > N -> Curr;
6.            true -> compfibo(N,S+1)
7.            end.
8.  fibo(0) -> 0 ;
9.  fibo(1) -> 1 ;
10. fibo(N) when N > 1 -> fibo(N-1) + fibo(N-2) .
```

Ohjelman suorittaminen näyttää tältä:

```
1>fibo(0) -> 0 ;
2>fibo(1) -> 1 ;
3>fibo(N) when N > 1 -> fibo(N-1) + fibo(N-2) .
```

Ensimmäisenä tulee määritellä moduulin nimi, tässä tapauksessa 'fibo' rivillä 1. Jotta moduulissa olevia metodeita pystytään kutsumaan ohjelman ulkopuolelta, tulee ne määritellä export-lauseella, kuten rivillä 2. Jokaisen määrityksen yhteydessä ilmoitetaan kutsussa tarvittavien parametrien määrä. Rivillä 3 on varsinainen päämetodi, jota kutsutaan halutulla parametrilla. Parametrina annetaan siis luku, jota suuremman (ensimmäisen) fibonaccin luvun ohjelma laskee. Kyseessä oleva metodi kutsuu ainoastaan apumetodia compfibo, lisäten mukaan apuparametrin, joka pitää kirjaa siitä, monettako fibonaccin lukua kulloinkin haetaan. Rivillä 4 compfibo-metodi tallettaa ensimmäisenä muuttujaan Curr fibonaccin sarjan S:nen luvun. Jos luku on suurempi kuin alkuperäinen annettu luku N, palautetaan se. Muussa tapauksessa kutsutaan compfibo-metodia uudelleen siten, että parametrin S arvoa kasvatetaan yhdellä, jotta seuraavalla kutsukerralla tutkittaisiin järjestyksessä seuraavaa fibonaccin lukua. Rivillä 5 fibo-metodi laskee fibonaccin N:nen luvun. Toimintatapa metodissa on hahmonsovitus, joka on yleinen myös eräässä toisessa ohjelmointikielessä, Haskellissa. Ensimmäisenä tulee määritellä moduulin nimi, joka on tässä tapauksessa 'fibo' rivillä 1. Jotta moduulissa olevia metodeita pystytään kutsumaan ohjelman ulkopuolelta, tulee ne määritellä export-lauseella, kuten rivillä 2. Jokaisen määrityksen yhteydessä ilmoitetaan kutsussa tarvittavien parametrien määrä. Rivillä 3 on varsinainen päämetodi, jota kutsutaan halutulla parametrilla. Parametrina annetaan siis luku, jota suuremman (ensimmäisen) fibonaccin luvun ohjelma laskee. Kyseessä oleva metodi kutsuu ainoastaan apumetodia compfibo, lisäten mukaan apuparametrin, joka pitää kirjaa siitä, monettako fibonaccin lukua kulloinkin haetaan. Rivillä 4 compfibo-metodi tallettaa ensimmäisenä muuttujaan Curr fibonaccin sarjan S:nen luvun. Jos luku on suurempi kuin alkuperäinen annettu luku N, palautetaan se. Muussa tapauksessa kutsutaan compfibo-metodia uudelleen siten, että parametrin S arvoa kasvatetaan yhdellä, jotta seuraavalla kutsukerralla tutkittaisiin järjestyksessä seuraavaa fibonaccin lukua. Rivillä 5 fibo-metodi laskee fibonaccin N:nen luvun. Toimintatapa metodissa on hahmonsovitus, joka on yleinen myös eräässä toisessa ohjelmointikielessä, Haskellissa.

3 Kielten alkiorakenteen vertailu

D-kielen tunnuksissa voi olla kirjaimia, alaviivoja tai universal alphoja (wtf), ja ensimmäistä merkkiä lukuunottamatta myös numerot ovat käytössä. Tunnusta ei voi myöskään aloittaa kahdella alaviivalla. D-kieli on "case sensitive", eli isot ja pienet kirjaimet on eroteltu toisistaan. Varattuja sanoja on

noin sata, esimerkiksi Javastakin löytyvät termit abstract, final, switch ja class.

Merkkijonoliteraali on D:ssä joko merkkijono kaksinkertaisissa lainausmerkeissä, "wysiwyg-merkkijono, ohjausmerkki, rajattu merkkijono, järjestetty merkkijono tai heksamerkkijono (?). Rivin loppumista kuvaava

```
"\n"
```

lasketaan yhdeksi merkiksi kaikissa merkkijonoliteraaleissa.

"Wysiwyg"on lyhennys sanoista "what you see is what you get", ja "wysiwyg-merkkijonoissa kaikki merkit toistetaan uskollisesti. "Wysiwyg-merkkijonot voidaan erotella joko "r- tai „-merkein. Merkkijonot

```
r"c:\root\foo.exe" ja ‘c:\root\foo.exe‘
```

ovat esimerkkejä "wysiwyg-merkkijonoista. Merkkijonot kaksinkertaisissa lainausmerkeissä ovat -merkkien ympäröimiä. Ohjausmerkkejä voi sulauttaa näihin merkkijonoihin monista ohjelmointikielistä tutulla "=merkintätavalla, kuten esimerkiksi merkkijonossa "c:
root
foo.exe". Heksamerkkijonot mahdollistavat merkkijonoliteraalien luomisen heksadatasta. Heksadatan ei tarvitse muodostaa valideja UTF-merkkejä. Valinnainen StringPostfix-merkki antaa merkkijonolle määritellyn tyypin.

Rajoitetut merkkijonot käyttävät useanlaisia erottimia, esimerkiksi merkkejä ja tunnuksia. Erottimen täytyy saumattomasti jatkaa -merkistä ilman välilyöntejä, ja päättävän erottimen täytyy saumattomasti edeltää merkkijonon päättävää -merkkiä ilman välilyöntejä.

Tekstialkiot alkavat merkeillä "q"ja päättyvät merkkiin

". Välissä tulee olla valideja D-kielen tekstialkioita. Merkkiliteraalit ovat yksittäisiä merkkejä tai ohjausmerkkejä, jotka on ympäröity yksinkertaisilla lainausmerkeillä, ’ ’.

Kokonaisluvut voidaan määritellä useissa eri kantalukujärjestelmissä. Kymmenjärjestelmän luvut ovat yksinkertaisesti kymmenjärjestelmän numeroiden sarjoja, mutta binäärijärjestelmän luvut koostuvat binäärijärjestelmän numeroista, joita edeltää merkkojono "0b". Liukuluvut voidaan esittää sekä kymmenluku- että heksadesimaalijärjestelmässä.

```
abstract alias align asm assert auto body
bool break byte
case cast catch cdouble cent cfloat char class const
continue creal dchar debug default delegate delete
deprecated do double else enum export extern false
final finally float for foreach foreach_reverse function
goto idouble if ifloat immutable import in inout int
interface invariant ireal is lazy long macro mixin
module new nothrow null out override package
```

```
pragma private protected public pure real ref
return scope shared short static struct super
switch synchronized template this throw
truetry typedef typeid typeof ubyte ucent uint
ulong union unittest ushort version void
volatile wchar while with __FILE__
 __LINE__ __gshared __traits __vector __parameters
```

- merkkijonoliteraalit: wysiwyg-merkkijonot lainausmerkein ymidyt merkkijonot heksamerkkijonot erotin-merkkijonot token-merkkijonot
- merkkiliteraalit: yksi merkki tai escape-character -merkien sisalla
- kokonaislukuliteraalit mieti miten kasitellaan
- liukulukuliteraalit mieti miten kasitellaan

3.1 Esimerkkejä ohjelmakoodista

... ...

3.2 Ratkaisujen vertailua

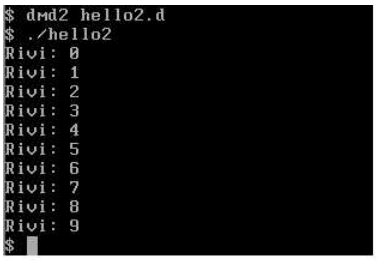
... ...

4 Kielten syntaksi spesifikaatioissa

D-kielen referenssidokumentti löytyy osoitteesta http://dlang.org.
ja tuloste näyttää kuvan 1 mukaiselta.

5 Yhtenveto

Yhteenvetona todettakoon, että D ja Erlang poikkeavat toisistaan...



Kuva 1: Tuloste konsolilla