

合
氣
道

Aikido

Esitlus aines Programmeerimiskeeled

Autor: Gleb Stsenov

Sissejuhatus

合
氣
道

- Aikido on suhteliselt uus programmeerimiskeel, mida saab kasutada kiireks skriptimiskeeleks, prototüüpide kirjutamiseks, ning teiste programmeerimisülesannete lahendamiseks.
- See oli loodud Sun-is, selle autor oli David Allison, ning see ilmus nagu avatud lähtekoodiga tarkvara 2003 septembris.
- Aikido on interpreteeritav objekt-orienteeritud, dünaamiliste tüüpidega keel sisse-ehitatud *multi-threading*-iga, mille süntaks on väga sarnane C++-ga, funktsionaalsus on aga sarnane Pythoni, Perlia, JS ning Javaga

合
氣
道

Nimi ja eesmärgid

- [Aikido](#) on jaapani võistluskunst, mis on suhteliselt uus ning võttis parimad aspektid teistest võistluskunstidest. Nii ka programmeerimiskeel võttis palju teistest programmeerimiskeeltest. Esialgelt nimi oli Darwin, aga Sun ei saanud seda kasutada juriidiliste põhjuste tõttu.
- Eesmärgid:
 - Tutvus. Igaüks, kes on tuttav programmeerimisega C++ pere keeles, peab aru saama Aikido keelest.
 - Evolutsioon. On olemas väga palju kasulikku aspekte teistes keeltes. Aikido võtab parimaid paljudest keeltest alustades [Basicust](#) kuni [Verilogist](#).
 - Kiirus. Muidugi, Aikido kiirust ei saa võrrelda kompileeritavate keelte kiirusega. Aikido eesmärk on olla kiireks oma rühmas.
 - Loetavus. Aikido programm peab olema kergelt loetav inimesena, kes ei ole seda kirjutanud.

Võimalused



- Osaliselt dünaamiliselt tüpiseeritud muutujad:

```
var x = 1  
var pi = 3.14159265  
x = pi // error, cannot change type once defined
```

Kui muutuja on deklareeritud kasutades sõna *generic*, saab selle tüüpi muuta.
- Blokkide pärimine. Saab pärida mitte ainult klassidest, vaid ka mistahes blokkist. N.t. funktsioonid, voogud, monitorid, enum´id.
- Stream operaator. Sisseehitatud operaator -> võimaldab saada andmed voogust programmi sisse ja vastupidi.
- *Native* funktsioonid. Aikido võib kasutada välisteegidest funktsioonid otseteed.

Võimalused (2)

合
氣
道

- Statement macros
- Switch statement comparison operators
- Smart statement boundary detection (;)
- Type casts
- Access control
- Interfaces
- Closures and coroutines
- Polymorphic functions
- Static members

Eeldefineeritud andmetüübid

合
氣
道

- Integer - 64 bit märgiga täisarv
- Real - topeltpäpsusega ujukomaarv
- Character - 8 bit ASCII sümbol
- Byte - 8 bit täisarv
- String - sümbolite järjend
- Vector - mingi tüüpi järjend
- Bytevector - baitide järjend
- Map - sõnastik paaridest value=value
- Object - kasutaja poolt defineeritud tüübi eksemplar
- Stream - I/O kommunikatsioonivoog
- Function - funktsioon
- Thread - funktsioon, mis töötab paralleelses voogus
- Class - kasutaja poolt defineeritud tüüp
- Monitor - kasutaja poolt defineeritud tüüp automaatse *thread-safe* andmetele juurdepääsuga
- Package - kasutaja poolt defineeritud programmi osa
- Interface - no body, no comments :)
- Enumeration - konstantide hulk
- Enumeration constant - konstantide hulka element
- None - tüüp ei ole määratud, null

Massiivid

合
氣
道

- Lihtsamate massiivide asemel on Aikido-s järgnevad konteinerid:
 - List
 - Vector
 - Map
 - Queue
 - Stack
 - Hashtable
 - Container (interface)

合
氣
道

Mäluhaldus

- Reference Counting kasutav *garbage collector*
- *finalize* function

```
class File (name)
var stream = openout (name)
// ...
public function finalize() {
close (stream)
}
}
```


Dynamic Loading

合
氣
道

- **Dynamic expression evaluation**

```
function f {  
  var x = 1 // local variable  
  var y = System.eval ("x + 1")  
  // evaluate  
  // same as var y = x + 1  
}
```

- **Dynamic code loading**

```
var str = System.openin ("code.aikido")  
var p = System.load (str)  
System.close (str)
```

p on klass, mille keha on failis *code*

Konstruktiooni

合
氣
道

- **if** (expression) statement1
- **if** (expression) statement1 **else** statement2
- **if** (expression1) statement1 **elif** (expression2) statement
- **if** (expression1) statement1 **elif** (expression2) statement2 ... **else** statementn
- **switch** (expression) switch-block
- **import** identifier-list
- **import** string-literal
- **using** package-expression
- **while** (expression) statement
- **for** (initializationopt ; expressionopt ; expressionopt) statement
- **do** statement **while** expression
- **foreach** variable **in**_{opt} expression statement
- **foreach** variable **in**_{opt} range-expression statement
- **return** expressionopt
- **break**
- **continue**
- **try** statement **catch** (variable) statement
- **throw** expression
- **delete** expression
- **synchronized** (expression) statement

合 氣 道

Runtime teek

- Aikido *runtime* teek on suhteliselt väike võrreldes teiste keeledega. Palju võimalusi, mis teistel keeledel on lisateekides, on sisseehitatud Aikido interpretaatori sisse. Näiteks, stringidega operatsioonid ei nõua lisateeki kasutamist.
- Paljud teegi funktsioonid ja klassid kasutatakse, et saada juurdepääsu süsteemi funktsioonideks. Näiteks on funktsioonid OSi shelli juurdepääsu saamiseks.
- *Runtime* teek on kättesaadav kasutades ühe või rohkem Aikido *package* failidest impordeerimist. Pärast impordeerimist kõik impordeeritud *runtime* teegi sektsiooni funktsioonid ja klassid saab kasutada programmis.

合
氣
道

Hello, world!

Programmi tekst:

```
println ("Hello world!")
```

Käivitamine:

```
$ aikido hello.aikido
```

Tulemus:

```
Hello world!
```

合
氣
道

Implementatsioon

Aikido on kättesaadav [lähtekoodina](#). See on kirjutatud C++-s, on kompileeritav Linuxis, Mac OS X-is ning Solaris-is. Kahjuks, Windowsi jaoks seda ei ole =(

Küsimused, ettepanekud, kommentaariid?..



合
氣
道

Lingid

- [Aikido Wiki lehekülg](#)
- [Aikido Language System](#) – Aikido SourceForge-is
- [Aikido Programming Reference Manual](#) PDF, 327 l.

Kõik lingid on kontrollitud 28.05.08