Inteligentne budynki

mgr inż. Jerzy Sobczyk

Język programowania Lua

Plan wykładu

- Co to jest język Lua?
- Jakie jest jego zastosowanie?

Co to jest język Lua?

- Lua to potężny, efektywny, lekki, skryptowy jezyk zagnieżdżony.
- "Lua" (wymawiany LUU-ach) oznacza "Księżyc" po portugalsku.
- Jest to jezyk proceduralny, obiektowym funkcjonalny, sterowany danymi.
- Lua jest jezykiem niezawodnym.
- Lua jest szybki.
- Lua jest przenośny.
- Lua moze być wbudowany.
- Lua jest mały.
- Lua jest darmowy.
- Lua jest uzywany w kontrolerach automatyki domowej (Fibaro, Girder, Domoticz, ...).

Zmienne

- Zmienne nie maja typu. Tylko wartości mają typ.
- Nie ma definicji typów.
- Jet 8 typów: *nil*, *boolean*, *number*, *string*, *function*, *userdata*, *thread*, and *table*.
- Są trzy rodzaje zmiennych: zmienne globalne, zmienne lokalne i elementy tablic.

Tablice

- Tablice asocjacyjne to jedyne struktury danych w Lua.
- Zapis a

  $$"name"$$

  jest równowazny z a.name

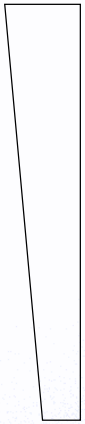
Instrukcje

- Instrukcje można oddzielac instrukcją pustą zapisywana przy pomocy średnika.
- Mozna stosować słowa **do** i **end** do tworzenia instrukcji złożonych.
- **while** wyrażenie **do** blok **end**
- **repeat** blok **until** wyrażenie
- **if** wyrażenie **then** blok **eleseif** wyrażenie **then** blok **else** blok **end**
- **goto** etykieta **::**etykieta**::**
- **break**
- **for** zmienna = wyrażenie **,** wyrażenie **,** wyrażenie **,** wyrażenie **do** blok **end**
- **for** zmienna1, zmienna2, zmienna3 **in** iterator, stan, wartość_początkowa, zakończenie **do** blok **end**

Podstawienia

- Lua zezwala na podstawienia wielokrotne.
- Instrukcja podstawienia najpierw wylicza wszystkiw wyrażenia i dopoero potem wykonywane są podstawienia.
- i = 3
  i, a[i] = i+1, i*2
  daje wynik i = 4 *i* a[3] = 6
- x, y, z = 1, 2, 3
  x, y, z = y, z, x
  daje wynik x = 2, y = 3, z = 1

Operatory



^

unary operators not # – ~

* / // %

\+ –

..

< >

&

~

|

< > <= >= ~= ==

and

or

Funkcje

- **function** *f* () *treść* **end**
  jest równoważne: *f* = **function** () *treść* **end**
- **function** *t.a.b.c.f* () *body* **end**
  jest równoważne: *t.a.b.c.f* = **function** () *treść* **end**
- **local function** *f* () *treść* **end**
  jest równoważne: **local** *f*; *f* = **function** () *treść* **end**
- **function** *f*(*a*, *b*) **end**
  function *g*(*a*, *b*, ...) **end**
  function *r*() bf return *1,2,3* **end**
  f(*r*(), *10*) a=1, b=10
  g(*5*,*r*()) a=5, b=1, ...=2, 3

Obsługa błędów

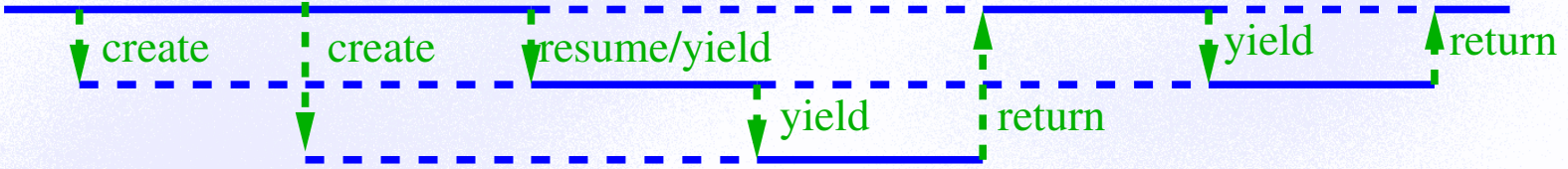
- obsługa błędów działa podobnie do wyjątków.
- **function** *f*(*arg1*, ...) ... **error** (*message*, *level*) ... **return** *res1*, *res2*, ... **end**
- If there were no errors
  true, *res1*, *res2*, ... = **pcall** (*f*, *arg1*, ...)
  true, *res1*, *res2*, ... = **xpcall** (*f*, *handler*, *arg1*, ...)
- If the error has been raised
  false, *errobj* = **pcall** (*f*, *arg1*, ...)
  false, *handler*(*errobj*) = **xpcall** (*f*, *handler*, *arg1*, ...)

Współprogramy a wątki i procesy

	proces	wątek	współprogram
szeregowanie	wywłaszczające	wywłaszczające	współpracujące
przełaczenia	przy zdarzeniach czasowych	przy zdarzeniach czasowych	na rządanie **yield()**
ochrona pamieci	tak	nie	nie
komunikacja	kolejki komunikatów	kolejki komunikatów	kolejki komunikatów
	pamięć wspólna	pamięć wspólna	pamięć wspólna
			wskaźniki/odwołania
synchronizacja	semafory	semafory	nie potrzebna
	sekcje krytyczne	sekcje krytyczne	
czas przełaczania	duży	mały	bardzo mały

Współprogramy

- *thread* = **coroutine.create**(*f*)
  Utworzenie nowego współprogramu.
- *state*, *val1*, ... = **coroutine.resume**(*thread*, *arg1*, ...)
  uruchomienie lub kontynuacja współprogramu.
- *state*, *val1*, ... = **coroutine.yield**(*arg1*, ...)
  Zawieszenie wykonania wywołującego współprogramu.



Metatablice

- Każda wartość w Lua moze mieć metatablicę.
- Metatablica to normalna tablica Lua definiująca zachowania wartości w różnych sytuacjach.
- Operacje sterowane metatablicami:
  - __**add** dodawanie, operacja + .
  - __**sub** odejmowanie, operacja **-**.
  - ...
  - __**len** długość, operacja #.
  - __**eq** porównanie, operacja ==.
  - ...
  - __**call** wywołanie funkcji.

Powtórzenie

- Jakie są zalety języka Lua?
- Do czego jest on stosowany?

Pytania?