

Programski jezik Lua

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Prvi autor, drugi autor, treći autor, četvrti autor
kontakt email prvog, drugog, trećeg, četvrtog autora

24. mart 2019

Sažetak

U ovom tekstu je ukratko prikazana osnovna forma seminarskog rada. Obratite pažnju da je pored ove .pdf datoteke, u prilogu i odgovarajuća .tex datoteka, kao i .bib datoteka korišćena za generisanje literature. Na prvoj strani seminarskog rada su naslov, apstrakt i sadržaj, i to sve mora da stane na prvu stranu! Kako bi Vaš seminarski zadovoljio standarde i očekivanja, koristite uputstva i materijale sa predavanja na temu pisanja seminarskih radova. Ovo je samo šablon koji se odnosi na fizički izgled seminarskog rada (šablon koji *morate* da koristite!) kao i par tehničkih pomoćnih uputstava.

Sadržaj

1	Uvod	3
2	Nastanak i istorijski razvoj, mesto u razvojnom stablu, uticaji drugih programskih jezika	3
3	Osnovna namena i mogućnosti programskog jezika Lua	3
4	Osnovne osobine ovog programskog jezika, podržane paradigme i koncepti	4
5	Najpoznatija okruženja (framework) za korišćenje ovog jezika i njihove karakteristike	4
6	Instalacija i uputstvo za pokretanje na Linux/Windows operativnim sistemima	4
7	Primer jednostavnog koda i njegovo objašnjenje	4
8	Sve ono što je specifično i važno za sam taj programski jezik	4
9	Zaključak	4
	Literatura	4
A	Dodatak	5

1 Uvod

Kada budete predavali seminarski rad, imenujete datoteke tako da sadrže redni broj teme, temu seminarskog rada, kao i prezimena članova grupe. Precizna uputstva na temu imenovnja će biti data na formi za predaju seminarskog rada. Predaja seminarskih radova biće isključivo preko veb forme, a NE slanjem mejla. Link na formu će biti dat u okviru obaveštenja na strani kursa. Vodite računa da prilikom predavanja seminarskog rada predate samo one fajlove koji su neophodni za ponovno generisanje pdf datoteke. To znači da pomoćne fajlove, kao što su .log, .out, .blg, .toc, .aux i slično, **ne treba predavati**.

2 Nastanak i istorijski razvoj, mesto u razvojnom stablu, uticaji drugih programskih jezika

3 Osnovna namena i mogućnosti programskog jezika Lua

Iako je Lua prvenstveno razvijena za potrebe dva projekta, danas se, zbog svoje jednostavnosti, efikasnosti i portabilnosti, koristi u najrazličitijim oblastima: ugrađenim sistemima, mobilnim uređajima, veb serverima i igricama.

Lua se obično koristi na jedan od sledeća tri načina: kao skript jezik u sastavu aplikacija pisanih na nekom drugom jeziku, kao samostalan jezik ili zajedno sa C-om.[1]

Ako se Lua upotrebljava u aplikaciji kao ugrađeni jezik, za njeno konfigurisanje u skladu sa domenom date aplikacije potrebno je koristiti Lua-C API. Pomoću njega se mogu, na primer, registrovati nove funkcije, praviti novi tipovi i vršiti izmene u ponašanja nekih operacija. Jedan od primera gde se Lua upotrebljava kao ugrađeni jezik je CGI Lua, alat za pravljenje dinamičkih veb stranica i manipulisanje podacima prikupljenih putem veb formi. Kao ugrađeni jezik, Lua je našla i široku primenu u igricama.

Sve češće, Lua se koristi i kao samostalan jezik, čak i za veće projekte. U te svrhe su razvijene biblioteke koje nude različite funkcionalnosti. Na primer, postoje biblioteke za rad sa stringovima, tabelama, fajlovima, modulima, itd.

Treća mogućnost za upotrebu Lue jeste u okviru programa pisanih u C-u. Lua se tada importuje kao C biblioteka. Programeri koji se opredele za ovakav način rada uglavnom najveći deo programa pišu u C-u, ali moraju dobro da poznaju i Luu kako bi kreirali jednostavne interfejse lake za upotrebu.

Lua je i tzv. jezik za spajanje (eng. glue language). Ona podržava razvoj softvera zasnovan na komponentama. U tom slučaju, aplikacija se kreira spajanjem postojećih komponenti višeg nivoa - komponenti koje su napisane u kompajliranom, statički tipiziranom jeziku kao što je C ili C++. One obično predstavljaju koncepte niskog nivoa koji se neće mnogo menjati tokom razvoja programa jer oduzimaju mnogo procesorskog vremena. Takve komponente se spajaju pomoću Lue. Dakle, Lua se koristi za pisanje onih delova koda koji će se verovatno menjati mnogo puta i na taj način ubrzava proces razvoja programa.

Kao i mnogi drugi jezici, Lua teži tome da bude fleksibilna. Ali, takođe teži tome da bude mali jezik - i u pogledu implementacije i u terminima specifikacije. Za ugrađene jezike ovo je veoma bitna osobina pošto se veoma često koriste u uređajima koji imaju ograničene hardverske resurse. Kako bi se postigla ova dva suprotstavljena cilja, dodavanju novih karakteristika u jezik se pristupa ekonomično. Zbog toga Lua koristi malo mehanizama. A pošto ih je malo, oni moraju biti efikasni. Neki od takvih mehanizama su, na primer, tabele (koje su u suštini asocijativni nizovi), funkcije prvog reda, korutine i refleksivne mogućnosti. Takođe, da bi jezik bio što manji, umesto hijerarhije numeričkih tipova (realni, racionalni, celi), Lua ima samo double floating-point (prevesti?) tip vrednosti.[2]

4 Osnovne osobine ovog programskog jezika, podržane paradigme i koncepti

5 Najpoznatija okruženja (framework) za korišćenje ovog jezika i njihove karakteristike

6 Instalacija i uputstvo za pokretanje na Linux/Windows operativnim sistemima

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

7 Primer jednostavnog koda i njegovo objašnjenje

Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst. Ovde pišem tekst.

8 Sve ono što je specifično i važno za sam taj programski jezik

9 Zaključak

Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak. Ovde pišem zaključak.

Literatura

- [1] Roberto Ierusalimsky. *Programming in Lua*. Lua.org, Rio de Janeiro, 2006.

- [2] Roberto Ierusalimschy. Programming with multiple paradigms in Lua. *Proceedings of the 18th International Workshop on Functional and (Constraint) Logic Programming*, page 5–13, 2009.

A Dodatak

Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe. Ovde pišem dodatne stvari, ukoliko za time ima potrebe.