

# Razvoj programskog jezika C#

Seminarski rad u okviru kursa  
Dizajn programskih jezika  
Matematički fakultet

Lana Šoškić  
soskiclana16@gmail.com

20. decembar 2019

## Sažetak

U ovom radu prikazaćemo osnove kako programskog jezika C#, tako i jezika koji su imali najveći uticaj na nastanak pomenutog. Pored ključnih karakteristika “preteča”, radi boljeg razumijevanja koncepta samog jezika, obradićemo i verzije C# 2.0 i 3.0.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Osnovno o C#-u</b>	<b>2</b>
2.1	.NET Framework . . . . .	3
2.2	C# 1.0 . . . . .	4
2.3	C# 2.0 . . . . .	4
2.4	C# 3.0 . . . . .	5
<b>3</b>	<b>Razvojno stablo</b>	<b>5</b>
3.1	C . . . . .	5
3.2	C++ . . . . .	6
3.3	Java . . . . .	6
3.4	Delphi . . . . .	7
3.5	ML, Haskell . . . . .	7
<b>4</b>	<b>Zaključak</b>	<b>8</b>
	<b>Literatura</b>	<b>9</b>

# 1 Uvod

Jezik je živa materija, i kako jezici koje koristimo u svakodnevnom govoru imaju potrebu za konstantnim promjenama, tako i programski jezici stalno evoluiraju. U nekim slučajevima je impresivno kako relativno stariji jezici kao što su C++, Java, ili JavaScript mogu imati dobru podršku za elemente funkcionalnog programiranja za koje nisu dizajnirani. Problem nastaje kada nove funkcije i biblioteke kao karakteristike ne zamjenjuju stare, već se samo na njih nadovezuju. U programiranju, više ne mora nužno da znači bolje, već sam princip izražavanja istog koncepta na različite načine čini dati jezik konfužnim. Uzimamo za primjer jezik Scala, čije je kodove zbog količine ugrađenih funkcija teško pratiti, dok je jezik Go svoju popularnost izgradio na jednostavnosti i “savršenosti” selekciji istih.

Jedan od primjera jezika, koji čini dobru kombinaciju ranije nastalih, namijenjen za izradu aplikacija je jezik C#, nastao 2000. godine. Kao i većina modernih jezika, podržava više paradigmi (objektno-orijentisanu, imperativnu, deklarativnu), a jedna od glavnih prednosti je jasna sintaksa koja proizilazi iz one koja opisuje C/C++. Logo jezika C# prikazan je na slici 1.



Slika 1: Logo programskog jezika C#

## 2 Osnovno o C#-u

Anders Hejlsberg<sup>1</sup> je 1999. osnovao tim za izradu novog programskog jezika idejno baziranog na jeziku C, ali objektno-orijentisanog. U julu 2000. godine, na Konferenciji profesionalnih programera, predstavljeno je .NET okruženje sa, sada već preimenovanim, jezikom C#<sup>2</sup>, koji je zvanično objavljen 2001., kao nasljednikom jezika C i C++. Hejlsberg, kao glavni dizajner C#-a u Microsoft-u, je u mnogim intervjuima isticao da su upravo nedostaci u drugim programskim jezicima (C++, Java, Delphi, Smalltalk) doveli do stvaranja istog.

---

<sup>1</sup>Anders Hejlsberg - Danski softverski inženjer koji je osmislio nekoliko popularnih programerskih jezika (Turbo Pascal, Delphi, C#, TypeScript)

<sup>2</sup>Dobio je ime inspirisano muzičkom notacijom koja znači da se napisana nota izvodi za pola koraka više

C# je objektno-orijentisan jezik opšte namjene koji pruža fleksibilnost i omogućava stvaranje što dugotrajnijeg softvera. Koje su ključne karakteristike navedenog jezika?

- C# je moderan i jednostavan. Dizajniran je da ima u vidu poslovanje i potrebe preduzeća, tako da se za iste mogu stvarati razne vrste softvera koristeći se jednim jedinim programskim jezikom. Podržava potrebe Web-a, mobilnih uređaja i aplikacija. Neke od funkcija C#-a podržavaju auto inicijalizaciju tipova i kolekcija, lambda izraze, dinamičko programiranje, asinhrono programiranje, napredno debagovanje (uklanjanje grešaka), rukovanje izuzecima. C# izbjegava složenost i nestrukturirane jezičke karakteristike.
- C# je brz i open source<sup>3</sup>. C# jezičke specifikacije, kompajleri, i vezane alatke su open source projekti u Github-u<sup>4</sup>.
- C# je portabilan. Mogu se izraditi .NET aplikacije koje se zatim mogu rasporediti na Windows, Linux i Mac platformama.
- C# je siguran i efikasan. C# je type safe<sup>5</sup> jezik. C# dozvoljava pisanje sigurnih i efikasnih kodova, gdje izbjegava tipiziranje/kastovanje koje može dovesti do gubitka podataka ili sličnih problema.
- C# je svestran. Nalazi primjenu u velikom broju tipova aplikacija (Web aplikacije, video igre, gaming konzole, Windows client aplikacije, i slično).

## 2.1 .NET Framework

Microsoft .NET je softverska platforma koja može biti instalirana na računarima koje pokreće Windows OS. Ista je sastavljena od alatki, programskih jezika i biblioteka za izradu različitih aplikacija.

.NET Framework je originalna implementacija .NET-a koja pruža podršku za vođenje web stranica, desktop aplikacija i njima sličnih na Windows-u.

Dvije bitne komponente čine .NET Framework:

- **CLR (Common Language Runtime)** je mehanizam za izvršavanje koji rukuje aplikacijama. Pokriva upravljanje nitima, garbage collection, validaciju tipova, rad sa izuzecima, i slično.
- **Class Library** kao skup API-ja (Application Programming Interface) i tipova za rad. Dakle, uključuje interfejse za čitanje i pisanje datoteka, povezivanje sa bazom podataka, crtanje itd.

.NET aplikacije su pisane u C#, F# ili Visual Basic programskim jezicima, dok je način rada samog mehanizma prikazan na slici 2.<sup>6</sup>

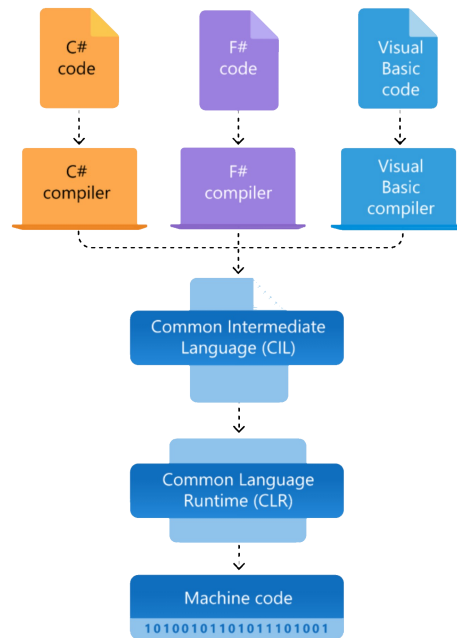
---

<sup>3</sup>Softver sa Open Source licencom je prema izvornom kodu dat na korišćenje svakome ko ima potrebu za tim, tako da korisnik može da njegov sadržaj, sa ciljem poboljšanja performansi

<sup>4</sup>Veb-baziran hosting servis za kontrolu verzije

<sup>5</sup>Type safety znači da će kompajler validirati tipove tokom prevođenja

<sup>6</sup>Tekst je većim dijelom preuzet sa stranica [2] i [5].



Slika 2:

## 2.2 C# 1.0

C# 1.0, objavljen u Visual Studio 2002. godine, imao je dosta sličnosti sa programskim jezikom Java. Upravo odatle potreba za stvaranjem "jednostavnog, modernog, objektno-orientisanog jezika opšte namjene". Međutim, sama verzija je pretrpjela mnogo kritika i okarakterisana je za imitaciju jezika Java u kojoj se isključuju pouzdanost, produktivnost i sigurnost. Posmatrajući verzije koje sljeduju, ovoj verziji su falile mogućnosti asinhronog programiranja<sup>7</sup>, kao i mnoge funkcionalnosti koje donose generički tipovi.

Pojam Managed Code-a<sup>8</sup> je takođe predstavljen ovom verzijom.

## 2.3 C# 2.0

Microsoft je objavio drugu verziju C#-a sa Visual Studio-om 2005. godine, koja omogućava programiranje aplikacija na generički način. Neke

<sup>7</sup>Za razliku od sinhronog programiranja, kod kog se mora čekati izvršavanje određene metode do kraja, i sam rezultat koji prosljeđuje, da bi se nastavilo sa izvršavanjem programa, kod asinhronog se pokreće ta metoda, i do završetka njenog izvršavanja programer ima mogućnost rada drugog dijela posla.

<sup>8</sup>CLR je zadužen za preuzimanje ovog koda, prevođenje istog u mašinski, a zatim se kao takav izvršava.

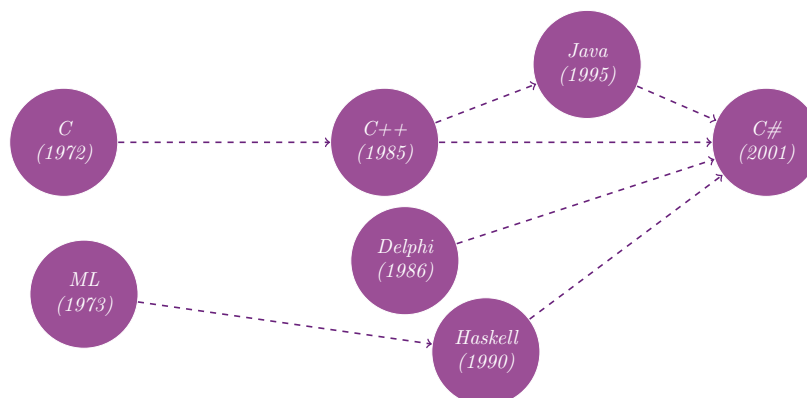
od karakteristika koje čine verziju 2.0 boljom jesu generički tipovi podataka, parcijalne klase i metode, anonimne metode, kovarijansa i kontravarijansa<sup>9</sup>, koje pružaju veliku fleksibilnost, i slično.

## 2.4 C# 3.0

Sa verzijom 3.0, objavljenom 2007. godine, jezik se počinje razlikovati od jezika Java, tako što se uvode karakteristike slične onim u funkcionalnim programskim jezicima. Definirane su funkcije za Lambda izraze, kao i implicitni i anonimni tipovi<sup>10</sup>, i stabla.

## 3 Razvojno stablo

Na slici 3 prikazani su programski jezici koji su imali najveći uticaj na nastanak C#-a. Sam jezik je dizajniran zato što Sun (kasnije Oracle) nije podržavao ideju da Microsoft pravi izmjene u jeziku Java, pa su se isti odlučili da umjesto toga naprave svoj, odakle i potiču značajne sličnosti među pomenutim.



Slika 3: Razvojno stablo jezika C#

### 3.1 C

C je proceduralni programski jezik opšte namjene, kreiran od strane Denisa Ričija 1972. godine. Ovo nije "high-level" opširan jezik, niti je jezik stvoren za neke posebne vrste primjena, ali je upravo zbog svoje opštosti i nepostojanja ograničenja prihvatljiviji i efikasniji od većine. Tijesno je povezan sa operativnim sistemom UNIX na kom je razvijen, jer su i sistem i većina programa na UNIX-u napisani baš u C-u. Bitne ideje C-a potiču

<sup>9</sup>Kovarijansa nam omogućava da zamijenimo osnovni tip izvedenim tipom, dok kontravarijansa radi obrnuto. Obije osobine se počinju primijenjivati i na generičkim tipovima, ali tek od verzije 4.0.

<sup>10</sup>Anonimni tip omogućava da se ne definiše eksplicitno tip, već tip definiše sam kompajler.

od programskog jezika BCPL<sup>11</sup>, indirektno preko jezika B nastalog 1970. Za razliku od pomenutih, C nudi mnoštvo tipova podataka, među kojima su osnovni (znakovi, cjelobrojne vrijednosti i vrijednosti iz područja realnih brojeva) u više veličina, i izvedenih kreiranih pokazivačima, poljima, strukturama. Nudi osnovne konstrukcije za kontrolu toka: grupisanje naredbi, donošenje odluka (if-else), izbor (switch), petlje sa uslovom na početku (while) i na kraju (do), te izlaz iz petlje prije kraja (break)[3].

Mnogi jezici kasnije direktno ili indirektno pozajmljuju karakteristike od C-a, među kojima su C++, C#<sup>12</sup>, Go, Java, JavaScript, Perl, PHP, Python. Većina izražava vrlo sličnu sintaksu C-u (Python kao jako bitan izuzetak), kombinujući istu sa proširenim sistemom tipova podataka, kao i semantikom koja može biti radikalno drugačija.

### 3.2 C++

C++ kao ideja nastaje 1979. godine kao dio doktorske disertacije Bjarne Stravstroupa. Bjarne je do tada imao prilike raditi sa Simulom 67, prvim jezikom koji je podržan objektno-orijentisanom paradigmom, kada dolazi do saznanja koliko ista može biti korisna za razvoj softvera. Tada počinje rad na "C-u sa klasama", kao "nadskupu" jezika C, poznatog po svojoj portabilnosti bez žrtvovanja brzine i "low-level" funkcionalnosti. Novi jezik uključuje klase, automatske argumente funkcija, jaku provjeru tipova podataka, kao i četiri osnovna principa objektno-orijentisane paradigme: enkapsulaciju, apstrakciju, nasljeđivanje i polimorfizme. Napokon, 1983. jezik dobija ime C++, gdje je ++ znak inkrementacije, kao uvid u ideju nastanka istog. Sa pomenutom godinom navode se i nove karakteristike, među kojima su virtuelne i overloaded<sup>13</sup> funkcije, reference sa simbolom &, ključna riječ const, komentari u jednom redu koristeći se znakom // [6].

Iako se vjeruje da je navođenje C++-a kao glavnog preteče C#-u, mjesto programskog jezika Java, čist marketinški trik, neosporno je da su ovi jezici, kao pripadnici iste porodice, u mnogim aspektima slični. Tako je sintaksa (ne računajući onu osnovnu) C#-a bliža C++-u u određenim područjima, među kojima su manipulacija pokazivačima (dok Java istu ne posjeduje), deklaracija izvođenja/porijekla (primjera radi, pišemo *public class Student: Kolokvijum, Ispit* umjesto *public class Student extends Ispit implements Kolokvijum*) i overloading operatora<sup>14</sup>.

### 3.3 Java

Java je objektno-orijentisani programski jezik, koji je razvila kompanija Sun Microsystems (sada već Oracle) početkom 1990-ih godina. To

---

<sup>11</sup>BCPL je proceduralni jezik kreiran od strane Martina Ričardsa, originalno namijenjen pisanju kompajlera za ostale jezike. Opisuje se kao jezik "bez tipova podataka".

<sup>12</sup>Jezik originalno nazvan Cool (C-like Object-Oriented Language).

<sup>13</sup>Dvije različite funkcije mogu imati isto ime ukoliko imaju ili različit broj parametara, ili se bar jedan par od navedenih razlikuje po tipu.

<sup>14</sup>C++ ima za cilj da se korisnički definisani tipovi ponašaju slično onim ugrađenim. Na taj način su dozvoljene operacije među istim, koristeći se identičnom sintaksom koja bi se koristila i za one osnovne.

je jezik opšte primjene koji ima podršku za konkurentno programiranje i osmišljen je uz *"write once, run anywhere"* princip, u smislu da jednom preveden Java kod može biti izveden na svim platformama koje podržavaju isti, bez potrebe za novim prevodenjem. Razvijen je od strane Džejsma Goslinga i objavljen 1995. godine kao jezik čija sintaksa vuče korijene iz programskih jezika C i C++, sa manje "low-level" funkcionalnosti.

Upravo zbog pomenute fleksibilnosti, Java predstavlja prvi izbor za razvoj web aplikacija. Svoju popularnost stiče osobinama kao što su razumljiv i jednostavan kod, JRE (Java Runtime Environment) koji čini jezik nezavisnim, Java API koji kompletira programiranje sa stotinama klasa i preko 50 ključnih riječi, stabilno i programiranje prilagođeno kako računarima, tako i mobilnim uređajima.

Pored sličnosti u korijenima, koje se naročito ističu u pogledu sintakse, C# i Java su jezici koji se konceptualno poklapaju. Tako se C#, takođe, odriče višetrakog nasljeđivanja klasa, ali ne i interfejsa, dok je u okviru samog jezika uključena nova funkcija, upravo po uzoru na programski jezik Java. U pitanju je garbage collector, odnosno automatsko upravljanje memorijom, koje radi po principu oslobađanja dijela memorije, zauzetog za objekat klase, momenta kada se sve akcije vezane za sam objekat izvrše. U suprotnom, taj dio bi ostao popunjen i nadalje neupotrebljiv.

### 3.4 Delphi

Delphi je i objektno-orijentisani jezik i integrisano razvojno okruženje (IDE), objavljeno od strane Embarkadero kompanije. Baziran je na jeziku Object Pascal, a dok IDE radi na Windows-u, kompajler cilja na Windows, macOS, Linux, iOS i Android. Plan dizajniranja samog proizvoda jeste da obezbijedi bazu podataka, kao ključno sredstvo za povezivanje programera. Glavni arhitekta bio je Anders Hejlsberg, koji je, takođe, razvio Turbo Pascal (i opisani C#). Sam jezik podržava brz razvoj aplikacija (rapid application development - RAD), i osobine kao što su application framework<sup>15</sup> i dizajner grafičkog interfejsa.

### 3.5 ML, Haskell

ML (Meta Language) je funkcionalan jezik opšte namjene, takođe poznat kao "Lisp sa tipovima podataka". Koristi Hindli-Miler sistem tipova<sup>16</sup>, koji izrazima automatski dodjeljuje tipove bez eksplicitnog navođenja istih. ML omogućava call-by-value strategiju procjene<sup>17</sup>, automatsko upravljanje memorijom putem garbage collector-a, statičko tipiziranje, algebarske tipove podataka<sup>18</sup>, upravljanje izuzecima. Jezik je razvio Robin

<sup>15</sup>Softver biblioteka koja pruža fundamentalnu strukturu za podršku razvoja aplikacija na određenom okruženju.

<sup>16</sup>Klasičan sistem tipova za lambda proračune sa parametričkim polimorfizmima.

<sup>17</sup>Strategija koja utvrđuje dvije stavke - kada izvršiti procjenu argumenata u pozivu funkcije i koju vrstu vrijednosti proslijediti funkciji.

<sup>18</sup>Pojam koji se posebno vezuje za funkcionalno programiranje, tzv. kombinatorni tipovi, odnosno oni stvoreni kombinacijom drugih tipova.

Milner sintaksom inspirisanom ISWIM<sup>19</sup>-om. Iako okarakterisan “nečistim” funkcionalnim jezikom, ML je svojim idejama uticao na razvoj jako “čistog” Haskell-a.

Haskell je isključivo funkcionalan jezik sa statičkim tipiziranjem i lijenom procjenom, tj. call-by-need strategijom<sup>20</sup>. Semantički je baziran na jeziku Miranda, objavljenom neposredno prije navedenog. Opisan je “čistim” funkcionalnim jezikom, što znači da funkcije generalno ne posjeduju “sporedne efekte”<sup>21</sup>.

Haskell je bio glavni projekat u sferi funkcionalnog programiranja za Erika Meijera, danskog informatičara koji kasnije, za vrijeme rada u Microsoft-u, isto razvija i u jezicima C#, Visual Basic, LINQ. Stoga je C# dobar primjer opšteg jezika dovoljno fleksibilnog da podržava više paradigmi. On od LINQ-a, koji je nastao direktnim uticajem Haskell-a, nasljeđuje jedan od glavnih funkcionalnih koncepata: funkcije višeg reda, kao i metode proširenja, IEnumerables i tako dalje. Dakle, C# sadrži neke izvorne karakteristike povezane sa funkcionalnom paradigmom, tako da je fer reći da možete napraviti sjajne stvari koristeći ga[7].

## 4 Zaključak

Iako relativno mlad, C# je već jednom proglašen za jezik godine. Zbog svoje svestranosti i mogućnosti da služi kako malim kompanijama, tako i globalnim korporacijama, nastavlja da bude prirodan izbor većini programera. Tekst pruža uvid u istoriju, razvoj i osnovne karakteristike samog jezika, dok se za svaki detaljniji prikaz i praćenje novih verzija preporučuje zvanična strana [1].

---

<sup>19</sup>ISWIM (If you See What I Mean) - abstraktan programski jezik koji se, iako neimplementiran, dokazao jako korisnim za razvoj funkcionalnih jezika kao što su SASL, Miranda, ML, Haskell.

<sup>20</sup>Strategija koja odlaže procjenu tipa izraza, sve dok vrijednost tog izraza nije potrebna za nastavak rada programa.

<sup>21</sup>Program ima sporedne efekte ukoliko izvodi primjetne aktivnosti pored one svoje primarne (vraćanje neke vrijednosti), kao što je prikazivanje slike na ekranu ili provjere tekuće godine.



## Literatura

- [1] Zvanična stranica programskog jezika C#: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
- [2] Zvanična stranica Microsoft-a: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction>
- [3] Dennis M. Ritchie, Brian W. Kernighan, "Programski jezik C": [http://www.tfzr.uns.ac.rs/Content/files/0/Programski\\_jezik\\_C.pdf](http://www.tfzr.uns.ac.rs/Content/files/0/Programski_jezik_C.pdf)
- [4] Brief History of C Programming: <https://cs-fundamentals.com/c-programming/history-of-c-programming-language.php>
- [5] Visual C# 2010: <https://camhang.files.wordpress.com/2012/09/beginning-visual-c-2010.pdf>
- [6] History of C++ : <http://www.cplusplus.com/info/history/>
- [7] C# Functional Programming: <https://dev.to/gabrielschade/c-functional-programming-4f7o>