

Razvoj programskog jezika Lua

Seminarski rad u okviru kursa
Dizajn programskih jezika
Matematički fakultet

Katarina Branković
e-mail: katarinab70@gmail.com

Jana Vučković
E-mail: jana.vuck@gmail.com

20. decembar 2019.

Apstrakt

U ovom radu ukratko je opisan razvoj programskog jezika Lua, kao i programski jezici koji su najviše uticali na njega: Modula, Scheme, Perl, CLU, C, C++ i Snobol. O svakom jeziku mogu se naći osnovne informacije, ali i njegove sličnosti sa Luom.

Sadržaj

1	Uvod	2
2	Osnovno o Lui	2
2.1	Nastanak jezika	3
3	Razvojno stablo	4
3.1	Modula	5
3.2	Perl	6
3.3	Scheme	6
3.4	C	7
3.5	C++	8
3.6	CLU	8
3.7	SNOBOL	9
4	Zaključak	9
	Literatura	10

1 Uvod

U svetu koji broji preko 2000 programskih jezika postavlja se pitanje da li su nam svi ti jezici neophodni. Ne možemo pobeći utisku da nam svi ti jezici nisu nužni, ali da nam ostavljaju mogućnost izbora da u svakom trenutku upotrebimo jezik koji je po našem ukusu, te postoji više njih koji se fokusiraju na rešavanje istog tipa problema.

Svaki novi jezik pisan je sa ciljem da preuzme dobre karakteristike jezika koji već postoje i pozabavi se eventualnim slabim tačkama već postojećih jezika.

U nastavku sledi kratak osvrt na jedan od retkih jezika koji je nastao van evropskog i severnoameričkog područja koji je našao globalnu primenu - jezik Lua.



Slika 1: Logo programskog jezika Lua

2 Osnovno o Lui

Lua je programski jezik koji spada u više različitih paradigmi - skript paradigmu, objektnoorijentisanu paradigmu i funkcionalnu paradigmu. Nastala je sa osnovnom namenom u programiranju ugrađenih sistema. Samim tim akcenat stavlja na:

- jednostavnost pri ugrađivanju
- prenosivost
- kompaktnost
- brzinu

Iako za skript jezik to nije uobičajeno, Lua spada u efikasne programske jezike i smatra se za jedan od najbržih skript jezika. Samim tim se koristi i u industrijskim aplikacijama, robotici, bioinformatičari, web programiranju, i najbitnije - u razvoju video igara. Danas, ona je jedan od predvodnika u toj industriji.

Još jedna neočekivana stvar kada su skript jezici u pitanju - Lua koristi jako mali broj biblioteka. Primera radi verzija 5.1 iz 2007. godine, ima svega 9 dodatnih biblioteka. Zauzima jako malo memorije, što je veoma bitno s obzirom na to da ima važnu ulogu u programiranju ugrađenih sistema. [8]

2.1 Nastanak jezika

Lua je nastala 1993. godine. Njeni tvorcii su programeri iz Tecgrafa - Roberto Ierusalimschy, Luiz Henrique de Figueiredo i Waldemar Celes [4]. U to vreme, oni su saradivali sa brazilskom firmom Petrobras. Projekat koji je rađen zahtevao je, između ostalog, pripremu velike količine podataka koji su predstavljali ulazne podatke određenih simulacija. Ti podaci su morali biti u striktno određenom formatu, i ta priprema je bila jako naporna i sklona greškama. Kao rešenje tog problema nastao je primitivan deklarativan jezik DEL (Data entry language). DEL je imao jako usku primenu, nije imao kocepte kontrole toka koji imaju moderni programski jezici, ali bio je brz i sposoban da obradi podatke. Kako su zahtevi inženjera iz Petrobrasa rasla, DEL se razvijao, ali jako brzo zahtevi su premašili mogućnosti DEL-a, i nastao je novi programski jezik SOL (Simple object language).

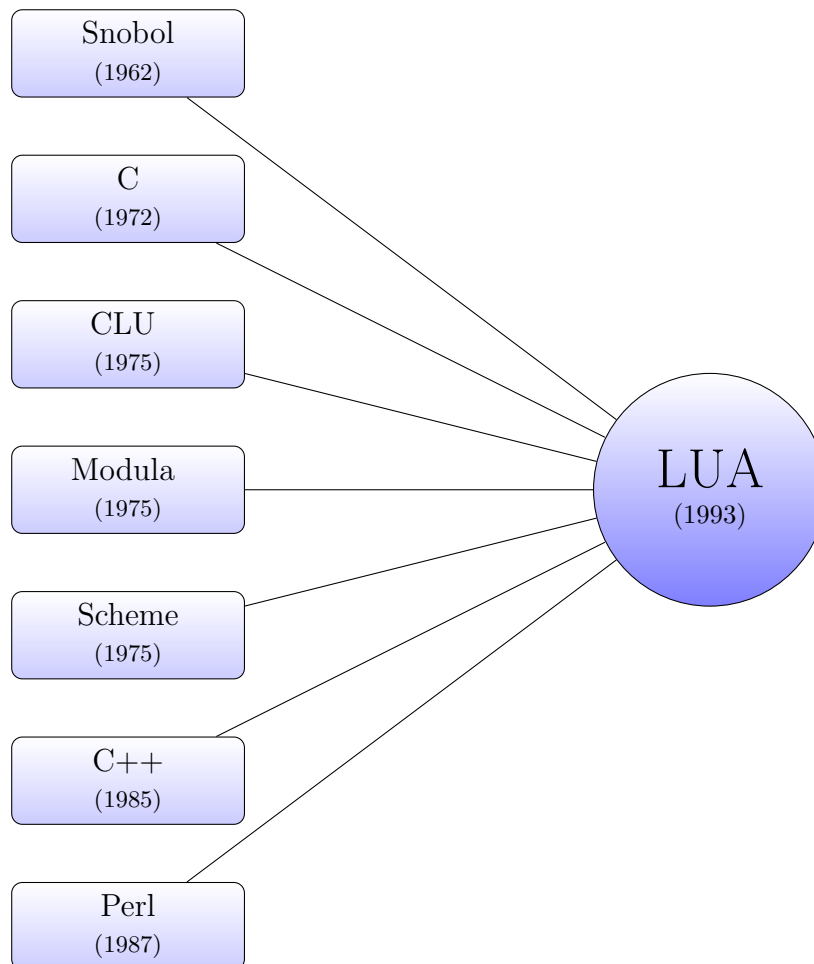
Za razliku od DELa, SOL je posedovao više programerske kocepte, kao što su objekat, kontrola toka i slično. Nastao je u istom periodu kao i DEL, takođe od strane programera iz Tecgrafa, ali na različitom projektu. Inženjeri su odlučili da kombinacijom ta dva jezika naprave novi programski jezik koji su nazvali Lua. Alternativa je bila da se koristi neki od već postojećih skript jezika, na primer Perl, ali ta ideja nije prihvaćena zbog njegove komplikovane sintakse. Motivacija za ime došla je baš od imena Sol. Sol znači sunce na portugalskom, dok Lua na portugalskom znači mesec.

Lua nastaje u periodu kada je Brazil praktikovao strogu razmenu dobara u trenutku kada i u samom Brazilu vladao duh patriotizma da Brazil može i treba da napravi sopstveni softver i hardver. Takođe, u to vreme klijenti Tecgrafa nisu mogli da priušte kupovinu personalizovanog softvera iz inostranstva, pa pravljenje programskoj jezika poput Lue deluje kao sasvim logičan potez. [8]

3 Razvojno stablo

Interesantno, Lua nije nastala u akademske svrhe. Nastala je kao potreba u industriji. Samim tim nije uvela mnogo novih koncepata, a njeni kreatori su pozajmljivali iz drugih programskih jezika ono što su smatrali korisnim. Njen API napisan je u jeziku C, što doprinosi njenoj prenosivosti. Sintaksa Lue najbližnja je sintaksi jezika Modula, što doprinosi njenoj jednostavnosti i minimalističkom duhu. Semantički liči na Scheme, što se kroz novije verzije sve više i više primećivalo. Zajedno sa prethodnim jezicima, značajan uticaj imao je i programski jezik Perl, a pominjemo i C++, CLU i SNOBOL kao jezike od kojih je Lua pozajmila po neku karakteristiku.

Na slici 2 je prikazano osnovno razvojno stablo programskog jezika Lua. Napominjemo da su jezici u stablu sortirani po vremenu nastanka, a ne po značaju koji su imali na nastanak programskog jezika Lua.



Slika 2: Razvojno stablo programskog jezika Lua

3.1 Modula

Modula je viši, proceduralni jezik, napravljen za programiranje računarskih sistema specifičnih namena. Nastala je 1970. godine, a njen tvorac je Niklaus Wirth, koji je takođe i tvorac programskom jezika Pascal. Modula je zato jako slična Pascalu, a najznačajnija novina je uvođenje modula. [9]

Modula ima jednostavnu sintaksu. Lua se po tom aspektu prvenstveno ugleda na nju. Koristi iste ključne reči: while, if, repeat until, end. U verziji 3.1 uvedena je nova mogućnost za iteriranje kroz tabele ¹ - uz pomoć ključne reči foreach. Ključna reč for uvedena je tek u verziji Lua 4.0, dok ključna reč switch nikad nije uvedena. [8]

Dajemo par primera jednostavnih kodova u Lui:

```
Kod 1: Program koji poziva funkciju faktorijal - implementiranu rekurzivno
function faktorijal(n) — definicija funkcije faktorijal
  if n == 0 then
    return 1
  else
    return n * faktorijal(n-1)
  end
end

print("Unesite broj:")
a = io.read("*number")      -- unosimo broj a
print(faktorijal(a))
```

```
Kod 2: Program koji poziva funkciju faktorijal - implementiranu iterativno
function faktorijal(n) — definicija funkcije faktorijal
  local res = 1
  for i in 1..n do
    res = res * i
  end
  return res
end

print("Unesite broj:")
a = io.read("*number")      -- unosimo broj a
print(faktorijal(a))
```

Možemo primetiti da kreatori Lue u duhu njenog minimalističkog duha, nisu hteli da u sintaksu jezika uključe skraćenice poput `res*=res` ili `i++`.

¹Tabele su asocijativni nizovi u Lui, kao rečnici u jeziku Python. Za razliku od nizova tabele se ne indeksiraju samo po brojevima i ne moraju imati unapred zadatu fiksnu dužinu.

3.2 Perl

Jezik Perl je stvoren 1987. godine od strane Larry Wall-a iz SAD-a, lingviste po profesiji koji je radio u Nasi [6]. Skraćenica PERL potiče od Practical Extraction and Report Language koja precizno objašnjava najjače osobine ovog jezika:

- P**ractical za praktičnost tj. brzo pisanje programa
- E**xtraction za izdvajanje delova i analizu podataka iz datoteka
- R**eport za generisanje izlaznih podataka
- L**anguage za programski jezik

Perl je prvenstveno dizajniran za obradu teksta ali je danas veoma rasprostranjen i koristi se i za sistemsko programiranje i razvijanje web aplikacija. Po sintaksi, veoma podseća na programski jezik C:

Kod 3: Ispisivanje elemenata niza u jeziku Perl [6]

```
my @array=(1..10);
for (my $i=0;$i<10;$i++)
{
    print "$array[$i]  ";
}
```

Ono što je sličnost programskog jezika Perl i Lue je jednostavna semantika i postojanje malog broja tipova podataka (u Lui je to osam, a u Perlu samo tri). Podsećamo, Perl je bio jedan od kandidata za projekat zbog kog je Lua nastala, ipak nije doveden u dalje razmatranje zbog svoje komplikovane sintakse.

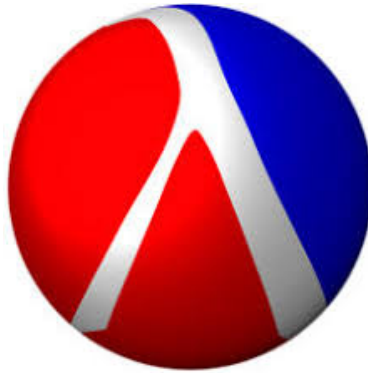


Slika 3: Logo programskog jezika Perl

3.3 Scheme

Scheme je programski jezik opšte namene. Nastao je 1970. godine, a na njega su najviše uticali Algol, proceduralni jezik, i Lisp, jedan od najznačajnijih predstavnika funkcionalne paradigme. Samim tim Scheme pripada i proceduralnoj i funkcionalnoj paradigmi. Zbog svoje kompaktnosti bio je pogodan za proširivanje, što su programeri itekako koristili. Scheme se samim tim, koristio u različite svrhe. Nažalost, to je dovelo do raznih nekompatibilnih kodova koji su u osnovi pisani u istom jeziku.

Lua i Scheme su semantički vrlo slični, iako se na prvi pogled to ne primećuje jer su sintaksički dosta različiti. Baš kao i Scheme, Lua je dinamički tipiziran jezik, što znači da se tipovi promenljivih određuju prilikom izvršavanja, a ne prilikom kompilacije koda. Sledeća osobina koju je Lua preuzela od Scheme-a je da su funkcije objekti prvog reda. Drugim rečima, funkcije se mogu prosleđivati kao argumenti drugih funkcija, mogu biti povratne vrednosti drugih funkcija, a drugim promenljivama mogu biti dodeljene funkcije. [7, 8]



Slika 4: Logo programskog jezika Scheme

3.4 C

Programski jezik C je programski jezik opšte namene koji je 1972. godine razvio Denis Riči u Belovim telefonskim laboratorijama. Ime C dolazi od činjenice da je nastao kao naslednik jezika B (koji je jednostavnija verzija jezika BCPL). C je jezik koji je bio namenjen prevashodno pisanju sistemskog softvera i to u okviru operativnog sistema Unix. Međutim, vremenom je počeo da se koristi i za pisanje aplikativnog softvera na velikom broju drugih platformi.

Jezik C spada u grupu imperativnih, proceduralnih programskih jezika. Kako je nje-gova prvobitna namena bila za sistemsko programiranje, programerima nudi poprilično direktan pristup memoriji i konstrukcije jezika su tako osmišljene da se jednostavno prevode na mašinski jezik. Jezik C je kreiran u minimalističkom duhu - karakteriše ga mali broj ključnih reči dok se dodatna funkcionalnost postiže uglvnom kroz korišćenje bibliotečkih funkcija. [1]

U razvoju jezika C se od samog pocetka insistiralo na standardizaciji i prenosivosti koda. Zahvaljujuci tome isti programi napisani u C-u se mogu kroristiti(tj. prevoditi) na različitim platformama.

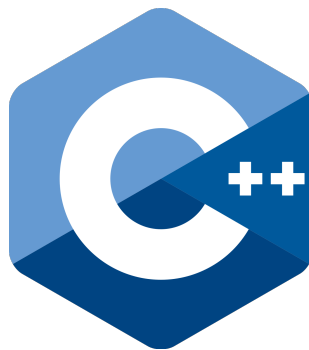
C API je skup funkcija koji dozvoljava programu napisanom u jeziku C da stupi u komunikaciju sa jezikom Lua. Glavna komponenta koja služi za komunikaciju između Lue i C-a predstavlja virtuelni stek. Skoro svi API pozivi se izvršavaju na osnovu vrednosti koja se nalazi na ovom steku. Takođe,sva razmena informacija između Lue i C-a se obavlja preko ovog steka. Pored toga, na njemu se mogu čuvati rezultati

međuizračunavanja.

Upravo je jedna od najboljih karakteristika jezika Lua C API. Ovo posebno dolazi do izražaja ako imamo postojaću C biblioteku koju želimo da sjedinimo sa Luom. [8]

3.5 C++

Pojavom novih tehnika u programiranju, prvenstveno pojavom objektno-orjentisane paradigme osetila se potreba za novim mogućnostima jezika C. Postepenim dodavanjem klasa, provera i konverzije tipova prilikom pozivanja funkcija, preopterećivanja operatora novonastali jezik 1983. godine dobija današnje ime C++. Ime ovog jezika nam naglašava da se ne radi o novom jeziku već proširenju starog (zapravo, ++ je operator povećavanja!) [2]. Preko 95 posto jezika C usvojeno je bez izmena dok se razlike ogledaju u detaljima koji su morali da budu promenjeni radi obezbeđivanja konzistentnosti novouvedenih konceptata.



Slika 5: Logo programskog jezika C++

Jezik C++ pripada proceduralnoj, objektno-orjentisanoj, komponentnoj paradigmi i paradigmi programiranja ograničenja.

Iz jezika C++ Lua je preuzela to što je omogućeno lokalnim promenljivim da budu deklarirane samo tamo gde ih i koristimo što doprinosi čitljivosti koda.

3.6 CLU

CLU je programski jezik koji je kreiran između 1974. i 1975. godina na MIT-u. Njegov kreator je Barbara Liskov i njeni učenici. Iako nije jedan od danas češće korišćenih jezika, zaslužan je za uvođenje (i popularizovanje) mnogih karakteristika koje su veoma korišćene danas. Takođe, smatra se kao značajan u razvijanju objektno-orjentisane paradigme. Njegovi ključni doprinosi uključuju apstraktne tipove podataka, iteratore, mogućnost vraćanja više povratnih vrednosti.

Sintaksa jezika CLU je bazirana na sintaksi jezika ALGOL, sa jednom ključnom razlikom - CLU uvodi i pojam klastera. Klaster odgovara konceptu klase u OOP i ima sličnu sintaksu.

Još jedna bitna osobina CLU jezika je mogućnost paralelnog izvršavanja zadataka. Naime,

$$x, y := y, x$$

vrši zamenu vrednosti promenljivih x i y . Na isti način, funkcija može da vrati i više vrednosti:

$$x, y, z := f(t)$$

Kreatori Lue su upravo ovu osobinu jezika CLU smatrali kao jednostvniju alternativu nego npr. korišćenje pokazivača u C-u i nju su umetnuli u svoj jezik. [3]

3.7 SNOBOL

SNOBOL — StriNg Oriented and symBOLic Language predstavlja porodicu jezika nastalih sredinom 60-ih godina prošlog veka, prvenstveno namenjenih za procesiranje teksta i analizu stringova.

U jeziku SNOBOL se tip podataka promenljivih menja dinamički u toku izvršavanja svakog programskog iskaza.

SNOBOL4 je bio veoma popularan jezik u periodu između 1960. i 1970. godine u učene ja mnogim univerzitetima širom Amerike i takođe je bio dosta korišćen osamdesetih godina prošlog veka. U međuvremenu, njegova popularnost je veoma opala zbog sve više korišćenog jezika Perl.

Ono sto je Lua preuzela od SNOBOLa su asocijativni nizovi (niz koji može biti indeksiran ne samo brojevima, već i stringovima ili bilo kojim drugim vrednostima, osim nil) koji se u Lui nazivaju tabele. Tabele se sastoje od parova ključ-vrednost i koriste se za skladištenje podataka. Tabele su efikasno implementirane, a i lake za upotrebu pa predstavljaju značajnu karakteristiku Lue. [5]

4 Zaključak

Od nastanka, pa sve do danas, kreatori Lue su se držali istog principa - sačuvaj jezik jednostavnim i malim, a implementaciju jednostavnom i prenosivom [8]. Zaista, lepota Lue jeste u njenoj jednostavnosti - jednostavna semantika, jednostavna sintaksa, jednostavna upotreba, jednostavna komunikacija!

Literatura

- [1] Predrag Janičić Filip Marić. *Programiranje 1*. Matematički fakultet univerziteta u Beogradu, 2019.
- [2] Laslo Kraus. *Programski jezik C++ sa rešenim zadacima*. Akademska misao, 2016.
- [3] Barbara Liskov. *CLU Reference Manual*. Massachusetts Institute of Technology, 1979.
- [4] Lua. Zvanični sajt programskog jezika lua. <https://www.lua.org/>.
- [5] Matthew Flatt Anton van Straaten Michael Sperber, R. Kent Dybvig. *Revised⁶ Report on the Algorithmic Language Scheme*. Cambridge University Press.
- [6] Perl. Zvanična stranica programskog jezika Perl. <https://www.perl.org/>.
- [7] I. P. Polonsky R. E. Griswold, J. F. Poage. *The SNOBOL4 Programming Language*. Bell Telephone Laboratories, Incorporated, September 2007.
- [8] Waldemar Celes Roberto Ierusalimschy, Luiz Henrique de Figueiredo. The evolution of lua. 2007.
- [9] Nicolas Wirth. A language for modular programming. 1976.