

# Razlika između programskih jezika C i C++

Seminarski rad u okviru kursa  
Tehničko i naučno pisanje  
Matematički fakultet

Ime i prezime autora  
kontakt email adresa autora

24. oktobar 2017.

## 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 ispoštujete!) kao i par tehničkih pomoćnih uputstava.

## Sadržaj

# 1 Uvod

Uvod

## 2 O jeziku C

Programski jezik C je programski jezik opšte namene koji je 1972. godine razvio Denis Riči<sup>1</sup>. C je jezik koji je bio namenjen prevashodno pisanju sistemskog softvera i to u okviru operativnog sistema Unix. C je danas prisutan na širokom spektru platformi – od mikrokontrolera do superračunara.

Jezik C spada u grupu imperativnih, proceduralnih programskih jezika. Kako je izvorno bio namenjen za sistemsko programiranje, programerima nudi prilično direktan pristup memoriji i konstrukcije jezika su tako osmišljene da se jednostavno prevodi na mašinski jezik. Jezik je kreiran u minimalističkom duhu – ima mali broj ključnih reči, a dodatna funkcionalnost programerima se nudi uglavnom kroz korišćenje bibliotečkih funkcija.

## 3 Nedostaci jezika C

Uprkos svojoj brzini i efikasnosti programski jezik C ima izvestan broj nedostataka, koji je možda najbolje opisan citatom Denisa Ričija, čoveka koji je napravio C: „C ima moć asemblerskih jezika i koristi se lako kao asemblerski jezici.” Asemblerski jezici su ozloglašeni po težini korišćenja i Denis Riči je ovime hteo da ukaže da je C takođe komplikovan i da ne pruža puno olakšica programerima koji ga koriste.

Neki od poznatijih zamerki na C su: način na koji se dinamički alocira memorija koristeći *malloc* i *calloc* koji može dovesti do curenja memorije ukoliko se ona ne oslobađa pravilno ili do pristupa delu memorije koji je već oslobođen što bi takođe dovelo do greske. Način na koji funkcionišu pokazivači takođe zna da bude problematičan i njihovo netačno korišćenje od strane programera može dovesti do korupcije memorije. Moguće je i da više pokazivača sadrže istu adresu i time program ne bude maksimalno optimizovan, kao što je slučaj u nekim drugim programskim jezicima. Kompajleri nisu od velike pomoći programeru i moguće je napraviti greške koje se mogu detektovati u fazi kompilacije, ali za to kompilator nije sposoban. Niske se čuvaju kao niz podataka tipa *char* sa terminirajućom nulom na kraju i njihovo korišćenje i obrada su veoma komplikovani i zahtevaju manipulaciju memorijom od strane programera. Ne postoji ništa što sprečava programera da piše neodrživ kod i istorijski se to dešavalo toliko često da postoje i takmičenja za pisanje što komplikovanijeg koda u programskom jeziku C. Objektno orijentisano programiranje takođe nije podržano u C-u kao i mnoge druge funkcionalnosti koje olakšavaju pisanje i održavanje koda (npr. introspekcija tipa).

## 4 Razlike između jezika C i C++

Najznačajniji direktni naslednik jezika C je jezik C++ koji se, u trenutku nastanka, mogao smatrati njegovim objektno-orijentisanim proširenjem.

---

<sup>1</sup>Dennis Ritchie (1941–2011), američki informatičar

Kreirao ga je Bjern Stroustrup<sup>2</sup> 1986. godine. C++ je i dalje jedan od najpopularnijih jezika i koristi se za razvoj zahtevnih aplikacija, s jedne strane zbog svojih objektno-orijentisanih svojstava, a s druge zbog bliske veze sa mašinom, u duhu jezika C.

Neka od svojstava jezika C++ koja ne postoje u jeziku C su

- mnoštvo objektnih tipova kao što su *vector*, *set*, *stack*, *queue*, implementiranih u standardnoj biblioteci
- mogućnost prenosa argumenata funkcije i po referenci
- range for petlje

Osobina	C	C++
Programska paradigma	Imperativna	Objektno-orijentisana
Osnova	Asembler	C programski jezik
Petprocesorske direktive	?	?
Pristup u programiranju	S vrha - nanže	S dna - naviše
Alokacija memorije	Korišćenjem bibliotечkih funkcija	Korišćenjem ključnih reči
Prenos argumenata funkcije	Po vrednosti	Po vrednosti, moguće i po referenci
Pokazivači	podržani	podržani
Upravljanje izuzecima	nije podržano	podržano

## 5 Zaključak

Zaključak

## Literatura

- [1] J. Laski and W. Stanley. *Software Verification and Analysis*. Springer-Verlag, London, 2009.
- [2] Free Software Foundation. GNU gcc, 2013. on-line at: <http://gcc.gnu.org/>.
- [3] A. M. Turing. *On Computable Numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, 2(42):230–265, 1936.

## 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.

---

<sup>2</sup>Bjarne Stroustrup (1950), danski informatičar