

# Ohjelmointikielten periaatteet 2013

Jari Koskinen, Hansi Keijonen, Eero Laine

29. tammikuuta 2013

## Sisältö

<b>1</b>	<b>D ohjelmointikieli</b>	<b>3</b>
<b>2</b>	<b>Erlang ohjelmointikieli</b>	<b>3</b>
<b>3</b>	<b>Kielten alkiorakenteen vertailu</b>	<b>4</b>
3.1	Esimerkkejä ohjelmakoodista . . . . .	5
3.2	Ratkaisujen vertailua . . . . .	5
<b>4</b>	<b>Kielten syntaksi</b>	<b>5</b>
4.1	Rakenne . . . . .	5
<b>5</b>	<b>Yhtenveto</b>	<b>6</b>

## 1 D ohjelmointikieli

D on kehitetty korjaamaan C ja C++ kielten puutteita ja samalla siihen on lisätty paljon ominaisuuksia, joiden ansiosta... D kielessä on ominaisuuksia plaaplaa...

- Automaattinen roskienkeruu
- Vahva tyyppitys
- Käännös natiivikoodiksi
- Rinnakkaisuuden tuki

Lisää kielestä plaaplaa

## 2 Erlang ohjelmointikieli

Erlang on funktionaalinen kieli, joka on tarkoitettu alunperin puhelinkeskusten ohjelmointiin. Tästä syystä kieli tukee hyvin rinnakkaisuutta... —  
Ohjelma, joka tulostaa ensimmäisen syöttölukua suuremman fibonaccin luvun, näyttää Erlangilla seuraavalta. Kommentit alkavat %-merkillä. Numeroidut kommentit, jotka selventävät ohjelman toimintaa, on selitetty koodin alla.

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

%1: Ensimmäisenä tulee määritellä moduulin nimi, tässä tapauksessa 'fibo'.  
%2: Jotta moduulissa olevia metodeita pystytään ohjelman ulkopuolella, tulee ne määritellä export-lauseella ulkopuolelta kutsuttaviksi. Jokaisen määrittelyn yhteydessä ilmoitetaan kutsussa tarvittavien parametrien määrä.

%3: Varsinainen päämetodi. Tätä 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.

%4: compfibo-metodi tallettaa ensimmäisenä muuttujaan Curr fibonaccin

sarjan S:n luvun. Jos luku on suurempi kuin alkuperäinen annettu luku N, palautetaan se.

Muussa tapauksessa kutsutaan compfibon-metodia uudelleen siten, että parametrin S arvoa kasvatetaan yhdellä, jotta seuraavalla kutsukerralla tutkittaisiin järjestyksessä seuraavaa fibonaccin lukua.

%5: fibo-metodi laskee fibonaccin N:n luvun. Toimintatapa metodissa on hahmonsovitin, joka on yleinen myös eräässä toisessa ohjelmointikielessä, Haskellissa.

### 3 Kielten alkiorakenteen vertailu

D on vahvasti tyypitetty kieli... Seuraavat tietotyypit ovat tuettuna:

- Int
- Double
- Char
- ...

D : Tunnukset alkavat kirjaimella, '\_'-merkillä tai universal alphalla (jotainhämminkiä tässä, ota selvää!). Seuraavina merkkeinä voi olla mikä tahansa edellisistä ja tunnuksen pituus voi olla mielivaltaisen. Kahdella '\_'-merkillä alkavat tunnukset ovat varattuja.

D: avainsanat ovat varattuja tunnuksia

```
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
true try 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 sisällä
- kokonaislukuliteraalit mieti miten kasitellaan
- liukulukuliteraalit mieti miten kasitellaan

### 3.1 Esimerkkejä ohjelmakoodista

... ..

### 3.2 Ratkaisujen vertailua

... ..

## 4 Kielten syntaksi

### 4.1 Rakenne

Näin tulostetaan Hello World! D-kielellä.

```
import std.stdio;

void main() {
    writeln("Hello World!");
}
```

Vastaavasti Erlangissa Hello World! toteutetaan näin:

```
-module(hello).
-export([hello_world/0]).

hello_world() ->
    io:fwrite("hello, world\n").
```

joka tulostaa konsolille kuvan 1 mukaisesti.

Kuva 1: Tuloste konsolilla

Iteroinnissa C-kielestä tuttu tapa on mahdollinen

```
import std.stdio;

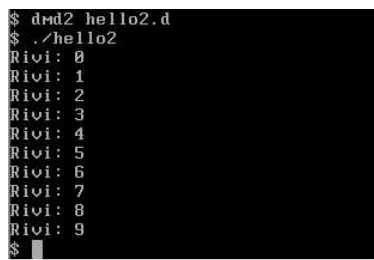
void main() {
    for(int i=0; i<10; i++) {
        writeln("Rivi: ", i);
    }
}
```

Iteraatio voidaan tehdä myös seuraavasti:

```
import std.stdio;

void main() {
    foreach(i; 0 .. 10) {
        writeln("Rivi: ", i);
    }
}
```

ja tuloste näyttää kuvan 2 mukaiselta.



```
$ dmd2 hello2.d
$ ./hello2
Rivi: 0
Rivi: 1
Rivi: 2
Rivi: 3
Rivi: 4
Rivi: 5
Rivi: 6
Rivi: 7
Rivi: 8
Rivi: 9
$
```

Kuva 2: Tuloste konsolilla

## 5 Yhtenveto

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