



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



PROGRAMIRANJE U SMART GRID SISTEMIMA

Izrada BSc diplomskog rada

Novi Sad, 2020

Sadržaj

Revizija dokumenta	3
Zvanična FTN uputstva za završne radove	3
Uputstvo za pisanje i formatiranje teksta.....	3
Prpratni materijal uz diplomski rad	3
Izvorni kod.....	3
Slike, grafici i ostali izvorni materijal	3
Opšti principi izrade završnog rada	3
Aktuelne teme.....	4
Smernice za strukturu rada.....	6
Proces prijave.....	7
Očekivan nivo mentorstva	7
Najčešća unapređenja	8

1. Revizija dokumenta

Verzija	Komentar	Datum
v1	Inicijalna verzija dokumenta	17.05.2020
v2	Dodato pogavlje devet, najčešća unapređenja (od 9.1 do 9.8)	13.09.2020
v3	Ekstenzija poglavlja 9.6	22.09.2020

2. Zvanična FTN uputstva za završne radove

- Pravilnik o završetku studija i sticanju zvanja: [ovde](#)
- Procedura prijave i obrane završnog (bachelor) rada: [ovde](#)

3. Uputstvo za pisanje i formatiranje teksta

Tekst rada je potrebno napisati i formatirati prema uputstvu datom u [ovom](#) Word fajlu. Ovo uputstvo sadrži parametre za formatiranje teksta, ali i smernice za stil pisanja i strukturu rada. Kada se ovako formatiran tekst odštampa dvostrano, on može poslužiti za koričenje u B5 formatu uz sečenje papira duž velikih margina.

Uputstvo za pisanje i formatiranje teksta se odnosi na sve vrste radova (bachelor, master, diplomski).

Novi format naslovne strane dat je [ovde](#).

Novi format stranice sa zadatkom rada dat je [ovde](#).

4. Prpratni materijal uz diplomski rad

Izvorni kod

Izvorni kod treba da se nalazi u GitHub privatnom repozitorijumu, gde je autor rada jedini *contributor*. Izuzetak može biti kolega iz tima u slučaju da se diplomski radov bazira na predmetu projekat, ali grana koja je dodatak za diplomski rad treba da bude sačinjena samo od promena autora rada. Dodati mentora na repozitorijum. Repozitorijum mora imati *Readme.md* popunjen gde će jasno biti objašnjeno kako se pokreće i postavlja rešenje tako da može lako da se pokrene i testira. Licenca na repozitorijumu mora biti MIT licence [*keyword: mit*]. Uputstvo za licenciranje se nalazi [ovde](#).

Slike, grafici i ostali izvorni materijal

Izvorne slike koje su delo autora treba da se predaju uz diplomski rad. Poželjno je da tekst na slikama bude font 12, na srpskom jeziku. Slike koje se prikazuju u samom radu treba da budu rezolucije od 300dpi.

5. Opšti principi izrade završnog rada

1. Završni BSc rad predstavlja zaokruženu celinu koja, kada god je to moguće, predstavlja proširenje rezultata koje je student ostvario kroz projekte u toku studiranja.

2. Teme za završne BSc radove obuhvataju implementaciju konkretnih softverskih rešenja, sa fokusom na istraživanju novih softverskih tehnologija i alata, kao i njihovim korišćenjem za razvoj softvera.
3. Konkretnu temu za završni BSc rad student može da odabere na sledeći način:
 - a. Da sam predloži temu koja nije na listi ponuđenih uz obrazloženje motivacije i predlog sadržaja završnog rada.
 - b. Da izabere temu sa ponuđene liste (vidi Aktuelne teme).
4. Način izrade završnog BSc rada:
 - a. U izradi završnog BSc rada, student sarađuje sa predmetnim nastavnikom koji je formalni mentor rada i opciono asistentom/asistentima koje mentor odredi za saradnju sa studentom.
 - b. Aktivnosti pri izradi završnog BSc rada obuhvataju
 - i. Prikupljanje literature relevantne za rešavanje zadatka završnog rada.
 - ii. Analizu literature i prikaz stanja u oblasti kojoj pripada zadatak završnog rada.
 - iii. Definisanje rešenja zadatka koji je predmet završnog rada.
 - iv. Analizu predloženog rešenja koja obuhvata prednosti i ograničenja rešenja i definiše pravce daljeg rada kojima bi se umanjila uočena ograničenja.
 - c. Struktura teksta završnog rada organizovana je tako da prati aktivnosti pri izradi završnog BSc rada.
 - d. Tekst završnog rada je formatiran u potpunosti u skladu sa šablonom koji je studentima dostupan u elektronskoj formi.
5. Vreme za izradu završnog BSc rada – Studenti se ohrabruju da završni BSc rad urade u što kraćem roku.

6. Aktuelne teme

Aktuelne teme su one koje se mogu uraditi dodavajući funkcionalnost ili primenom novih pristupa nad rešenjem razvijenim na PUSGS predmetu i predmetu Projekat. Napomena, teme ne predstavljaju finalni naslov teze. Naslov se piše na kraju tako da ima između 5 i 10 reči, a da najpribližnije oslikava ideju i doprinos rada.

Tema 1: Verzionisanje mikroservisa i automatska provizija novih rešenja

Ključne reči: Mikroservisi, kontejneri, automatizacija

Zadatak: Izvršiti pregled literature, identifikovati moguće pravce implementacije, obrazložiti odluku i implementirati rešenje u kom je proces promocije nove verzije mikroservisa u produkcionom okruženju potpuno automatizovan i omogućava postepen prelazak sa starije na noviju verziju servisa. Takođe, razmotriti opcije i vraćanja na staru verziju ukoliko je neophodno iz nekog razloga. Ovu temu je moguće objediniti i sa temom oporavka više mikroservisa iz arhiva (*backup/restore*). Dokumentovati datu proceduru, i obavezno adresirati Docker i Kubernetes kao moguću/glavnu opciju.

Tema 2: Revizija bezbednosti i implementacija bezbednosnih mehanizama u *web* sistemima

Ključne reči: Bezbednost, GDPR, mikroservisi

Zadatak: Izvršiti pregled literature, identifikovati moguće pravce implementacije sveobuhvatne softverske bezbednosti, izučiti modelovanje pretnji. Analizirati postojeće rešenje u pogledu OWASP najčešćih ranjivosti za *REST* servise kao i za *web* aplikacije. Implementirati nedostatke.

Identifikovati gde se nalaze podaci koje je neophodno štititi na osnovu zahteva GDPR uredbe, i predložiti mehanizam njihove zaštite.

Tema 3: Implementacija bezbedne podele resursa sa postojećeg *web* sistema

Ključne reči: Bezbednost, REST, OAuth 2.0

Zadatak: Izvršiti pregled literature, identifikovati moguće pravce implementacije autorizacije nad postojećim resursima za koje je odgovoran postojeći *web* sistem. Izučiti modelovanje pretnji i primeniti ga na podelu resursa. Omogućiti dozvolu pristupa postojećim resursima, i omogućiti korisniku granulaciju dozvola nad postojećim resursima. Istražiti uputstva za logovanje operacija po OWASP preporukama i upotrebiti ih na mestima gde se iz spoljnog sistema pristupa resursima.

Tema 4: Optimizacija prednje strane koristeći mehanizme Angular okvira

Ključne reči: Angular, optimizacija, front-end arhitektura

Zadatak: Izvršiti pregled literature, identifikovati moguće pravce implementacije sveobuhvatne softverske bezbednosti. Analizirati postojeće rešenje u pogledu mogućnosti modularizacije. Primeniti lenjo učitavanje nad modulima. Izučiti i implementirati *ngrx* i *effects* mehanizme u okviru Angular okvira. Implementirati automatsko testiranje u cilju otkrivanja grešaka u optimizaciji rešenja.

Tema 5: Optimizacija prednje strane koristeći *GraphQL* mehanizme

Ključne reči: Angular, optimizacija, front-end arhitektura, *GraphQL*

Zadatak: Izvršiti pregled literature, identifikovati moguće pravce implementacije oslanjajući se na *GraphQL* pristup. Analizirati postojeće rešenje u predložiti mogućnosti migracije na *GraphQL*. Potom implementirati. Implementirati automatsko testiranje u cilju otkrivanja grešaka u optimizaciji rešenja.

Tema 6: Migracija monolitnog sistema na distribuiranu mikroservisnu arhitekturu

Ključne reči: Mikroservisi, optimizacija, back-end arhitektura, saga

Zadatak: Izvršiti pregled literature, identifikovati moguće podele postojećih servisa na mikroservisne koristeći metode dostupne u literaturi. Implementirati podelu. Upotrebiti *Saga* šablon na mestima gde je potrebna distribuirana transakcija.

Tema 7: Migracija monolitnog sistema na mikroservisnu arhitekturu baziranu na događajima

Ključne reči: Mikroservisi, optimizacija, back-end arhitektura, event sourcing, *cqrs*

Zadatak: Izvršiti pregled literature, identifikovati moguće podele postojećih servisa na mikroservisne koristeći metode dostupne u literaturi. Implementirati podelu. Upotrebiti *event sourcing* šablon kako bi se implementirala arhitektura bazirana na događajima i predstaviti par slučajeva gde ona donosi prednosti. Razmotriti upotrebu *CQRS* šablona i implementirati ga na neophodnim mestima.

Tema 8: Iscrtavanje putanja u realnom vremenu koristeći WebGL tehnologiju

Ključne reči: WebGL, 2D vizuelizacija

Zadatak: Izvršiti pregled literature, identifikovati moguća mesta gde bi iscrtavanje nad HTML5 canvas objektom dovelo do boljeg korisničkog iskustva. Napraviti biblioteku za iscrtavanje informacija u realnom vremenu (putanje na mapi i dodatne tekstuelne informacije). Koristeći WebSocket tehnologiju, implementirati iscrtavanje u realnom vremenu pri čemu će funkcionalnosti uveličavanja biti omogućene.

Tema 9: Implementacija observabilnosti mikroservisne arhitekture

Ključne reči: Mikroservisi, monitoring, DevOps

Zadatak: Izvršiti pregled literature, identifikovati moguće podele postojećih servisa na mikroservisne koristeći metode dostupne u literaturi. Implementirati podelu. Upotrebiti šablone poput *Health check API*, *Log aggregation*, *distributed tracing*, *application metrics*, *exception tracking* i *audit logging* kako bi se implementirali mikroservisi sa potpunom vidljivošću u realnom vremenu. Implementirati stranicu za pregled ili adapter za postojeća monitoring rešenja.

Tema 10: Implementacija mikroservisne arhitekture oslanjajući se na mikroservisne šasije

Ključne reči: Mikroservisi, monitoring, DevOps

Zadatak: Izvršiti pregled literature, identifikovati moguće podele postojećih servisa na mikroservisne koristeći metode dostupne u literaturi. Implementirati podelu. Istražiti i uporediti platforme za mikroservise kao što su *Istio*, *Linkerd*, *Conduit* i implementirati rešenje koristeći jednu od ponuđenih. Analizirati iz ugla funkcionalnosti koje donose i prikazati način implemetnacije.

U narednom poglavlju se nalaze smernice za strukturu rada. One su primenljive na aktuelne teme, a verovatno i na teme predložene od strane studenta. Smernice su tu da pomognu u pisanju rada, a nikako ne služe kao strogo pravilo koje se treba poštovati.

Smernice za strukturu rada

Rad se sastoji grubo od sledećih celina (u nastavku okvirno dat procenat koliko treba da se posveti teksta datom poglavlju u odnosu na ceo rad):

1. Uvod – 5%
2. Opis problema (može biti i potpoglavlje u uvodu) – 5%
3. Opis korišćenih tehnologija – 10%
 - a. Opis same tehnologije, ali sa fokusom na ono što je zaista važno za rešenje u samom radu: npr. *NgRx* i *effects*, a ne opisivati ceo *Angular* framework
 - b. Kratak opis svih komponenti koje učestvuju u rešenju. Npr. *Angular* na prednjoj strani, na zadnjoj *Kubernetes*, *Docker*, *.NET*, *NodeJS* itd.
4. Opis trenutnog rešenja – 30%
 - a. definisati sve učesnike u sistemu i njihove uloge
 - b. definisati slučajeve korišćenja sistema
 - c. definisati model podataka
 - d. definisati tok podataka i arhitekturni pogled na rešenje
 - e. definisati probleme koje BSc rad treba da reši
 - f. Napomena: ovde opisivati ono što je student konkretno realizovao u timskom projektu. Staviti fokus na stvari koje su bitne iz ugla problema koji se rešava kao i opisa rešenja problema.
5. Opis rešenja problema – 35%
 - a. Jako zavisi od teme. Razdvojiti na poglavlja tako da čitaocu bude što jasnije šta je urađeno, šta je unapređenje, i zašto je autor doneo neke odluke koje je doneo.
 - b. Može biti dobar pristup da se unapređenja opisuju tako što se referencira na opis postojećeg rešenja (tačka 4), i potom prikažu unapređenja.
6. Zaključak – 15%
 - a. rezime onoga što je urađeno
 - b. karakteristike dobijenog rešenja
 - c. pravci daljeg razvoja
7. Literatura

7. Proces prijave

Uloga	Opis	Ime	kontakt
Student	Student koji diplomira iz oblasti PUSGS predmeta		
Mentor	Profesor na PUSGS predmetu	Nikola Dalčeković	Videti ovde
Asistent	Asistenti na PUSGS predmetu	Dušan Marjanski, Nenad Gligorov, Marko Arambašić, Marijana Matkovski, Ivana Dalčeković	Videti ovde

Mentor objavljuje rokove za prijavu izrade diplomskog (BSc) rada putem SOVA platforme, na sajtu PUSGS predmeta u vidu obaveštenja. **Student** se može prijaviti elektronskom poštom do navedenog roka, pri čemu treba da pojasni svoju temu ukoliko ima predlog, ili da navede listu želja od postojećih tema.

Nakon navedenog roka, **mentor** objavljuje listu **studenata** sa dodeljenim temama koristeći sledeće kriterijume:

1. Svaka tema može imati maksimalno dva studenta sa istim početnim rešenjem (npr. tema 1 na primeru projekta *rent-a-car*).
2. Kriterijum odabira studenta: student sa više osvojenih poena iz predmeta PUSGS ima prednost za dobijanje željene teme. Ukoliko studenti imaju isti broj poena, temu dobija student koji se prvi prijavio.

Maksimalni broj **studenata** koje **mentor** može da prihvati u kvartalu je 40. Nakon četrdesetog studenta, sledeći rok za prijave je za tri meseca od prvog roka za prijave. Ovo se radi kako bi **studenti** imali kvalitetnije posvećeno vreme od strane **mentora**. Ukoliko je **studentima** previše da čekaju tri meseca, mogu se odlučiti za izradu BSc rada na drugom predmetu.

Ukoliko se ukaže potreba, **mentor** može odrediti jednog ili više **asistenata** da pomažu određenom **studentu** vezano za izradu radova.

Obično je prvi rok objave prijava za teme prva nedelja nakon završenog semestra.

8. Očekivan nivo mentorstva

Uzimajući u obzir dozvoljena odstupanja od oko 30%, smernica za očekivanu veličinu završnog rada je: ~8.000 *words*. *Words* metrika se može videti u *Word* procesoru, i odnosi se na tekst bez naslovne strane, sadržaja i teksta ključne dokumentacije.

Finalna verzija rada se šalje mentoru putem elektronske pošte u .docx formatu. Očekivani rok za reviziju je **između 10 i 20** kalendarskih dana. U zavisnosti od revizije, primena sugestija može potrajati. Iz navedenih razloga, očekuje se da **student** pošalje prvi kandidat finalnog rada **minimalno 30 dana** pre planiranog termina odbrane.

Nakon procesa prijave, a pre slanja finalnog rada, student se može konsultovati sa mentorom na sledeći način:

1. Kada **student** prouči temu, može pre početka pisanja rada i implementacije softverskog rešenja zatražiti konsultacije kako bi sa **mentorom** proverio razumevanje i bio siguran da je na dobrom putu izrade teze. Ovo nije obavezan korak. Nakon održanih konsultacija, važe koraci dva i/ili tri.

2. Slanjem konkretnog pitanja vezanog za izradu rada, po potrebi.
3. Slanjem zahteva za konsultaciju, koji treba da sadrži izučavanu literaturu (može biti i reč o manje formalnim izvorima znanja poput članka na internetu) koju je student prethodno istražio i ima nedoumice. U zavisnosti od pitanja, mentor može odgovoriti na *e-mail* ili predložiti termin konsultacije u trajanju do 30 minuta.

9. Najčešća unapređenja

U ovoj sekciji se nalaze odgovori na najčešće greške ili prostor za unapređenja izdvojena na osnovu dosadašnjeg rada na diplomskim radovima.

9.1 Literatura

Brojeve referenci postaviti u uglaste zagrade [1]. Reference navoditi redosledom kojim se pojavljuju u tekstu i ne uključivati reference koje nisu pomenute u tekstu rada.

Radovi objavljeni u časopisima navode se na sledeći način:

[1] Autor(i), Naziv rada, Naziv časopisa, Volumen, broj (godina izdanja) strana x–y.

Radovi objavljeni u zbornicima:

[2] Autor(i), Naziv rada, Naziv zbornika, Mesto izdavanja godina, strana x–y.

Knjige se navode na sledeći način:

[3] Autor(i), Naziv knjige, Izdavač, Mesto izdavanja godina, str. x–y

Obratite pažnju: reference se navode u skladu sa IEEE stilom (za više informacija videti [ovde](#)) kako su iznad dati i primeri relevantni za sam diplomski rad.

Predlog je da se koristi [Google Scholar](#) na sledeći način:

1. Pretražiti knjigu, rad ili izvor koji je od interesa. Idealno je da se preko *google scholar* servisa zaista i pronađu izvori iz kojih će se naučiti i izdvojiti ono što je od interesa za sam diplomski rad.
 - a. Wikipedia i ostali nepozdani izvori ne treba da se nađu u literaturi diplomskog rada.
2. Pored rezultata koji je od interesa, kliknuti na link *cite*.
3. Odabrati IEEE za stil citata.
4. Prekopirati citat u diplomski rad, u sekciju literatura.

Najčešća greška je da se u samom tekstu uopšte ne referencira literatura, navođenjem izvora pored izjave [1]. U prethodnoj rečenici je dat samo primer na kraju rečenice kako bi u tekstu izgledalo referenciranje, što bi značilo da je zaključak ili znanje iz prethodne rečenice stečeno iz literature navedene pod rednim brojem 1. Poželjno je što više izjava referencirati. Praktično, sve izjave koje se preuzimaju iz drugih izvora, a nemaju adekvatnu referencu su u suštini plagijarizam.

Za one koje zanima naprednije upravljanje literaturom u smislu čuvanja i izmene referenci, kao i traženje literature, prepuka je da se koristi besplatni servis, *mendeley.com*. *Mendeley* čak sadrži i *plugin* instalacije za *word* alat, s tim da je *mendeley* svakako van fokusa za pripremu diplomskih radova i neće biti detaljnije razjašnjava.

9.2 Slike izvornog koda

Izvorni kod nikada ne treba prikazivati kroz slike (engl. *screenshot*). Obično se delovi koda prikazuju kroz listinge, koji treba da su numerisani, kao i eksplicitno referencirani u tekstu, što važi i za slike i tabele. Font koji se koristi je *consolas*, ali u sustini može ostati i format kada se uradi kopiranje iz *Visual Studio* alata. Tekst omogućava lakše izmene na kodu, pretragu i kvalitet štampe ne zavisi od rezolucije slike. Takođe, veličina teksta ne zavisi od toga kako smo sliku skalirali.

Dodatna napomena: listing bi trebao da sadrži samo delove koda koji su od interesa za suštinu rada. Nikako ne treba raditi kopiranje koda koji se ne može dovesti u direktnu vezu sa suštinom koja se u radu nalazi, tj. ne treba kopirati nepotreban kod.

9.3 Odnos teksta i slika

Iako nije pravilo, mnogo je informativnije koristiti slike umesto samog teksta. U mnogo radova su dobro opisani slučajevi korišćenja, dobro objašnjena arhitektura i određeni delovi koda, tako da, iako se mogu sve važne informacije pronaći u tekstu, često je teško doći do informacije koja je od interesa u samom trenutku. Dodatno, teško je steći utisak o radu na osnovu brzinskog prolaska kroz dokument, što je često ideja nekoga ko bi razmatrao da uopšte čita određeni rad. U nastavku su dati primeri kakvi se sreću u diplomskim radovima, i njihove alternative:

1. Detaljno opisani slučajevi korišćenja – *use case* dijagrami,
2. Detaljan opis arhitekture – arhitektura nacrtana u *visio* alatu ili drugom alatu po izboru,
3. Izdvojeni opisi važnih komponenti ili delovi koda, interfejsi itd. – *UML* dijagrami ili se mogu pokriti kao u tački 2. uz dodatne listinge ukoliko je potrebno pokazati određenu implementaciju
4. Bitne izmene u samom domenu i njihov opis – ER model dijagrami.

Ako se u radu i nađu slike koje opisuju širi kontekst, to ne znači da sam tekst treba da izostane. Potrebno je i dalje opisati te slike i dati informacije koje se naravno ne mogu pronaći na slici. Može se zaključiti da bi slike dale preglednost dobro opisanoj suštini rada za lakše čitanje rada, kao i lakše vraćanje na suštinu ukoliko u kasnijem delu teksta se javi potreba da se čitalac podseti neke informacije.

9.4 Engleske reči

Engleske reči se pišu uvek kurzivom ako je nemoguće prevesti reč na srpski. Obično iz navike smatramo da ne možemo prevesti reči, ali u velikoj većini možemo i trebamo to da uradimo. Neki od primera:

1. *Cloud* – oblačno računarstvo,
2. *Endpoint* – metode servisa (može se drugačije prevesti, zavisno od konteksta),
3. *Publish/Update/Upgrade/Patch* – postavljanje nove verzije/zakrpe,
4. *Runner/Executor* – izvršavanje, izvršno okruženje (zavisno od konteksta),
5. *Node* – čvor,
6. *Middleware* – srednji sloj,
7. *Host* – računar, ali može zavisiti od konteksta rečenice. Najbezbednije je kada se prvi put prevodi da stoji u zagradi originalna reč, npr. *virtuelna mašina se izvršava na fizičkom računaru (engl. host)*.

Imena kompanija se mogu pisati na srpskom jeziku, dok bi se prvi put u zagradi navelo pravo ime. Primer: *Gugl* (engl. *Google*).

Često menjanje engleskih reči po padežima treba izbegavati. Npr. inženjeri *Google*-a su zaključili, ili slično. Koristiti srpski: inženjeri Gugla (engl. *Google*) su zaključili.

9.5 Aktivno i pasivno pisanje

U radu se često sreće mešanje stilova pisanja u smislu da se koristi pisanje u stilu uputstava gde se koriste lični glagolski oblici (npr. pokazaćemo, videćemo), a kasnije bezlični (može se videti, može se zaključiti). Oba su pravilna, dok se u srpskom kod pisanja diplomskih radova preporučuje korišćenje bezličnih glagolskih oblika. U svakom slučaju, stilovi se ne smeju mešati, mora biti uniforman stil.

9.6 Ključna dokumentacija, predlozi za unapređenja

Polje izvod (IZ): vrlo kratko, do 100 reči, potrebno je opisati uvod u kom se objašnjava sam rad i daje širi kontekst šta je početno rešenje, a šta se pokušava postići, sa akcentom na to zašto je rad važan, što je u slučaju diplomskih u stvari glavni fokus samog rada. Potrebno je u opisu da postoji svrha rada, tj. šta je ono što je ključno za sam diplomski rad. Na kraju izdvojiti zaključak autora do kog je došao nakon istraživanja i implementacije softverskog sistema za potrebe diplomskog rada.

Sledeća dva polja se popunjavaju ovako:

Научна област, НО :	Електротехника и рачунарство
Научна дисциплина, НД :	Примењени софтверски инжењеринг

Popunjavanje komisije:

Potrebno je pogledati na ftn.uns.ac.rs titulu profesora i pravilno je navesti. Od koristi je znati sledeće prevode:

1. docent je *assistant professor*
2. vanredni profesor je *associate professor*
3. redovni profesor je *full professor*
4. dr je *PhD*

Format zapisa na srpskom: dr Ime Prezime, titula (docent ili vanredni profesor ili redovni profesor)

Format zapisa na engleskom: *Full name, PhD, title (assistant professor or associate professor or full professor)*

Ne treba ulaziti u proces štampanja rada pre nego što mentor potvrdi ključnu dokumentaciju, kao i celokupan rad.

9.7 Poravnanje teksta

Tekst treba da bude poravnat po obe margine (engl. *justified*). Iako je naznačeno u šablonu, često ovo ne bude slučaj u prvim verzijama diplomskih radova.

9.8 Skraćenice

Uvek se pri uvođenju zagrade piše punim tekstom dok se skraćenica uvodi u zagradi. Nakon toga je u redu koristiti samo skraćenicu. Primer: *Data Transfer Object* (DTO) ... DTO se koristi...