

Naslov seminarskog rada

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

Tijana Jevtić, Jelena Mrdak, David Dimić, Zorana Gajić
tijanatijanajevtic@gmail.com, mrdakj@gmail.com,
daviddimic@hotmail.com, zokaaa_gajich@bk.ru

6. april 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.

Sadržaj

1 Uvod	2
2 Nastanak i istorijski razvoj	2
3 Osnovna namena, svrha i mogućnosti	3
4 Osnovne osobine	3
5 Okruženja (framework) i njihove karakteristike	3
6 Instalacija i pokretanje	3
7 Primeri kodova sa objašnjenjima	3
8 Specifičnosti	3
9 Zaključak	3
Literatura	3
A Dodatak	3

1 Uvod

Primer 1.1 *I tabele treba da budu u svom okruženju, i na njih je neophodno referisati se u tekstu. Na primer, u tabeli 1 su prikazana različita poravnanja u tabelama.*

Tabela 1: Različita poravnanja u okviru iste tabele ne treba koristiti jer su nepregledna.

centralno poravnanje	levo poravnanje	desno poravnanje
a	b	c
d	e	f

2 Nastanak i istorijski razvoj

1981. godine je oformljena nova laboratorija, Erikson CSLab (eng. *The Ericsson CSLab*) u okviru firme Erikson sa ciljem da predlaže i stvara nove arhitekture, koncepte i strukture za buduće softverske sisteme. Eksperimentisanje sa dodavanjem konkurentnih procesa u programski jezik Prolog je bio jedan od projekata Erikson CSLab-a i predstavlja začetak novog programskog jezika. Taj programski jezik je 1987. godine nazvan Erlang¹. Sve do 1990., Erlang se mogao posmatrati kao dijalekt Prologa. Od tada, Erlang ima svoju sintaksu i postoji kao potpuno samostalan programski jezik. Godine rada su rezultirale u sve bržim, boljim i stabilnijim verzijama jezika, kao i u nastanku standardne biblioteke OTP (eng. *The Open Telecom Platform*) [2]. Od decembra 1998. godine, Erlang i OTP su postali deo slobodnog softvera (eng. *open source software*) i mogu se slobodno preuzeti sa Erlangovog zvaničnog sajta [4]. Danas, veliki broj kompanija koristi Erlang u razvoju svojih softverskih rešenja. Neke od njih su: Erikson, Motorola, Votsap (eng. *Whatsapp*), Jahu (eng. *Yahoo!*), Amazon, Fejsbuk (eng. *Facebook*).

2.1 Mesto u razvojnom stablu

2.2 Uticaji drugih programskih jezika

Erlang je funkcionalan i konkurentan programski jezik. Na njega, kao na funkcionalan jezik, uticao je Lisp funkcionalnom paradigmom koju je prvi predstavio. Na planu konkurentnosti Erlang svojevrsan primer (detaljnije u odeljku 4).

Na početku, Erlang je stvaran kao neki dodatak na Prolog, vremenom prerastao u dijalekt Prologa, a kasnije je zbog svoje kompleksnosti i sveobuhvatnosti evoluirao u potpuno novi programski jezik. Stoga je uticaj Prologa na Erlang bio neminovan. Sintaksa Erlanga u velikoj meri podseća na Prologovu (npr. promenljive moraju počinjati velikim slovom u oba jezika, svaka funkcionalna celina se završava tačkom), oba jezika u velikoj meri koriste poklapanje obrazaca (eng. *pattern matching*).

Sa druge strane, Erlang je uticao na nastanak programskog jezika Elixir (eng. *Elixir*).

¹Erlang je jedinica saobraćaja u oblasti telekomunikacija i predstavlja kontinuirano korišćenje jednog kanala (npr. ako jedna osoba obavi jedan poziv telefonom u trajanju od sat vremena, tada se kaže da sistem ima 1 Erlang saobraćaja na tom kanalu).

3 Osnovna namena, svrha i mogućnosti

4 Osnovne osobine

4.1 Podržane paradigme

4.2 Koncepti

5 Okruženja (framework) i njihove karakteristike

6 Instalacija i pokretanje

Postoji više načina da se instalira Erlang sa neophodnim paketima. U ovom odeljku će biti predstavljena instalacija korišćenjem prekompajliranih binarnih fajlova za neke operativne sisteme zasnovane na Linuksovom kernelu i pokretanje na jednom od njih, kao i instalacija za Windows.

6.1 Linux

Nakon uspešne instalacije, Erlang kod je moguće kompajlovati ili interpretirati i pokretati u interpretatoru. Interpretator se pokreće kucanjem komande *erl* u terminalu, a iz istog se izlazi sa *Ctrl+G* iza kog sledi *q* [1]. Erlang interpretator ima u sebi ugrađen editor teksta koji je baziran na *emacs-u* [3].

Kod iz datoteke se kompajluje komandom *erlc* i navođenjem imena fajla sa ekstenzijom *erl* nakon iste. Nakon toga se dobija izvršna datoteka sa ekstenzijom *beam* koja se može pokrenuti uz navođenje adekvatnih flegova.

6.2 Windows

7 Primeri kodova sa objašnjenjima

8 Specifičnosti

9 Zaključak

Literatura

- [1] J. Armstrong. *Programming Erlang (2nd edition)*. Pragmatic Bookshelf, 2013.
- [2] Joe Armstrong. *Making reliable distributed systems in the presence of software errors*. PhD thesis, The Royal Institute of Technology, Stockholm, Sweden, Decembar 2003.
- [3] F. Hebert. *Learn You Some Erlang for Great Good!* No Starch Press, 2013.
- [4] OTP team. Erlang. on-line at: <http://www.erlang.org/>.

A Dodatak