

**POREĐENJE ANGULARJS I ANGULAR 5 TEHNOLOGIJE I UPOTREBA  
BOOTSTRAP BIBLIOTEKE****COMPARISON OF ANGULARJS AND ANGULAR 5 TECHNOLOGY AND THE USE OF  
THE BOOTSTRAP LIBRARY**

Tamara Mrkšić, *Fakultet tehničkih nauka, Novi Sad*

**Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – *Objektno-orijentisano programiranje predstavlja jednu od osnovnih programskih paradigmi u računarstvu. Zato tehnologije pokušavaju da poštuju glavne principe objektno-orijentisanog programiranja. U radu je istražena AngularJS i Angular 5 tehnologija. Zatim je izvršeno upoređivanje Angular 5 i AngularJS tehnologije. Objasnjena je upotreba Bootstrap biblioteke.*

**Ključne reči:** *Objektno-orijentisano programiranje, AngularJS, Angular 5, Bootstrap, AngularJS vs Angular5*

**Abstract** – *Object-oriented programming is one of the basic programming paradigms in computing. Therefore, technologies try to follow the main principles of object-oriented programming. The paper deals with AngularJS and Angular 5 technology. Then Angular 5 and AngularJS Technology were compared. The use of the Bootstrap library is explained.*

**Keywords:** *Object-oriented programming, AngularJS, Angular 5, Bootstrap, AngularJS vs Angular5*

**1. UVOD**

Rad je podeljen u šest poglavlja. Prvih pet poglavlja opisuje problematiku rada iz drugog ugla, kako bi se njihovim spajanjem dobila sveobuhvatna slika teme.

Kroz drugo poglavlje će biti date teorijske osnove, neophodne za problematiku rada. U trećem poglavlju je dato poređenje tehnologija kroz prikaz uvedenih izmena u Angular 5 tehnologiju. Kroz četvoro poglavlje će biti opisana Bootstrap biblioteka. U petom poglavlju je dat zaključak, odnosno sumiranje samog rada. Šestim poglavljem je predstavljena literatura, neophodna za pisanje rada.

Na kraju rada se nalazi kratka biografija autora.

**2. TEORIJSKE OSNOVE**

U ovom poglavlju će biti opisane teorijske osnove, koje su vezane pre svega za oblast objektno-orijentisanog programiranja. Zatim biće pojašnjeno čemu su namenjeni JavaScript, AngularJS, TypeScript, kao i Angular 5.

**2.1 Objektno-orijentisano programiranje**

U računarstvu, objektno-orijentisano programiranje (OOP) je jedna od programskih paradigmi. Od ranih 1980-tih pa do danas ova paradigma je postala najuticajnije paradigme u komercijalnom razvoju softvera. Programski jezici poput C++-a i Jave programskog jezika su svojom popularnošću utvrdili vodeći status OOP-a kao *de facto* obaveznog pristupa pri razvoju softvera danas, te i jezici koji tradicionalno nisu bili objektno-orijentirani su naknadno dodali OOP koncepte [1].

Osnovni koncepti objektno-orijentisanog programiranja su:

**a. Apstrakcija**

Osnovni nivo apstrakcije se svodi na uočavanje šablona za objekte sa više zajedničkih osobina. Na osnovu tih šablona se grade klase sa kontrolisanim uključivanjem i isključivanjem nekih detalja o objektima, što je još jedan nivo apstrakcije.

**b. Enkapsulacija**

Ona predstavlja postupak objedinjavanja stanja i ponašanja objekta u jednu celinu. Tako organizovane objekte je lakše kontrolisati i onemogućiti neovlašćeni pristup. Korisnici sa objektom komuniciraju samo kontrolisano, pomoću javnih metoda i ne mogu neovlašćeno menjati unutrašnja stanja objekta. Samo internim metodama objekta je omogućen pristup stanjima.

**c. Nasleđivanje**

To je posledica generalizacije kao metoda za modelovanje objekata. Obezbeđuje redukciju i lakše održavanje koda. Tako se zajedničke osobine i funkcionalnosti pišu samo jednom, a njihove eventualne izmene se vrše u osnovnoj klasi.

**d. Polimorfizam**

To je osobina da se ista metoda osnovne klase izvršava na različite načine u zavisnosti od toga kojoj izvedenoj klasi objekat koji ga poziva pripada. Tako se u izvedenim klasama mogu predefinisati neke od nasleđenih funkcionalnosti u skladu sa specifičnostima izvedene klase. Na taj način je omogućeno da objekti različitih izvedenih klasa reaguju različito izvršavajući istu funkcionalnost osnovne klase.

**NAPOMENA:**

**Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinać, vanr. prof.**

## 2.2 Angular

Bez obzira na vrstu Angular tehnologije, bitno je pojasniti čemu je ona uopšte namenjena. Angular je objavljen kao projekat još 2009. pod nazivom `<angular/>`, i njegova osnovna namena je bila pomoć web programerima, kao i web dizajnerima da lakše izrade web stranice, koristeći html oznake. Sam naziv Angular potiče upravo od zagrada odnosno `<>`, koje obuhvataju sve HTML oznake, kao na primer `<div>`, `<span>` i druge.

Angular je ustvari JavaScript okvir, koji pomaže programerima da razvijaju svoje aplikacije. Ova biblioteka nudi niz funkcija, koje ga čine trivijalnim za implementaciju kompleksnih zahteva modernih aplikacija, kao što su vezivanje podataka, rutiranje i animacije.

On takođe nudi niz konverzija kako pristupati razvoju aplikacije, što je veoma korisno za velike timove, koji moraju raditi zajedno. Angular predstavlja jedinu JavaScript biblioteku, koja sadrži sveobuhvatni vodič za stil pisanja koda. Dobro radi u form-based aplikacijama, a pogodan je i za velike i složene aplikacije, kao i one koje treba da rade u više razvojnih okruženja [2].

## 2.3 JavaScript

JavaScript ili skraćeno JS predstavlja objektno zasnovani skriptni jezik. Pored HTML-a i CSS-a, on predstavlja jedan od osnovna tri jezika WWW-a. Omogućava interaktivne web stranice, i zato predstavlja osnovni deo web aplikacija. Velika većina web stranica ga koristi, a web pretraživači imaju i JavaScript mehanizam za njegovo izvršavanje. Uključuje se u web stranice kako bi postale dinamičnije. HTML se koristi samo za oblikovanje i uređivanje elemenata stranice (tekst, forme, linkovi, tabele), ali ne postoji način kako da im se nametne ponašanje. Mogućnost uključivanja JavaScript skripte daje mnogo veću kontrolu kako da se ponaša web stranica.

## 2.4 AngularJS

AngularJS ili poznatiji pod nazivom Angular.js je JavaScript okvir, baziran na „front-end“ web aplikacijama. Cilj ove platforme jeste pojednostavljenje razvoja i testiranje takvih aplikacija pružanjem okvira za arhitekturu klijent-model-pogled-kontroler (MVC) i model-pogled-pogled-model (MVVM), zajedno sa komponentama, koje se obično koriste u bogatim internet aplikacijama.

Bitno je napomenuti da se pod AngularJS-om podrazumeva ustvari prva verzija i sve njene podverzije Angulara.

Ovaj okvir radi po principu tako što prvo čita HTML stranicu, koja ima dodatne prilagođene tag attribute. Zatim tumači te attribute kao uputstva za povezivanje ulaznih i izlaznih delova stranica sa modelom, koji predstavlja standardne JavaScript promenljive. Vrednosti ovih promenljivih se mogu ručno postaviti u kodu, ili preuzeti iz statičkih ili dinamičkih JSON resursa.

AngularJS je zasnovan na uverenju da se deklarativno programiranje treba koristiti za kreiranje korisničkog interfejsa i povezivanje softverskih komponenti. Sa druge strane, neophodno je imperativno programiranje za definisanje same poslovne logike aplikacije. Samim tim, ovaj okvir produžava i proširuje tradicionalni HTML da bi predstavio dinamički sadržaj putem dvosmernog povezivanja podataka, koji omogućava automatsku sinhronizaciju modela i prikaza. Kao rezultat, AngularJS naglašava eksplicitnu manipulaciju DOM stabla sa ciljem povećanja performansi i testiranja.

Ciljevi dizajna AngularJS uključuju:

- razdvajanje DOM manipulacije od logike aplikacije (Na težinu dosta utiče sama struktura koda.)
- odvajanje klijentske strane od servera (Ovo omogućava ravnopravnim timovima da paralelno napreduju i omogućavaju ponovnu upotrebu obe strane.)
- obezbeđivanje strukture za putanju izgradnje aplikacije (Ovo podrazumeva dizajniranje UI-ja, pisanje poslovne logike i testiranje.) [3]

On pojednostavljuje razvoj aplikacija predstavljanjem višeg stepena apstrakcije programeru. Kao i svaka apstrakcija, ona dolazi po cenu fleksibilnosti. Drugim rečima, nije baš svaka aplikacija najbolja za AngularJS, jer je on napravljen uz primenu CRUD aplikacije. Međutim, na sreću većina web aplikacija predstavlja upravo CRUD aplikacije. Da bi se bolje razumelo za šta je dobar, najlakše je dati primere za šta nije. Igre i GUI uređaji su primeri aplikacija sa intenzivnom i neobičnom manipulacijom DOM staba. U takvim slučajevima je bolje koristiti neku biblioteku sa nižim stepenom apstrakcije, kao što je JQuery.

## 2.5 TypeScript

TypeScript predstavlja programski jezik, namenjen za izgradnju web aplikacija, kojeg razvija i održava Microsoft. Njegova sintaksa je veoma slična sintaksi JavaScript-a, s tim da su dodate razne napredne mogućnosti. Jedna od njih je opciono statično pisanje, koje olakšava pisanje velikih web aplikacija. Statično pisanje zahteva dodeljivanje tipa promenljivima, koje se kreiraju. JavaScript je sam po sebi dinamički pisan jezik, u kojem se ne mora unapred dodeliti tip promenljivoj. Sa statičkim pisanjem, kompajler može detaljnije da prikaže i da uvid u grešku, ukoliko do nje dođe. Nastao je zbog nedostataka JavaScript-a za razvoj velikih aplikacija od strane Microsoft-a i njegovih klijenata. Izazovi sa kompleksnim JavaScript kodom doveli su do potražnje za prilagođenim alatima da bi se olakšao razvoj komponenti u jeziku.

## 2.6 Angular 5

Angular 5 predstavlja novu verziju AngularJS okvira, koju je razvio Google. Predstavljen je kao kompletno prepisan AngularJS, ali sa različitim poboljšanjima. Ta poboljšanja se ogledaju u optimizovanoj građi i bržem kompajliranju. On je JavaScript okvir, koji koriste

programeri širom sveta za izgradnju web, desktop i mobilnih aplikacija. Takođe, Angular 5 kombinuje deklarativne „template“-ove, „dependency injection“ i integrisane najbolje prakse za rešavanje raznih razvojnih problema.

Mnogobrojne web platforme, kao što su Google Adwords, Google fiber, Adsense i druge koriste Angular za izgradnju korisničkog interfejsa.

### **3. POREĐENJE ANGULARJS I ANGULAR 5 TEHNOLOGIJE**

Novine, koje je uneo Angular 5, je najbolje shvatiti kroz njegovo poređenje sa AngularJS-om. Samim tim, u ovom poglavlju će biti opisane razlike između AngularJS i Angular 5 tehnologije, prolazeći kroz poglavlja koja predstavljaju njihove razlike. Angular 5 predstavlja noviju verziju Angulara, i samim tim je doneo dosta različitih promena u pogledu funkcionalnosti, kao i određenih novina. Date promene će detaljnije biti opisane u nastavku.

#### **3.1 TypeScript**

TypeScript ustvari predstavlja jezik, kao i set alatki za generisanje JavaScripta. On predstavlja projekat otvorenog koda, koji pomaže programerima da pišu velike JavaScript projekte.

On ustvari generiše JavaScript. Umesto da zahteva potpuno novo radno okruženje, JavaScript generisan preko TypeScripta, može ponovo da upotrebi sve postojeće alatke JavaScripta, radne okvire i biblioteke, koje su dostupne za JavaScript.

Međutim, TypeScript jezik i kompajler približavaju razvoj JavaScripta tradicionalnijem objektno-orijentisanom programiranju.

#### **3.2 Progresivne web aplikacije**

Trenutni razvoj progresivnih web aplikacija je razrađen i složen proces. Potrebno je voditi računa kako tokom razvoja, tako i u upotrebi da ne dođe do greške ili da nijedna starija verzija ne bude oštećena. Međutim ovde je došlo do promene u Angularu 5.

Razvoj progresivnih web aplikacija treba toliko da se pojednostavi da mogu podrazumevano da se kreiraju. U ovom slučaju korisnici, kao i programeri bi trebali imati jednake koristi od ovog. Uz Angular CLI, Angular 5 ima mogućnost samostalnog kreiranja konfiguracije i koda.

#### **3.3 Komponente**

Komponenta ustvari predstavlja dekorator, koji se mapira na klasu kao Angular komponentu. Ona pruža metapodatke konfiguracije kako se komponenta treba obrađivati, instancirati, i koristiti u toku izvršavanja.

Komponente čine najosnovniji UI Angular aplikacije. Aplikacija ustvari predstavlja stablo komponenti. One su podgrupa direktiva, koje su uvek povezane za neki šablon. Za razliku od drugih direktiva, samo jedna komponenta se može instancirati po elementu u šablonu.

#### **3.4 Podrška za multi alijase**

U Angularu 5 je moguće davati više imena komponentama i direktivama prilikom eksporta. Eksport komponente sa više imena može pomoći korisnicima da migriraju bez prekidanja postojećeg koda.

#### **3.5 Rad sa događajima**

Ukoliko je potreban rad sa događajima, to je omogućeno kroz korišćenje interpolacije, vezivanje osobina, vezivanje podataka, kao i vezivanje događaja. Na drugačiji način je implementirano u Angularu 5, u odnosu na AngularJS tehnologiju.

#### **3.6 Forme**

U Angularu 5, kada su u pitanju forme, poboljšane su performanse i moguće je specificirati kada je potrebno izvršiti validator u formi. Svaki put kada se promeni vrednost u „FormControl“, validacija će se izvršiti shodno tome, i to potencijalno prilikom svakog pritiska na tastaturu. U slučaju složenih validacija, to može dovesti do veoma loših performansi. Sa novom opcijom „UpdateOn“, Angular 5 omogućava preciznije određivanje kada se validacija treba izvršiti. Tu se može izabrati promena, koja određuje prethodno ponašanje, potvrđivanje forme i „blurovanje“.

#### **3.7 HttpClient API**

Pre verzije Angulara 5 za HTTP zahteve je korišćen „@angular/http“ u Angular aplikacijama. Međutim, sa verzijom 5, ovakvo korišćenje je postalo zastarelo. Umesto njega se koristi HttpClient API iz „@angular/common/api“, i on je preporučen za upotrebu u aplikacijama.

#### **3.8 RxJS**

Reaktivno programiranje je paradigma asinhronog programiranja, koja se odnosi na tokove podataka i širenje promena. RxJS je biblioteka za reaktivno programiranje pomoću „posmatrača“, što olakšava sastavljanje asinhronog poziva ili poziva zasnovanog na odgovoru. RxJS obezbeđuje implementaciju tipa „Observable“, koji je potreban sve dok tip ne postane deo jezika i dok ga pretraživači ne podrže.

#### **3.9 Rutiranje i moduli**

Ruter predstavlja opcioni servis, koji predstavlja određeni prikaz komponente za datu URL adresu. Rutirana Angular aplikacija ima samo jedan primerak instanciranog Router servisa. Kada se URL adresa pregledača promeni, taj ruter traži odgovarajuću rutu, iz koje može odrediti komponentu, koja bi trebalo da se prikaže. U okviru rutera ne postoje neke podrazumevane rute. Sve rute je potrebno konfigurirati.

#### **3.10 Animacije**

Paket animacija u Angularu 5 je proširen sa nekoliko sintakasnih elemenata. Sada je moguće reagovati na numeričke promene vrednosti primenom inkrementa i dekrementa, kako bi animirali u skladu sa tranzicijom.

Takođe, u Angularu 5 se animacije mogu aktivirati i deaktivirati korišćenjem vrednosti, koje su povezane sa vezivanjem podataka. Tako na primer kada je u pitanju događaj „disabled“ ukoliko je on „okinut“, samim tim se može pozvati odgovarajuća animacija.

#### 4. UPOTREBA BOOTSTRAP BIBLIOTEKE

Bootstrap predstavlja najpopularniji HTML, CSS i JavaScript okvir za web „front-end“ razvoj aplikacija. Takođe čini odlično rešenje za razvoj odzivnih, „mobile-first“ web stranica. Bootstrap okvir se može koristiti zajedno sa JavaScript web okruženjem, kao i mobilnim okvirima kao što je na primer Android. On uključuje HTML i CSS zasnovane dizajnerske šablone za tipografiju, dugmiće, forme, tabele, navigaciju, modele i mnoge druge, kao i JavaScript dodatke. Ti dodaci omogućavaju lako kreiranje željenog dizajna. Ova biblioteka takođe omogućava kreiranje prilagodljivog dizajna. Takva vrsta dizajna dozvoljava kreiranje web aplikacija, koje se automatski prilagođavaju kako bi dobro izgledale na svim uređajima, od malih telefona do velikih desktop računara.

Kada je Bootstrap u pitanju, izdvajaju se njegove glavne prednosti, a to su:

- jednostavan za upotrebu – potrebno je samo osnovno znanje HTML-a i CSS-a,
- prilagodljive funkcije – odziv CSS-a se prilagođava prema vrsti uređaja, na kom se koristi,
- „mobile-first“ pristup – od verzije 3, „mobile-first“ stilovi predstavljaju deo osnovnog okvira,
- kompatibilnost ka pretraživačima – u te pretraživače spadaju: Chrome, Firefox, Internet Explorer, Edge, Safari i Opera [4].

Postoji više načina kako „doći“ do Bootstrapa prilikom razvoja web aplikacije, a to su njegovim preuzimanjem ili uključivanjem od CDN-a. Ukoliko se koristi način uključivanja, kao rezultat će se učitavati iz keš memorije kada se pokrene aplikacija, što dovodi do bržeg vremena učitanja. Takođe, većina CDN-ova će se pobrinuti da kada se od njega zatraži datoteka, omogućice pristup ka najbržem serveru, što opet vodi do bržeg vremena učitanja. Bootstrap koristi JQuery za JavaScript dodatke kao što su modalni dijalozi i slično. Međutim, ako se koristi Bootstrapov CSS, onda nije potreban JQuery.

#### 5. ZAKLJUČAK

U ovom radu vršeno je upoređivanje AngularJS i Angular 5 tehnologije. Takođe je opisana u najbitnijim crtama Bootstrap biblioteka.

Promena, koja je obeležila Angular 5 pre svega predstavlja uvođenje TypeScripta, i na njemu se on temelji. On predstavlja jezik, koji generiše JavaScript, i zajedno sa kompajlerom približava razvoj JavaScripta tradicionalnijem objektno-orijentisanom programiranju.

Zatim omogućava razvoj progresivnih web aplikacija i pruža podršku za multi alijase. Izgradnja projekta pomoću komponenata znatno olakšava pisanje koda, i njegovo ponovno korišćenje. Takođe, dosta promena je vezano za rad sa događajima i u formama. Angular 5 omogućava razvoj aplikacija korišćenjem HttpClient API-a i reaktivnim programiranjem. Osim toga, promene su nastale i u samom načinu rutiranja i animacijama.

Bez obzira koja je verzija Angulara u pitanju, jedna od najpoznatijih biblioteka, koja se koristi za uređivanje HTML stranica uz njega, je Bootstrap. Ono što se moglo zaključiti jeste da se primenom ove biblioteke dobija savremeni dizajn, „user-friendly“ izgled aplikacije, kao i na određeni način očekivano ponašanje. To je posebno bitno korisnicima, koji su do tada koristili neku sličnu aplikaciju, i samim tim će se lakše prilagoditi novoj, ako se poštuju određeni šabloni prilikom izrade. Na taj način se dobija aplikacija, koja je „lepa na oko“, a sa druge strane prilagođena i lakša za korišćenje.

Rad je praćen sa dosta teorije, koja je bila neophodna za ulazak dublje u samu problematiku teme rada. Kada je u pitanju generalno Angular tehnologija, moglo se zaključiti da predstavlja odličan okvir i pruža dosta mogućnosti za izradu „front-end“ aplikacija. Kroz razlike u verzijama se uvidelo koliko je ustvari bilo „šupljina“ u samom AngularJS-u, koje je popunio Angular 5. Uz povećavanje mogućnosti, koje do tada nisu bile, uvedena su i poboljšanja postojećih funkcionalnosti. Takvim radom je postignuta mogućnost lakše izrade velikih aplikacija, bolje organizacije koda, bržeg pokretanja aplikacije i eventualnih otklanjanja nastalih grešaka tokom izrade.

#### 6. LITERATURA

- [1] <http://www.stroustrup.com/whatis.pdf> (pristupljeno u avgustu 2018.)
- [2] <https://developer.telerik.com/topics/webdevelopment/what-is-angular/> (pristupljeno u avgustu 2018.)
- [3] <https://docs.angularjs.org/> (pristupljeno u avgustu 2018.)
- [4] [https://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](https://www.w3schools.com/bootstrap/bootstrap_get_started.asp) (pristupljeno u avgustu 2018.)

#### Kratka biografija:



**Tamara Mrkšić** rođena je u Novom Sadu 1994. god. Diplomski rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Soft computing odbranila je 2017.god.  
kontakt: tamarica94mrksic@gmail.com