

# Razvoj programskog jezika D

Seminarski rad u okviru kursa  
Dizajn programskih jezika  
Matematički fakultet

Dunja Čitlučanin i Nina Matić  
dunjac98@gmail.com, nina.matic121@gmail.com

December 2019

## Sažetak

Tema ovog rada je razvoj programskog jezika D. Opisani su jezici koji su imali uticaja na njegov nastanak: C, C++, Eiffel, Python, Java i C, kao i jezici na koje je uticao sam jezik D: Genie, MiniD, Qore, Swift i Vala. U radu se nalazi i razvojno stablo koje oslikava veze između pomenutih jezika.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Osnovno o jeziku D</b>	<b>2</b>
<b>3</b>	<b>Razvojno stablo</b>	<b>3</b>
3.1	C . . . . .	3
3.2	C++ . . . . .	4
3.3	Eiffel . . . . .	5
3.4	Python . . . . .	5
3.5	Java . . . . .	6
3.6	C . . . . .	6
3.7	MiniD . . . . .	7
3.8	Vala . . . . .	7
3.9	Genie . . . . .	7
3.10	Swift . . . . .	8
3.11	Qore . . . . .	8
<b>4</b>	<b>Zaključak</b>	<b>8</b>
	<b>Literatura</b>	<b>9</b>

# 1 Uvod

Ne postoji osoba, niti program koji bi mogli dati odgovor na pitanje koliko programskih jezika danas postoji. Taj broj je svakoga dana u porastu, a uticaji drugih jezika na novonastali su vrlo česti. Kako bismo bolje razumeli motivaciju nastanka nekog jezika, njegove važne karakteristike, kao i potencijalna unapredjenja u odnosu na njegove prethodnike, vrlo je važno da se upoznamo i sa jezicima koji su imali uticaja na njegov nastanak. Više različitih jezika uticalo je na jezik D, što je rezultiralo time da dobijemo jezik koji objedinjuje različite paradigme, zadržava neka svojstva ovih jezika, ali se ipak razlikuje od njih. Više o tome u nastavku. Logo jezika D prikazan je na slici 1.



Slika 1: Logo programskog jezika D

## 2 Osnovno o jeziku D

**Programski jezik D**, poznat kao **Dlang**, podržava više paradigmi [3]. Nastao je 2001. na Digital Mars-u a njegov tvorac je Walter Bright. Osim njega, na jeziku je od 2007. radio i Andrei Alexandrescu. Iako je prvenstveno nastao kao redizajn programskog jezika C++, D se, ipak, dosta razlikuje od njega. Ima sličnosti i sa jezicima Java, Eiffel, Ruby, C i Python.

Prilikom dizajniranja ideja je bila da programski jezik D objedini karakteristike kompajliranih jezika, performanse i sigurnost, sa ekspresivnom snagom savremenih dinamičkih jezika. D nudi mogućnost generisanja efikasnijeg koda istovremeno obezbeđujući manju verovatnoću programskih grešaka. Iako jezik koristi više koncepata C i C++, neke odbacuje i nije kompatibilan sa izvornim kodom u ovim jezicima. D dodaje funkcionalnost jeziku C++ tako što implementira sakupljače smeća, dinamičke matrice, ugnježdene funkcije, a što se tiče deklaracija i izraza usko se podudara sa jezikom C++. Takođe, D ima ugrađenu podršku za komentare u dokumentaciji, što omogućava automatsko generisanje dokumentacije.

D podržava sledeće programske paradigme: imperativnu, objektno-orijentisanu, kao i funkcionalnu.

Imperativno programiranje u jeziku D je gotovo identično kao u jeziku C. Funkcije, podaci, deklaracije i izrazi funkcionišu isto kao i u C-u, dok,

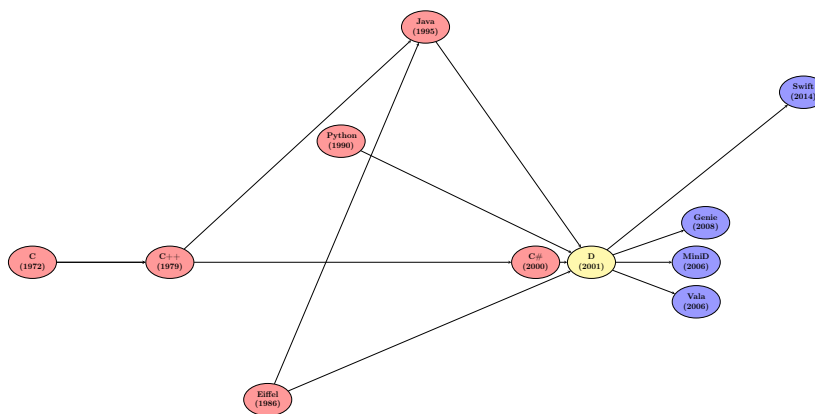
s druge strane, imamo neznatne razlike, kao što je na primer mogućnost konstrukcije D-ove petlje koja omogućava petlju preko kolekcije [12].

Objektno-orientisano programiranje u programskom jeziku D temelji se na hijerarhiji nasleđivanja. Višestruko nasleđivanje nije podržano, ali se to kompenzuje korišćenjem interfejsa, slično kao i u Javi.

Memorijom se obično upravlja korišćenjem sakupljača otpadaka. Sakupljanje smeća se može kontrolisati na više načina, programeri mogu omogućiti i onemogućiti prikupljanje i mogu dodavati i isključivati memorijske opsege koje posmatra sakupljač otpadaka.[17]

### 3 Razvojno stablo

Najveći uticaj na nastanak **programskog jezika D** imao je **C++**, upravo zato što je D prvenstveno zamišljen kao redizajn ovog jezika, a kako je veliki uticaj na jezik C++ imao **C**, neki koncepti preuzeti su iz njega, na primer, imperativno programiranje je gotovo identično kao i u programskom jeziku C. Veliki uticaj na D imali su i programski jezici **Eiffel**, **Python**, **Java**, kao i **C** sa kojima ima dosta sličnosti, ali i značajnih razlika. Sa druge strane, jezici na koje je uticao jezik D su **MiniD**, **Genie**, **Vala**, **Qore** i **Swift**. Razvojno stablo programskog jezika D može se videti na slici 2. Roze bojom su predstavljeni jezici koji su uticali na njegov nastanak, žutom je predstavljen sam jezik D, a ljubičastom jezici na koje je imao uticaj.



Slika 2: Razvojno stablo programskog jezika D

#### 3.1 C

**Programski jezik C** je proceduralan strukturirani jezik kompajlerskog tipa koji kombinuje efikasnost sličnu assembleru sa lakoćom rada svojstvenom jezicima visokog nivoa. Razvijen je u ranim 70-godinama dvadesetog veka. Dizajnirali su ga tvorci operativnog sistema Unix. Dennis

Ritchie je bio glavni u kreiranju ovog jezika, a veliki doprinos u nastanku imali su i Ken Thompson i Martin Richards. Ritchie je stvorio C za rešavanje praktičnih problema kodiranja sistemskih programa i jezgra operativnog sistema Unix, koji je inicijalno bio napisan u assembleru, ali onda je 1972. ceo reimplementiran u C-u. Tokom godina C se dosta menjao pa zbog toga ima više formalnih standarda. [2]

Kao jedan od najvažnijih jezika u istoriji komercijalne računarske industrije, C je do danas ostao jedini programski jezik prilagođen za sve računarske platforme, od malih sistema pa do mrežnih superračunara. Programi napisani u njemu vrlo su bliski načinu rada hardvera te zahtevaju od programera dobro razumevanje rada procesora, memorije, ulazno-izlaznih sklopova... Bjarne Stroustrup u saradnji sa drugim istraživačima u Bell Labs-u vrši nadogradnju jezika C dodajući mu objektno-orijentisana svojstva, čime je stvoren novi programski jezik, C++. Efikasno i sigurno programiranje u C-u je vrlo zahtevno pa se ne preporučuje početnicima. Jedan od problema je, na primer, upravljanje memorijom koju programer mora eksplicitno kodirati. Glavno područje primene su sistemski programi na strani poslužitelja(eng.servers), programi prevodioci(eng. compilers) i jezgra operativnih sistema(eng. Operating systems kernels), gde su potreba za najvećom mogućom brzinom izvođenja, efikasnom kontrolom resursa i kontrola hardvera od primarne važnosti.

### 3.2 C++

C++ je stvorio Bjarne Stroustrup, danski informatičar, 1979. godine. Formirao ga je kombinujući karakteristike jezika Simula sa jezikom C, a uticaj su imali i jezici Ada, ML kao i Algol 68 [7]. Prvobitno je kreirao "C sa klasama", a na njega je dodavao mnogo toga, poput konstanti, referenci, preopterećenja imena funkcija i operatora, poboljšanu proveru tipova podataka i jednodredne komentare. Vremenom se dosta menjao, menjajući i imena, do konačnog naziva 1982. godine, pod kojim ga danas znamo. Savremeni C++ ima objektno-orijentisane, generičke i funkcionalne karakteristike pored mogućnosti manipulacije memorije niskog nivoa. Gotovo uvek se implementira kao kompajlerski jezik. Godine 1985. objavljeno je prvo izdanje Programskog jezika C++, koje je postalo definitivna referenca za jezik, jer još nije postojao zvanični standard. Prva komercijalna implementacija C++ objavljena je u oktobru iste godine. Vremenom je dodat novi tip podataka bool, uvedena je mogućnost višestrukog nasleđivanja, hvatanje izuzetaka, apstraktne klase... Zbog svoje brzine C++ se čini idealnim izborom za razvoj operativnih sistema, bilo da je to Microsoft Windows, Mac OSX ili Linux – svi su programirani u C++. Takodje ga koriste i brojni pregledači jer onda korisnici ne moraju dugo da čekaju da se sadržaj pojavi na ekranu. Tensorflow, jedna od najčešće korišćenih biblioteka za mašinsko učenje koristi C++ kao pomoćni programski jezik. Danas C++ spada među najkorišćenije jezike među programerima. Ima bogatu standardnu biblioteku sa bogatim nizom funkcija koje manipulišu datotekama i metodama koje manipulišu strukturama podataka itd. Imao je velikog uticaja na jezike Java, Python, Perl, Php, Rust...

### 3.3 Eiffel

**Eiffel** je dobio ime po Gustave Eiffel, inženjeru koji je stvorio čuvenu kulu. 1985. godine je osmišljen u kompaniji Eiffel Software (tada poznata kao ISE) [6]. U početku je služio kao interno sredstvo za razvoj različitih unutrašnjih aplikacija. Eiffel-ovim inženjerima softvera bilo je potrebno snažno objektno-orijentisano okruženje koje je integrisalo koncepte modernog softverskog inženjeringa, a ništa slično im nije bilo na raspolaganju. Zbog tog je dr Bertrand Meyer za njih dizajnirao kompajler The Eiffel 1. Javnosti je predstavljen na prvoj konferenciji OOPSLA u oktobru 1986. godine gde je privukao veliku pažnju, što je dovelo do toga da krajem 1986. godine bude objavljen kao komercijalni proizvod. Tokom narednih godina tehnologija se brzo širila, što je dovelo do mnogo uspešnih industrijskih projekata u SAD-u, Kanadi, Evropi i Dalekom istoku. Eiffel je već od samog početka privukao pažnju akademske zajednice kao idealno sredstvo za učenje programiranja na svim nivoima. Od tada ga brojni univerziteti širom sveta koriste kao primarni jezik predavanja. Otprilike jednom godišnje pojavljivale su se nove verzije. Popularnost je porasla objavljivanjem knjige Objektno-orijentisane softverske konstrukcije od strane dr Bertranda Meyera, 1988. godine. Ta knjiga je ubrzo postala jedan od najprodavanijih naslova i prevedena je na osam jezika. Objavljeno je i prošireno, novo izdanje ove knjige. Tehnologija Eiffel-a i dalje pomerena granice razvoja softvera. Ključna karakteristika je objektno-orijentisana programska struktura u kojoj klasa služi kao osnovna jedinica dekompozicija. Posедуje automatsko upravljanje memorijom, koje se obično vrši sakupljačem otpadaka. Podržava višestruko nasleđivanje, polimorfizam, enkapsulaciju i sigurnu konverziju tipova podataka. Ima podršku za ograničeno i nesputano generičko programiranje. Sintaksa je zasnovana na ključnim rečima u Algol/Pascal tradiciji. Cilj nije samo učiniti kod čitljivijim, već i omogućiti programerima da se koncentrišu na važne aspekte programa, a da se ne zamaraју detaljima implementacije. Takođe, cilj dizajna koji stoji iza Eiffel jezika, biblioteka i metoda programiranja je omogućiti programerima da stvore pouzdane softverske module koji se opet mogu koristiti. Nekoliko drugih programskih jezika sadrži elemente koji su prvi put predstavljeni u jeziku Eiffel, neki od njih su Sather i BlueJ.

### 3.4 Python

**Python** je široko korišćen programski jezik opšte namene [13]. Interpretatorskog je tipa dizajniran da podržava više različitih programskih paradigmi uključujući funkcionalno, proceduralno i objektno-orijentisano programiranje. Prvobitno je osmišljen 1989. godine kao naslednik programskog jezika ABC za distribuirani operativni sistem Amoeba. Dizajnirao ga je Guida Van Rossum, a razvio Python Software Foundation. On je bio zaposlen u kompaniji Centrum Wiskunde Informatica (CWI) sa sedištem u Holandiji. Van Rossum je projekat započeo iz hobija, tako što je tražio zanimljiv projekat koji bi ga držao okupiranim tokom Božića. [11] Razvijen je radi naglašavanja čitljivosti koda, a njegova sintaksa omogućava programerima da izraze koncept u manje linija koda. Inspiraciju za ime dobio je iz BBC-ijeve

emisije „Leteći cirkus Monti Python-a“, pošto je Guida Van Rossum bio veliki obožavalac TV emisije i hteo je kratko, jedinstveno i pomalo misteriozno ime za svoj izum. Jezik je objavljen 1990. godine. Kada je objavljen koristio je mnogo manje kodove u odnosu na recimo Javu, C++ i C, njegova filozofija dizajna je bila prilično dobra. Glavni cilj je bio da se obezbedi čitljivost koda. Kada je počeo da se upotrebljava omogućavao je korišćenje klasa za nasleđivanje, nekoliko osnovnih podataka za obradu izuzetaka i funkcije. Dinamički je tipiziran jezik i poseduje sistem za prikupljanje smeća.

### 3.5 Java

U junu 1991.godine James Gosling, Mike Sheridan i Patrick Naughton započeli su rad na jeziku **Java** [9]. Java je prvobitno bila dizajnirana kao projekat firme Sun Microsystems za interaktivnu televiziju, ali je u to vreme bila previše napredna za industriju digitalne kablovske televizije. Prvobitni naziv je bio Oak(hrast), po hrastu koji je stajao ispred kancelarije Goslinga i jezik je bio namenjen za programiranje kućnih elektronskih uređaja(televizora, video rekordera). Kasnije je dobio naziv Green, da bi došao do finalnog naziva, Java, po kafi iz Indonezije[10]. Prva javna implementacija, Java 1.0 objavljena je 1995. godine od strane Sun Microsystems-a. Glavni veb pregledači ubrzo su ugradili mogućnost pokretanja Java apleta na veb stranicama i Java je ubrzo postala popularna. Većina Java implementacija bila je dostupna bez naplate. Osnovni ciljevi pri stvaranju Java jezika bili su da jezik mora biti jednostavan- da se u njemu lako programira, objektno-orijentisan, arhitektonski neutralan i prenosiv, mora se izvršavati sa visokim performansama. Java je jezik koji se izvršava korišćenjem specijalne virtuelne mašine, Java virtuelne mašine-JVM.To omogućava da se izvršni kod ne mora kompajlirati zasebno za pojedinačne arhitekture već se isti izvršni fajl može koristiti na bilo kom računaru koji ima podršku za Javu. JVM je apstraktna mašina koja postoji samo u memoriji. Programski jezik Java je jezik koji se prevodi i interpretira. Izvorni kod se nakon kompajliranja prevodi u izvorni kod, bajtkod. Ovaj bajtkod se dešifruje u Java virtuelnoj mašini i interpretira u izvršni kod pogodan za izvršavanje na računaru. U zavisnosti od virtuelne mašine, za isti izvorni kod možemo dobiti različite izvršne kodove, za različite platforme.

### 3.6 C

U januaru 1999. godine Anders Hejlsberg započeo je projekat kreiranja programskog jezika koji je trebao biti objektno-orijentisan i sličan jeziku C. Prvobitni naziv bio je **Cool**. U julu 2000. godine jezik je preimenovan u **C#**, inspirisano muzičkom notacijom gde šarp označava da se napisana nota izvodi za pola koraka više. Ovaj jezik je naslednik jezika C i C++. Hajsberg je više puta isticao da su upravo nedostaci drugih programskih jezika, kao što su Java, C++ i Delphi doveli do stvaranja programskog jezika C. U početku, postojale su optužbe da je C u velikoj meri sličan jeziku Java, međutim, od 2005. godine ova dva jezika počinju da se razvijaju u različitim pravcima. Jedna od najvažnijih razlika jeste dodavanje generika

u oba jezika i to sa potpuno različitom implementacijom. Takođe, C se zahvaljujući nekim karakteristikama po kojima se izdvajao od drugih programskih jezika približio funkcionalnom stilu. Neke od tih karakteristika su podržavanje lambda izraza, anonimni tipovi i metode proširenja, što omogućava programerima da koriste tehnike funkcionalnog programiranja kad god se naiđe na potrebu za tim.

Programski jezik C nije sintaksno složen, ali je vrlo izražajan kada je potrebno rešiti neki problem softverskog razvojnog procesa. Podržava direktan pristup memoriji pomoću pokazivača, upozoravače sakupljača smeća koji sprečavaju uništavanje objekata na koje pokazuju pokazivači dok se ti objekti ne oslobode, kao i rezervisane reči za izdvajanje nesigurnih operacija.[8] Ne postoje ograničenja u pogledu toga kakve aplikacije je moguće napraviti korišćenjem jezika C, a najčešći tipovi pravljenih aplikacija su Veb aplikacije( veb strane koje možemo videti kroz bilo koji čitač), Veb servisi i Windows aplikacije.

### 3.7 MiniD

Nastanak programskog jezika **MiniD** počinje u junu 2006.godine, kao ideja za jezik statičkog tipa, kao i jedna od verzija jezika D, otuda i naziv. Tvorač je Jarrett Billingsle. On je prilikom stvaranja kompajlera za ovaj jezik shvatio koliko to veliki projekat postaje i odlučio da jezik preoblikuje u nešto jednostavnije za implementaciju. Rezultat je bio jezik u obliku jezika Lua sa sintaksom u stilu jezika C. Tokom narednih meseci MiniD uveo funkcije iz različitih jezika. Tokom godina nastajale su nove verzije, dok razvoj konačno nije zaustavljen u junu 2011. godine. Korišćen je i kao kao osnova za novi jezik koji je isti autor nazvao Croc. MiniD pruža mali ali fleksibilan skup tipova podataka, sličan kao kod jezika Lua ili Squirrel. Za razliku od Lua, MiniD pruža eksplicitnu podršku za objektno-orijentisano programiranje sa klasama.[5] Takođe ima sakupljač smeća(eng. Garbage collector) i pokušava biti robusniji od tipičnih dinamičkih jezika, čineći hvatanje grešaka bržim. Još jedna korisna karakteristika su “ograničenja tipa parametara”, koji su način odredjivanja važećih tipova koje parametri funkcija mogu prihvatiti.

### 3.8 Vala

Programski jezik **Vala** se prvi put pojavio 2006. godine [16]. Valu je osmislio Jurg Billeter, a realizovao ga je sa Raffaelom Sandrinijem. Pretežno je objektno-orijentisan ali podržava i imperativno i strukturalno programiranje. Sintaksički je sličan C i uključuje značajne funkcije kao sto su anonimne funkcije, signale, generike, rukovanje izuzecima kao i zaključivanje tipa.

### 3.9 Genie

Razvoj programskog jezika **Genie** počinje od 2008. godine. On je moderan, jezik opšte namene.[4] Sinatksa je izvedena iz brojnih savremenih jezika kao što su Python, Boo, D i Delphi. Kao i programski jezik Vala koristi sistem tipa GObject za kreiranje klasa i interfejsa deklarisanih u

Genie izvornom kodu, bez nametanja dodatnih uslova rada, tj. za razliku od Python, Java ili C ne zahteva virtuelnu mašinu.

### 3.10 Swift

Razvoj ovog jezika započet je u julu 2010.godine od strane Chris Lattner-a, uz saradnju mnogo programera zaposlenih u Applu [15]. Cilj Swift projekta je da stvori najbolji raspoloživi jezik za upotrebu u rasponu od programiranja sistema, preko aplikacija za mobilne uređaje i desktop računare, sve do cloud usluga. Ono što je najvažnije, Swift je dizajniran da programeru olakša pisanje i održavanje ispravnih programa. Zvanična verzija objavljena je 2014. godine.[1] Da bi se postigao cilj, očigledno je da pisanje Swift koda mora biti sigurno, ne želimo nedefinisano ponašanje, zatim brz- zamišljen je kao zamena za jezike zasnovane na C-u, kao i ekspresivan. Swift je kompajlerski jezik opšte namene koji podržava više paradigmi. Ima karakteristike objektno-orijentisane, funkcionalne i imperativne paradigme. Prilikom njegovog formiranja mnogi jezici su imali uticaja, poput Objectiv-C, Rust, Haskell, Ruby, Python, C...

### 3.11 Qore

Qore je dinamički, interpretirani skript jezik viskog nivoa. [14]Tvorac Qora je David Nichols. Podržava više paradigmi, imperativnu, funkcionalnu, proceduralnu i objektno-orijentisanu. Ima podršku za multithreading(to znači da se više od jednog dela koda može pokrenuti istovremeno). Takođe u njemu je moguće i automatsko upravljanje memorijom, a sadrži i sakupljač smeća(eng. garbage collector). Na njega su imali uticaja Perl, Java, D i C++ .

## 4 Zaključak

Kroz ovaj seminarski rad upoznali smo se sa razvojem i značajnim karakteristikama programskog jezika D. Uticaj drugih jezika na njegov nastanak slikovito je predstavljen razvojnim stablom, a kroz sekcije ovog rada ukratko su obrađeni i oni, kao i jezici koji su nastali uzimajući neke koncepte jezika D. U Literaturi su priložene zvanične stranice ovih programskih jezika, kao i sajtovi na kojima možete dobiti više informacija o njima.



## Literatura

- [1] Apple. O programskom jeziku Swift. <https://developer.apple.com/swift/>.
- [2] C-history. Istorija jezika C. <https://fresh2refresh.com/c-programming/c-language-history/>.
- [3] D. Zvanična stranica programskog jezika D. <https://www.dlang.org/>.
- [4] Dpedia-Genie. O programskom jeziku Genie. [http://dbpedia.org/page/Genie\\_\(programming\\_language\)](http://dbpedia.org/page/Genie_(programming_language)).
- [5] Dpedia-MiniD. O programskom jeziku MiniD. <http://dbpedia.org/page/MiniD>.
- [6] Eiffel. Zvanična stranica programskog jezika Eiffel. <https://www.eiffel.org/>.
- [7] GeeksforGeeks. O programskom jeziku C++. <https://www.geeksforgeeks.org/c-plus-plus/>.
- [8] GeeksforGeeks-CSharp. O programskom jeziku C. <https://geeksforgeeks.org/csharp-programming-language/>.
- [9] Java. Zvanična stranica programskog jezika Java. <https://java.com/>.
- [10] Javatpoint-Java. Istorija o Javi. <https://www.javatpoint.com/history-of-java>.
- [11] Javatpoint-Python. Istorija jezika Python. <https://www.javatpoint.com/python-history>.
- [12] Opensource. O programskom jeziku D. <https://opensource.com/article/17/5/d-open-source-software-development>.
- [13] Python. Zvanična stranica programskog jezika Python. <https://www.python.org/>.
- [14] Qore. Zvanična stranica programskog jezika Qore. <http://qore.org/>.
- [15] Swift. Zvanična stranica programskog jezika Swift. <https://docs.swift.org/swift-book/>.
- [16] Vala. Zvanična stranica programskog jezika Vala. <https://www.vala-project.org/>.
- [17] WikiBooks. Dodatna literatura o jeziku D. [https://en.wikibooks.org/wiki/D\\_Programming](https://en.wikibooks.org/wiki/D_Programming).