VELEUČILIŠTE U BJELOVARU

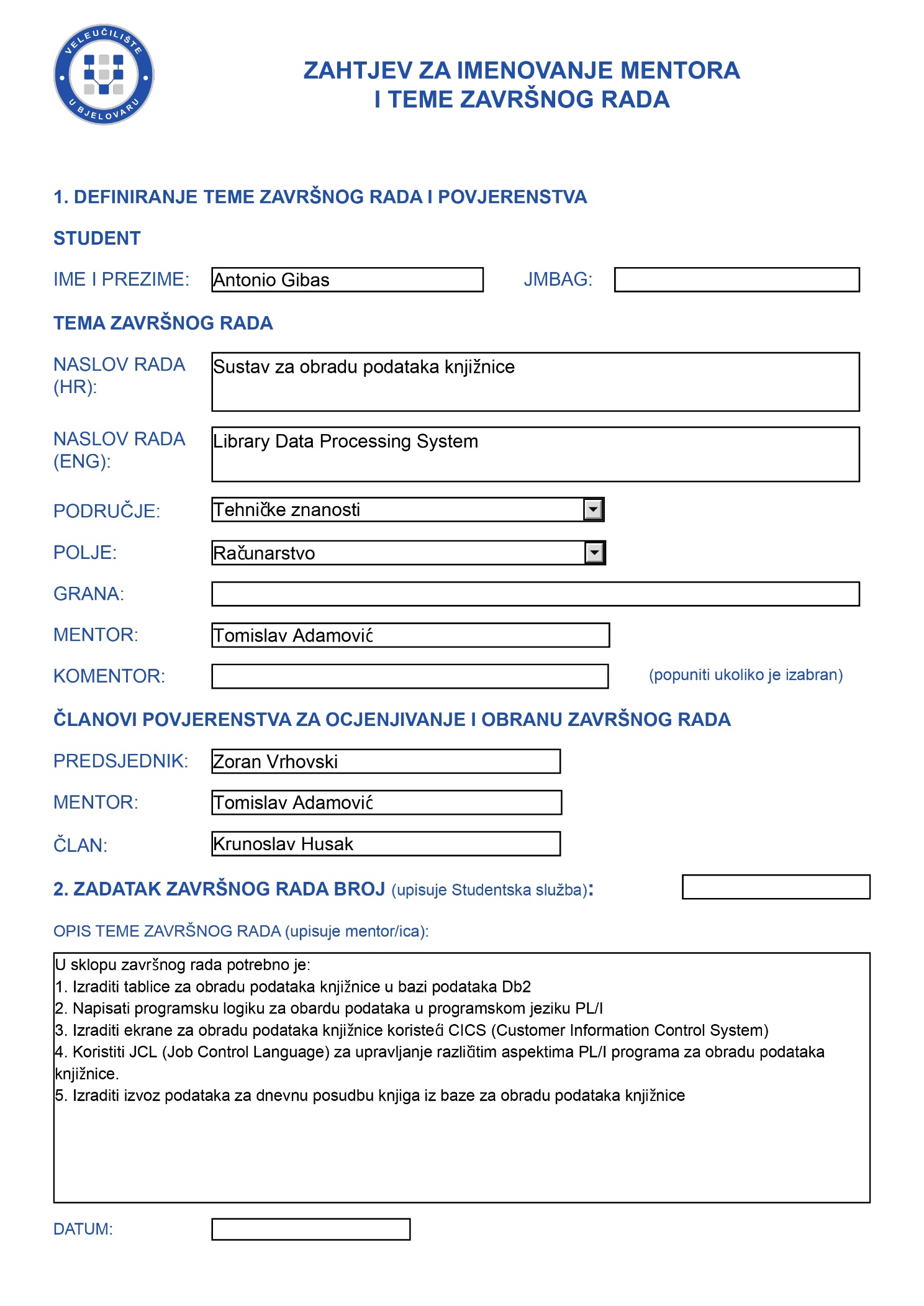
STRUČNI PRIJEDIPLOMSKI STUDIJ MEHATRONIKA

**SUSTAV ZA OBRADU PODATAKA KNJIŽNICE**

Završni rad br. XY/RAČ/GGGG

Antonio Gibas

Bjelovar, Svibanj 2024.



*Zahvala*

Zahvaljujemo svim studentima Veleučilišta u Bjelovaru koji će se pri pisanju svojeg završnog rada pridržavati *Uputa o pisanju završnog rada* i koristiti ovaj predložak. Ukoliko ne želite napisati zahvalu, ovu stranicu ostavite praznu.

Sadržaj

[1. Uvod 1](#_Toc179383550)

[2. SUSTAV ZA OBRADU PODATAKA KNJIŽNICE 2](#_Toc179383551)

[2.1 Poslovna logika 3](#_Toc179383552)

[3. OPIS TEHNOLOGIJA KORIŠTENIH U IMPLEMENTACIJI 4](#_Toc179383553)

[3.1 Operacijski sustav z/OS 5](#_Toc179383554)

[3.2 Job Control Language 7](#_Toc179383555)

[3.3 Db2 Baza podataka 9](#_Toc179383556)

[3.4 PL/I programski Jezik 10](#_Toc179383557)

[3.5 CICS upravitelj transakcija 11](#_Toc179383558)

[4. PROCES IZRADE SISTEMA 12](#_Toc179383559)

[4.1 Izrada Db2 tablica 14](#_Toc179383560)

[*4.1.1 Tablica KNJIGE 16*](#_Toc179383561)

[*4.1.2 Tablica CLANOVI 18*](#_Toc179383562)

[*4.1.3 Tablica VEZNA1 19*](#_Toc179383563)

[*4.1.4 Tablica AUTORI 21*](#_Toc179383564)

[*4.1.5 Tablica REZERV 22*](#_Toc179383565)

[*4.1.6 Tablica DOBAVLJ 24*](#_Toc179383566)

[*4.1.7 Tablica NABAVA 25*](#_Toc179383567)

[*4.1.8 Tablica POSUDBA 27*](#_Toc179383568)

[*4.1.9 Tablica ZAKAS 29*](#_Toc179383569)

[4.2 Postavljanje PL/I, CICS i SQL kompajlera 31](#_Toc179383570)

[4.3 Izrada CICS ekrana za 3270 terminale 31](#_Toc179383571)

[4.4 Analiza i razvoj PL/I aplikacija 31](#_Toc179383572)

[4.5 Kompiliranje aplikacija 31](#_Toc179383573)

[4.6 Kreiranje, instaliranje i testiranje transakcija 31](#_Toc179383574)

[5. MOGUĆA POBOLJŠANJA SUSTAVA 31](#_Toc179383575)

[6. ZAKLJUČAK 33](#_Toc179383576)

[7. LITERATURA 34](#_Toc179383577)

[8. OZNAKE I KRATICE 37](#_Toc179383578)

[9. SAŽETAK 39](#_Toc179383579)

[10. ABSTRACT 40](#_Toc179383580)

[11. PRILOZI 41](#_Toc179383581)

Kada je rad gotov, tablicu sadržaja **treba ažurirati** i provjeriti nalaze li se u njoj **svi naslovi** koji se nalaze u radu.

­

# Uvod

Knjižnice predstavljaju nezamjenjive institucije za pristup znanju, informacijama i literaturi. no često se suočavaju s izazovima upravljanja velikih količina knjiga i ostalih fizičkih informacijskih medija. Tradicionalni sustavi za obradu podataka u knjižnicama mogu biti spori i nepouzdani, što otežava učinkovito upravljanje zbirkama, posudbama, te informacijama o korisnicima. Svrha ovog završnog rada je implementacija sustava za obradu podataka knjižnice koristeći PL/I programski jezik s CICS-om (Customer Information Control System), te Db2 bazu podataka za pohranu podataka knjižnice. Učinkovit sustav za obradu podataka i sama brzina sustava su presudne u kvaliteti poslovanja svake institucije čiji je glavni cilj optimizirati svoje poslovanje i usluge korisnicima.

Struktura rada je podijeljena na generalni tehnološko opisni uvod uz opis poslovne logike, uz opis svake korištene tehnologije i metodologije, te osnovni razlog odabira ovakvog načina implementacije u poglavljima „Uvod“, „Sustav za obradu podataka knjižnice“. Popratno opisima, ovaj rad će ići u detaljniji opis implementacije relevantnih navedenih teorijskih koncepta korištene tehnologije na praktičnom primjeru primjene u poglavlju „Proces izrade sistema“.

Slika na kojoj se prikazuje namještaj, stol, u dvorani, stolica

Opis je automatski generiran

Slika 1. Knjižnica (generirano s DALL-E modelom)

# SUSTAV ZA OBRADU PODATAKA KNJIŽNICE

Sustav za obradu podataka knjižnice osmišljen je korištenjem CICS ekrana, prikazanih na terminalu 3270-2, za reprezentaciju poslovne logike. Svaki ekran kreiran je pomoću CICS Asembler makro jezika, koji se piše planski prema potrebama izgleda sučelja za interakciju s korisnikom. Kompilirani Asembler kod tvori modul koji se izvodi CICS u „Application Owning Region“ (AOR) [1]. Taj isti modul se naziva „fizička mapa“. Fizička mapa dolazi u enkodiranom formatu, tipa EBCDIC [2]. Također, uz fizičku mapu, automatski je kreirana i simbolička mapa. Simbolička mapa je jezično-specifična podatkovna struktura, reprezentirana varijablama i tipovima podataka programskog jezika u kojem kodiramo poslovnu logiku i CICS naredbe [2]. Preko varijabli simboličke mape se prenose podaci korisniku na terminal, prosljeđuju podaci između transakcija i spremaju u bazu podataka.

Korisnička interakcija na svakom setu mapa je uvjetovana povratnom logikom CICS sustava programu koji se izvodi u transakciji. Sve transakcije u ovom radu su uvjetovane naredbama korisnika, te iste unutar transakcije ostavljaju povratni kod CICS-u. Podatkovno polje koje skuplja informacije o samom izvođenju transakcije se naziva „Execute Interface Block“ (EIB). Aplikacijski program pristupa EIB-u tijekom izvođenja transakcije [3]. Ovisno o EIB-u, aplikacijska logika se baždari po poželjnom i nepoželjnom korisničkom unosu. Tipke na tipkovnici imaju svoj simbol definiran na kodnoj stranici samog z/OS operacijskog sustava, te njihov izlazni povratni kod služi za uvjetovanje izvođenja transakcije. Korisnik prilikom rada na sustavu, ključnom riječi inicira se transakcija, koja prikazuje izbornik, iz tog izbornika korisnik može odabrati poželjnu poslovnu logiku koja mu zatreba u tom trenutku odvijanja posla. Svaki korisnički upit o prikazu inventara knjiga, narudžbi, dobavljača, članova i bivših članova šalje upit preko CICS-a na Db2 bazu podatka. Upit se odvija preko Db2 adresnog prostora za komunikaciju sa servisima [4]. Logika slanja, dohvata i ažuriranja podataka iz Db2 baze podataka, realizirana je u samom PL/I programu putem SQL upita. SQL upite u ovakvom sustavu izvršava „CICS Db2 Attachment Facility“ koji je pokrenut kada god aplikacijski program unutar transakcije izvrši SQL upit [5].

Ovakvim pristupom korisniku je omogućen nesmetan pristup svim informacijama o raspolaganju poslovanja knjižnice kao ustanove.

## Poslovna logika

U radu knjižnice istaknuo je nekoliko ključnih operacija potrebnih za njezino učinkovito funkcioniranje.

Među najvažnijima su:

1. Upravljanje članovima

* Učlanjenje i registracija novih korisnika
* Ažuriranje podataka korisnika
* Upravljanje članarinama
* Ažuriranje podataka korisnika
* Ukidanje članarine korisnika

1. Upravljanje knjižnim fondom

* Unos novih knjiga
* Ažuriranje podataka knjiga
* Katalogizacija knjiga prema temama, žanru i drugim kriterijima

1. Proces posudbe i vraćanja knjiga

* Posudba knjiga
* Vraćanje knjiga
* Produljenje posudbe
* Rezervacija posudbe knjiga

1. Nabava knjiga

* Odabir i narudžba
* Evidencija nabave
* Unos podataka nabavljača

1. Izvještaji

* Izvještaji o posudbama
* Izvještaji o korisnicima
* Inventura
* Analiza potreba

Većina temeljnih poslovnih potreba rada knjižnice može se realizirati implementacijom baze podataka, čije tablice su korespondentne s navedenim poslovnim logikama. Navedene poslovne logike se implementiraju tako da ostane mjesta za dodatne izmjene u budućnosti, te je prilagodljiva realističnijim potrebama rada.

# OPIS TEHNOLOGIJA KORIŠTENIH U IMPLEMENTACIJI

Implementacija sustava za obradu podataka knjižnice koristi niz tehnologija koje

omogućuju efikasnu i sigurnu obradu podataka, uz korisničko predznanje kako koristiti dizajnirani sustav pri svom radu. U ovom poglavlju detaljno je opisana korištena tehnologija u dizajnu i implementaciji sustava, uključujući CICS [6], Db2 bazu podataka [7] i programski jezik PL/I [8], na operacijskom sustavu z/OS [9].

Nadvedene tehnologije za sustav knjižnice odabrane su zbog brzine i pouzdanosti, dok u isto vrijeme zahtijevaju detaljno poznavanje funkcija rada na sustavu i ne pružaju dobru korisničku interaktivnost. Zbog tih razloga, CICS terminalni ekrani izrađeni su da pružaju samo-objašnjivu funkcionalnost sa strane korisničke interakcije.

Prikaz arhitekture završnog rada opisan je pomoću veza tehnologija bez poslovne logike. U slici ispod, prikazan je međuodnos komponenti na z/OS operacijskom sustavu koje omogućuju korisničku komunikaciju s bazom podataka preko CICS-a.

Slika na kojoj se prikazuje tekst, snimka zaslona, dijagram, Trokut

Opis je automatski generiran

Slika 2 - Vizualizacija arhitekture komponenata završnog rada.

## Operacijski sustav z/OS

Enterprise z/OS je skalabilno – pouzdan operacijski sustav koji se primarno koristi za rad na IBM Mainframe računalima. IBM Z je operacijski sustav dizajniran za kontinuirano odvijanje poslovne logike i transakcija u svijetu [10]. Prvo izdanje softwarea izdala je kompanija IBM u Prosincu 2000. godine [11], za „Z“ seriju Mainframe računala. Operacijski sustav z/OS podržava rad s velikim brojem istovremenih korisnika, uz obradu velikih količina podataka s visokom učinkovitošću. Takva svojstva čine ga neophodnim u institucijama kao što su banke, bolnice, poštanski servisi, ministarstva, vojske i sl.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Font

Opis je automatski generiran

Slika 3 Arhitektura z/OS operacijskog sustava

Operacijski sustav se pokreće u logičkoj particiji LPAR u Mainframe-u. LPAR je određeni podskup Mainframe resursa na jednom sistemu. Moguće je imati do 85 aktivnih logičkih particija, svaka s različitim definicijama i konfiguracijama operacijskih sustava [10]. Mainframe hardver se sastoji od procesora i uređaja za zapis podataka (diskovi, trake, DASD), magnetni diskovi i više raznih vrsta korisničkih konzola za rad i upravljanje [12]. Operacijski sustav koristi procesorsku memoriju (RAM) tijekom izvođenja kako bi obavljao svoje funkcije i upravljao resursima sustava [12].

Jedni od najbitnijih dijelova operacijskog sustava z/OS su: sistemski servisi, komunikacijski servisi, aplikacijski servisi, servisi za podršku aplikacijama, distributni sustavi, sigurnosni kriptografski servisi i podrška za mrežu. Sistemski servisi služe kao potpora za cijeli operacijski sustav, u njima se odvija aplikacijski sloj, UNIX servisi, „batch“ obrade i podrške za komunikaciju s bazama. Aplikacijski servisi služe za kontrolu kreiranja, izvođenja i obradu svih sistemskih i novih aplikacija. Komunikacijski servisi omogućuju umreženost svim korisnicima, operatorima, programerima na sustavu. Sam sustav podržava velik broj korisnika koji rade na svojoj instanci sučelja sustava. Po svakoj z/OS Mainframe instalaciji, sustav podržava istovremeni rad s nekoliko stotina do nekoliko tisuća korisnika istovremeno, ovisno o hardverskim specifikacijama [9]. Distributni servisi omogućavaju podršku za pristupanje distribuiranim aplikacijama pomoću DFS-a. Sigurnosni servisi osiguravaju čitav z/OS sustav, svaki korisnik na operacijskom ustavu ima svoj određeni set prava rada na istom. Servisi sa slike 2, omogućuju z/OS operacijskom sustavu neprekidni rad i visoku dostupnost 24 sata u danu. Sve aplikacije na sustavu rade nesmetano, što je esencijalno organizacijama koje ovise o kontinuiranim obradama i dostupnosti podataka [9]. Upravljanje sustavom obuhvaća korištenje alata za pregled spool[[1]](#footnote-2)-a i administraciju.

Kroz alate kao što su SDSF, JES i RACF, sistemski administratori mogu osigurati stabilan i učinkovit rad z/OS operacijskog sustava, brzo reagirati na potencijalne probleme i prilagodbu sustava prema zahtjevima posla ili korisnika. Arhitektura, sigurnosne značajke, integracija s drugim servisima i alati za administraciju čine z/OS potpuno neophodnim za mnoge organizacije koje zahtijevaju stabilno i sigurno odvijanje posla na sistemu.

## Job Control Language

Job Control Language, skraćeno (JCL), je specijalizirani skriptni jezik dizajniran za pokretanje i odvijanje JOB-a (skripti) u z/OS-u. Ovaj jezik omogućuje definiranje serije instrukcija koje operacijskom sustavu preciziraju koje zadatke treba izvršiti i s kojim resursima [13]. JCL pruža niz instrukcija operacijskom sustavu, specificirajući kako obraditi ulazne podatke, gdje pohraniti izlazne podatke, koje programe pokrenuti i kako upravljati ispisom. [13].

Sintaksa JCL-a slijedi stroga, pozicijski definirana pravila. Primjer JCL JOB kartice je prikazan u tekstu „Programski kod 1: JOB kartica“. JCL naredba počinje s dvije kose crte ("//"). Početak JCL-a naziva se JOB kartica. Prvi izraz je *JOB* naredba koja definira početak posla. Naziv posla može biti sistemski ili korisnički definiran, nakon čega slijedi niz parametara koji konfiguriraju različite postavke posla. Naslov izraza popraćen je naredbom, u slučaju prvog izraza to je *JOB* naredba, koja definira početak izraza. Nakon *JOB* naredbe dolazi ključna riječ, ona može biti naslov ili informacija o samoj skripti. Parametri koje je poželjno definirati su *MSGCLASS*, klasa izlazne informacije o *JCL JOB*-u, određuje na što ide ispis samog *JOB*-a. Startna opcija *MSGLEVEL*, definira razinu izlaznih poruka na spoolu. Parametar *NOTIFY* služi kao upozorenje korisniku kako se odvio njegov JCL.

Programski kod 1: JOB kartica

**//JCLJOB JOB (KNJIZNICA),MSGCLASS=X,MSGLEVEL=(1,1), // NOTIFY=&SYSUID**

Vrijednost parametra za klasu poruke „X“, simbolizira ispis u spool. Vrijednost parametra *MSGLEVEL* simbolizira od koje vrste informacija o izvršenom JOB-u želimo na ispisu u spoolu, u ovom slučaju, parametar 1 simbolizira da će biti ispisani svi JCL izrazi iz skripte, svi kontrolni izrazi, navedene procedure (ako ih ima), dok parametar 2 simbolizira koje sve informacijske poruke treba ispisati na ispis u spoolu, u ovom slučaju to je cijela JCL skripta za referencu, operator, i sve relevantne izlazne poruke o izvršavanju [14]. Vrijednost izraza „1“ govori sistemu da ispiše sve JCL izraze, JES kontrolne izraze, izraze JCL procedura i informacijskih poruka [15]. Parametar *NOTIFY* signalizira sustavu da izda obavijest kada dođe do promjene u statusu JOB-a ,vrijednost *&SYSUID* je varijabla koja dopremi JCL-u ime korisnika koji podnosi („Submita“) JOB [16]. Zarez na kraju prvog retka simbolizira nastavak pisanja istog u novi red, parametar bez zareza na kraju simbolizira kraj JCL izraza.

JCL je vitalna komponenta z/OS sustava, omogućujući precizno definiranje i upravljanje poslom na z/OS sustavu. Precizne i moćne funkcionalnosti JCL-a ga čine nezamjenjivim za sistemske i z/OS aplikacijske programere. U ovom radu, JCL se primarno koristi za kompiliranje programa i ekrana za CICS upravitelj transakcija.

## Db2 Baza podataka

Db2 je baza podataka i sustav za upravljanje (DBMS) na z/OS operacijskom sustavu, dizajniran za obradu velikih količina podataka u poslovnim aplikacijama [7]. Bazu podataka kreirao je IBM, kao dio serije proizvoda za upravljanje podacima 1983. godine [17]. Db2 omogućuje pouzdano i sigurno upravljanje podacima, pružajući skalabilnost i visoku dostupnost. Db2 na z/OS-u koristi relacijsku arhitekturu koja omogućuje jednostavno modeliranje podataka i provođenje kompleksnih upita.

Arhitektura Db2 sastoji se od više komponenata koje surađuju kako bi osigurale optimalne performanse i pouzdanost. Upravljač radnog opterećenja z/OS operacijskog sustava određuje servisne klase za Db2 performanse, te tako optimizira rad na samoj Db2 bazi podataka [18]. Administrator baze podatka postavlja definicije željenih performansi i prioriteta poslovnih logika unutar servisnih klasa baze podataka [18]. Međuspremnici Db2 obnašaju ulogu kontinuiranog privremenog pohranjivanja prostora tablica ili indeksa [19]. Kada program ili korisnik pristupa tablici ili retku putem upita, Db2 stavlja rezultat tog upita u međuspremnik, što osigurava značajne brzine pri radu s Db2 bazom podataka [19]. Baza također koristi mehanizme za zaključavanje, koji upravljaju kontinuiranim pristupom podataka, osiguravajući integritet. Sadržajem baze i administracijom upravlja se pomoću *Structured Query Language* (strukturno upitni jezik, SQL). Upravljanje Db2 bazom podataka zahtijeva planiranje i nadzor posla. Alati za oporavak podataka, pregled performansi i sigurnosne mjere zaštite su neophodne inačice Db2 baze podataka [7].

Db2 na z/OS operacijskom sustavu je temeljna tehnologija za upravljanje masovnom količinom podataka u poslovnim okruženjima. Robusna arhitektura, napredne funkcionalnosti i alati za upravljanje čine Db2 neizostavnom komponentom u arhitekturi većine Mainframe sustava. Db2 omogućuje organizacijama i institucijama, efikasno upravljanje podacima i poslovnim procesima.

Slika na kojoj se prikazuje tekst, grafika, Font, logotip

Opis je automatski generiran

Slika 4 IBM Db2 Logo [20]

## PL/I programski Jezik

PL/I (eng., Programming Language One) (Programski kod 2) je višenamjenski, proceduralni programski jezik razvijen 1960-ih godina s ciljem ujedinjavanja karakteristika iz nekoliko popularnih jezika tog vremena, poput Fortrana, COBOL-a i Algola [21]. Namijenjen je prvenstveno za znanstvene, poslovne i tehničke primjene, te je dizajniran kako bi pružio fleksibilnost u različitim okruženjima, uključujući poslovne aplikacije, inženjerske simulacije, kao i aplikacije u području znanosti.

PL/I omogućava programerima da pišu visoko strukturirani kod koristeći napredne koncepte kao što su više niti (multithreading), rukovanje iznimkama, te mogućnosti dinamičke alokacije memorije.

Njegova sposobnost za rad s velikim količinama podataka, kao i integracija s bazama podataka kao što je Db2, čine ga vrlo korisnim u okruženju velikih poslovnih sustava, osobito na platformama poput IBM-ovog z/OS mainframe sustava.

|  |
| --- |
| Hello:procedure options(main);  put skip edit('Hello World')(A);  end Hello; |

Programski kod 2: Hello, world u programskom jeziku PL/I

Jedna od glavnih prednosti PL/I jezika je njegova svestranost, jer omogućava implementaciju kako proceduralnih, tako i objektno-orijentiranih paradigmi. To ga čini pogodnim za složene transakcijske sustave kao što su oni koji koriste CICS (Customer Information Control System), gdje je ključna brzina obrade velikog broja korisničkih zahtjeva.

U kontekstu ovog završnog rada, PL/I je korišten kao osnovni programski jezik za razvoj sustava upravljanja knjižnicom, uz implementaciju CICS transakcija za komunikaciju s korisnikom putem terminala. Njegova sposobnost rada s Db2 bazom podataka omogućava efikasnu pohranu i dohvaćanje podataka o korisnicima i knjigama unutar knjižničnog sustava. PL/I se pokazao kao idealan alat za razvoj ovakvih poslovnih aplikacija zbog svoje stabilnosti, performansi i integracije s ostalim alatima na z/OS platformi.

## CICS upravitelj transakcija

CICS („Customer Information Control System“) je upravitelj transakcija koji je razvio IBM za z/OS i z/VSE operacijske sustave [6]. Uz mogućnost upravljanja transakcijama, CICS je i višejezični aplikacijski server koji omogućuje razvoj i izvršavanje aplikacija koje istovremeno pristupaju bazama podataka i obavljaju online transakcije [6]. CICS pruža visok stupanj pouzdanosti, brzine i sigurnosti pri izvršavanju svog rada na sustavima, što ga čini ključnim elementom poslovnih okruženja u industrijskim sektorima poput financija, bankarstva, maloprodaje i javnih usluga.

Osmišljen za veliki opseg posla i broj korisnika na sustavu, uz upravljanje kritičnim aplikacijama, CICS je nezaobilazan softver za većinu Z[[2]](#footnote-3) sustava. Glavne funkcionalnosti CICS-a su: upravljanje transakcijama, upravljanje resursima, sigurnost i integracija s bazama podataka.

Efikasno upravljanje transakcijama postignuto je uz očuvanje integriteta podataka i simultano izvršavanje tisuća istih bez značajnih gubitaka performansi. Upravljanje resursima poput baza podataka, datoteka, terminala i komunikacijskih kanala uz aplikacijsko pristupanje podacima na siguran i kontroliran način. CICS također podržava standardne sigurnosne mehanizme poput identifikacije i autorizacije korisnika, autorizacija resursa, te šifriranje podataka u komunikaciji. CICS posjeduje solidnu integraciju s Db2 bazom podataka. Svaka od ovih operacija tretira se kao transakcija u CICS-u, čime se osigurava da su svi podaci konzistentni i da nema gubitka informacija u slučaju nepredviđenih događaja poput prekida rada sustava.

CICS upravitelj transakcija ključan je element za razvoj robusnih, sigurnih i efikasnih poslovnih aplikacija u z/OS okruženju. Njegova integracija s DB2 bazom podataka omogućuje stvaranje složenih transakcijskih sustava, kao što je sustav za upravljanje knjižnicama. Razumijevanje osnovnih principa rada CICS-a i njegove primjene u različitim poslovnim scenarijima od iznimne je važnosti za uspješnu implementaciju poslovnih aplikacija na mainframe platformi.

# PROCES IZRADE SISTEMA

U procesu izrade sustava za obradu podataka knjižnice, ključno je pokrenuti sve funkcionalne cjeline na kojima se izvršavaju aplikacije. Prvi koraci ovog procesa uključuju detaljnu analizu pokretačkih skripti za Db2 bazu podataka, CICS transakcijski sustav, te postavljanje modula za kompiliranje aplikacija.

Kako bi se Db2 baza podataka i CICS transakcijski sustav uspješno pokrenuli, potrebno je provjeriti dokumentacijske upute za verziju koja se podiže. Nakon provjerenih pokretačkih skripti, potrebno je unesti komande u System Display and Search Facility (SDSF) za određeno pokretanje sustava. Za pokretanje Db2 sustava, koristi se JCL pokretački JOB *DBCG,* naredbom: */-DBCG START DB2* [22]koja će pokrenuti procedure s profilom iz svoje sistemske biblioteke DSNC10.PROCLIB (Db2 verzija 12). Nakon puštanja naredbe na SDSF administracijsku konzolu, vrši se provjera postoji li pokrenuta instanca JOB-a na samom sistemu. Pokrenuti JOB-ovi s profilom DBCG nalaze se na Job Entry Subsystem (JES) spool-u, u stanju „EXECUTION“ (Slika 5 – Status izvršavanja komponenti Db2 baze podataka) , što znači da je Db2 pokrenut i da je u stanju izvođenja na z/OS-u. Isti proces se vrši i za CICS transakcijski sustav, samo se naredba za isti pušta pomoću START komande: */S CICSTS61* (Start CICS v6.1) [23], te se također nalazi u „EXECUTION“ stanju na JES spool-u (Slika 6 – Status izvršavanja CICS transakcijskog sustava). Nakon podizanja ključnih stavki za sustav knjižnice, potrebno je izraditi ekrane pomoću kojih korisnik ima interakciju s CICS sustavom i bazom podataka, te samu Db2 bazu podatka, dizajniranu prema potrebama knjižnice. CICS ekrani za interakciju sa sustavom kreirani su po samoj prezentacijskoj i upravljačkoj logici Db2 baze podataka, tako da svaki ekran simbolizira jednostavniji primjer *SELECT,* *INSTERT* i *DELETE* SQL upita na sadržaj podataka knjižnice.  
 Kroz daljnja poglavlja opisan je detaljniji pristup realizacije sustava obrade podataka knjižnice, uključujući i logiku kreiranja sheme baze podataka, normalizacija podataka, logiku odnosa tablica i automatsko ažuriranje. Nakon postavljanja i kreiranja svih ključnih komponenti, na fokus dolazi izrada CICS transakcija u PL/I jeziku, što je vitalni dio ovog sustava. Transakcijska logika opisana je zajedno s logikom izrade ekrana, jer ekrani su vizualna reprezentacija poslovne logike sustava za obradu podataka knjižnice.

Slika na kojoj se prikazuje tekst, snimka zaslona

Opis je automatski generiran

Slika 5 - Status izvršavanja komponenti Db2 baze podataka

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

Slika 6 - Status izvršavanja CICS transakcijskog sistema

## Izrada Db2 tablica

Proces izrade tablica Db2 baze podataka za potrebe sustava knjižnice opisano je pomoću produkcijskog načina razvoja i standarda. Svaka potreba knjižnice zadovoljena je implementacijom isplanirane strukture tablica baze podataka (Slika 7 – Prikaz sheme Db2 baze podataka sustava knjižnice). Kroz ovo poglavlje je detaljno obrađen pristup kreiranja svake strukture, njihovih stupaca, primarnih i sekundarnih ključeva, indeksa, te način na koji su tablice među povezane. Naglasak je na pojašnjenju svrhe i načina izrade svakog elementa unutar baze podataka zbog osiguranja optimalne organizacije i učinkovitosti korištenja sustava.

Kreirane tablice uključuju osnovne podatke o autorima, članovima knjižnice, inventarom knjiga, informacijama o dobavljačima, posudbama, rezervacijama i zakasnini. Svaka tablica ima jasno definiranu strukturu koja uključuje primarni ključ zbog jednoznačne identifikacije svakog retka, kao i potrebne strane ključeve za osiguranje konzistencije i povezivanja podataka između različitih tablica. Osim same izrade tablica, obrađuje se i definiranje jedinstvenih indeksa koji omogućuju brži programski pristup podacima, kao i kreiranje pogleda (eng., views) koji olakšavaju rad s podacima iz više tablica, čineći upite jednostavnijima i preglednijima. Prikazi omogućuju složene operacije pretraživanja i dohvaćanja podataka, uz lakši korisnički pristup i navigaciju podacima. Uz indekse i poglede, dodani su i komentari entiteta baze podataka na razini stupaca, s ciljem boljeg dokumentiranja i lakšeg razumijevanja baze podataka. Komentari pružaju dodatne informacije o svrsi i značenju svakog stupca, čime se olakšava održavanje i mogućnost dodatnog poboljšanja sustava.

Detaljnim opisima postupka kreiranja tablica, indeksa, pogleda i komentara, ovo poglavlje pruža jasan uvid u tehnički aspekt izgradnje baze podataka koja podržava svakodnevno poslovanje knjižnice. Također, naglašava se važnost pravilnog modeliranja podataka kako bi se osigurala efikasnost, integritet i skalabilnost sustava knjižnice.

Slika na kojoj se prikazuje tekst, dijagram, snimka zaslona, broj

Opis je automatski generiran

Slika 7- Prikaz sheme Db2 baze podataka sustava knjižnice (Kreirano pomoću DB Designer alata)

### Tablica KNJIGE

Tablica knjige jedna je od ključnih tablica sustava obrade podataka knjižnice, jer sadrži informacije o svim knjigama dostupnim u knjižnici. Ova tablica služi za pohranu podataka kao što su naslov knjige, naziv autora, kategorija, godina izdanja, te broj primjeraka u inventaru. Sama tablica ima prefiks „TAB“ na imenovanju svakog stupca, dok pogled (eng., view) ima samo ime stupca, zbog jednostavnosti .Opis strukture u Programskom kodu 3 – SQL za kreiranje tablice KNJIGE. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake knjige.

Struktura tablice *KNJIGE* sastoji se od stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake knjige.
* *TABNAZIV* – Naziv knjige, ne može biti prazan, pohranjen je kao tekst od 60 znakova.
* *TABAUTOR* – Ime i prezime ili pseudonim autora knjige, pohranjen kao tekst od 60 znakova
* *TABKATGRJ* – Kategorija knjige, omogućuje bolje sortiranje i pretragu, pohranjen kao tekst od 20 znakova.
* *TABGODIZD* – Godina izdanja, pohranjena kao cjelobrojna vrijednost koja omogućuje praćenje povijesti izdanja.
* *TABBRKNJ* – Broj knjige, cjelobrojna vrijednost koja označava broj primjeraka iste knjige u inventaru.

Za tablicu *KNJIGE* definiran je jedinstveni indeks na stupac *TABID* kako bi se omogućilo brže pretraživanje prema samom identifikacijskom broju knjige. Kreirani pogled (eng., view), naziva *VIEW\_KNJIGE,* omogućuje jednostavan pristup najvažnijim podacima o knjigama bez potrebe za kompleksnim imenovanjem stupaca. Dodani komentari na razini svakog stupca olakšavaju razumijevanje entiteta i održavanje baze podataka.

Programski kod 3 - SQL za kreiranje tablice KNJIGE

|  |
| --- |
| /\* Kreiranje tablice KNJIGE s primarnim ključem \*/  CREATE TABLE KNJIGE (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID knjige \*/      TABNAZIV VARCHAR(60) NOT NULL, /\* TABNAZIV - Naziv knjige \*/      TABAUTOR VARCHAR(60), /\* TABAUTOR - Naziv autora knjige \*/      TABKATGRJ VARCHAR(20), /\* TABKATGRJ - Kategorija knjige \*/      TABGODIZD INTEGER, /\* TABGODIZD - Godina izdanja \*/      TABBRKNJ INTEGER, /\* TABBRKNJ - Broj knjige \*/        /\* Definiranje primarnog ključa s imenom \*/      CONSTRAINT PK\_KNJIGE PRIMARY KEY (TABID)  );  /\* Kreiranje jedinstvenog indeksa za primarni ključ \*/  CREATE UNIQUE INDEX IDX\_UNIQUE\_KNJIGE\_ID ON KNJIGE (TABID);  /\* Kreiranje view-a za tablicu KNJIGE \*/  CREATE VIEW VIEW\_KNJIGE AS      SELECT          TABID AS ID, /\* ID knjige \*/          TABNAZIV AS NAZIV, /\* Naziv knjige \*/          TABAUTOR AS AUTOR, /\* Naziv autora \*/          TABKATGRJ AS KATGRJ, /\* Kategorija knjige \*/          TABGODIZD AS GODIZD, /\* Godina izdanja \*/          TABBRKNJ AS BRKNJ /\* Broj knjige \*/      FROM KNJIGE;  /\* Dodavanje komentara na stupce \*/  COMMENT ON COLUMN KNJIGE.TABID IS 'TABID - ID knjige';  COMMENT ON COLUMN KNJIGE.TABNAZIV IS 'TABNAZIV - Naziv knjige';  COMMENT ON COLUMN KNJIGE.TABAUTOR IS 'TABAUTOR - Naziv autora knjige';  COMMENT ON COLUMN KNJIGE.TABKATGRJ IS 'TABKATGRJ - Kategorija knjige';  COMMENT ON COLUMN KNJIGE.TABGODIZD IS 'TABGODIZD - Godina izdanja';  COMMENT ON COLUMN KNJIGE.TABBRKNJ IS 'TABBRKNJ - Broj knjige'; |

### Tablica CLANOVI

Tablica *CLANOVI* (Programski kod 4– SQL za kreiranje tablice *CLANOVI*)sadrži informacije o svim članovima knjižnice. Ova tablica pohranjuje osnovne podatke o članovima, kao što su prezime, ime, datum učlanjenja i status članstva. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svakog člana.

Struktura tablice *CLANOVI* sastoji se od sljedećih stupaca:

* *TABID* – Primarni ključ automatski generiran za jednoznačnu identifikaciju svakog člana.
* *TABPREZ* - Prezime člana, obavezan podatak, pohranjen kao tekst od 60 znakova.
* *TABIME* - Ime člana, pohranjeno kao tekst od 60 znakova.
* *TABDATOD* – „Datum od…“ Datum učlanjenja, pohranjen kao tip podataka DATE, koji omogućuje praćenje kada se član pridružio knjižnici.
* *TABSTATUS* - Status člana, pohranjen kao cjelobrojna vrijednost, koja označava trenutni status člana (aktivan, čekanje potvrde uplate članarine ili neaktivan).

Za tablicu *CLANOVI* definiran je i jedinstveni indeks koji omogućuje bržu pretragu podataka po *ID-u* člana. Također, kreiran je i pogled (Eng., view) pod nazivom *VIEW\_CLANOVI*, koji olakšava dohvaćanje ključnih informacija o članovima, kao što su *ID,* prezime, ime, datum učlanjenja i status.

Programski kod 4 – SQL za kreiranje tablice CLANOVI

|  |
| --- |
| /\* Kreiranje tablice CLANOVI s primarnim ključem \*/  CREATE TABLE CLANOVI (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID člana \*/      TABPREZ VARCHAR(60) NOT NULL, /\* TABPREZ - Prezime člana \*/      TABIME VARCHAR(60), /\* TABIME - Ime člana \*/      TABDATOD DATE, /\* TABDATOD - Datum uclanjenja \*/      TABSTATUS INTEGER, /\* TABSTATUS - Status člana \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_CLANOVI PRIMARY KEY (TABID)  ); |

### Tablica VEZNA1

Tablica *VEZNA1* (Programski kod 5 – SQL za kreiranje tablice *VEZNA1*) koristi se za povezivanje autora i knjiga te osigurava referencijalni integritet između tablice *AUTORI* i tablice *KNJIGE*. Ova tablica omogućuje pohranu informacija o vezi između određenog autora i njegovih knjiga. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake veze.

Struktura tablice *VEZNA1* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake veze.
* *TABIDAUT* - *ID* autora, strani ključ koji se povezuje s tablicom *AUTORI*, osiguravajući referencijalni integritet.
* *TABIDKNJ* - *ID* knjige, strani ključ koji se povezuje s tablicom *KNJIGE*, osiguravajući referencijalni integritet.

Za tablicu *VEZNA1* definiran je jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* veze. Također, kreiran je i pogled (eng., view) pod nazivom *VIEW\_VEZNA1*, koji olakšava dohvaćanje informacija o vezama između autora i knjiga. Tablica *VEZNA1* koristi strane ključeve kako bi se osigurala konzistentnost podataka između tablica *AUTORI* i *KNJIGE*. Definirani su strani ključevi *FK\_VEZNA1\_AUT* i *FK\_VEZNA1\_KNJ*, koji osiguravaju da se svaka veza odnosi na postojećeg autora i knjigu u odgovarajućim tablicama.

Programski kod 5 – SQL kod za kreiranje tablice VEZNA1

|  |
| --- |
| /\* Kreiranje tablice VEZNA1 s primarnim i stranim ključevima \*/  CREATE TABLE VEZNA1 (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID veze \*/      TABIDAUT INTEGER NOT NULL, /\* TABIDAUT - ID autora (strani ključ) \*/      TABIDKNJ INTEGER NOT NULL, /\* TABIDKNJ - ID knjige (strani ključ) \*/        /\* Definiranje primarnog ključa s imenom \*/      CONSTRAINT PK\_VEZNA1 PRIMARY KEY (TABID),      /\* Definiranje stranog ključa za autore \*/      CONSTRAINT FK\_VEZNA1\_AUT FOREIGN KEY (TABIDAUT)          REFERENCES AUTORI(TABID)          ON DELETE NO ACTION,      /\* Definiranje stranog ključa za knjige \*/      CONSTRAINT FK\_VEZNA1\_KNJ FOREIGN KEY (TABIDKNJ)          REFERENCES KNJIGE(TABID)          ON DELETE NO ACTION  );  /\* Kreiranje jedinstvenog indeksa za primarni ključ \*/  CREATE UNIQUE INDEX IDX\_UNIQUE\_VEZNA1\_ID ON VEZNA1 (TABID);  /\* Kreiranje view-a za tablicu VEZNA1 \*/  CREATE VIEW VIEW\_VEZNA1 AS      SELECT          TABID AS ID, /\* ID veze \*/          TABIDAUT AS IDAUT, /\* ID autora \*/          TABIDKNJ AS IDKNJ /\* ID knjige \*/      FROM VEZNA1;  /\* Dodavanje komentara na stupce \*/  COMMENT ON COLUMN VEZNA1.TABID IS 'TABID - ID veze';  COMMENT ON COLUMN VEZNA1.TABIDAUT IS 'TABIDAUT - ID autora';  COMMENT ON COLUMN VEZNA1.TABIDKNJ IS 'TABIDKNJ - ID knjige'; |

### Tablica AUTORI

Tablica *AUTORI* (Programski kod 6 – SQL za kreiranje tablice *AUTORI*) sadrži informacije o svim autorima knjiga u knjižnično-informacijskom sustavu. Ova tablica služi za pohranu podataka kao što su prezime i ime autora. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svakog autora.

Struktura tablice *AUTORI* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svakog autora.
* *TABPREZ* - Prezime autora, obavezan podatak, pohranjen kao tekst od 60 znakova.
* *TABIME* – Ime autora, pohranjeno kao tekst od 60 znakova.

Za tablicu *AUTORI* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* autora. Također, kreiran je i pogled (view) pod nazivom *VIEW\_AUTORI*, koji omogućuje jednostavan pristup najvažnijim podacima o autorima, kao što su *ID*, prezime i ime.

Programski kod 6 – SQL za kreiranje tablice AUTORI

|  |
| --- |
| /\* Kreiranje tablice AUTORI s primarnim ključem \*/  CREATE TABLE AUTORI (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID autora \*/      TABPREZ VARCHAR(60) NOT NULL, /\* TABPREZ - Prezime autora \*/      TABIME VARCHAR(60), /\* TABIME - Ime autora \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_AUTORI PRIMARY KEY (TABID)  ); |

### Tablica REZERV

Tablica *REZERV* (Programski kod 7 *– SQL za kreiranje tablice REZERV*) koristi se za pohranu podataka o rezervacijama knjiga od strane članova knjižnice. Ova tablica omogućuje praćenje rezervacija, uključujući informacije o knjizi koja je rezervirana, članu koji je napravio rezervaciju, te status i datum rezervacije. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake rezervacije.

Struktura tablice *REZERV* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake rezervacije.
* *TABIDKNJ* - *ID* knjige, strani ključ koji se povezuje s tablicom *KNJIGE*, osiguravajući referencijalni integritet.
* *TABIDCLAN* - *ID* člana, strani ključ koji se povezuje s tablicom *CLANOVI*, osiguravajući referencijalni integritet.
* *TABDATREZ* - Datum rezervacije, pohranjen kao tip podataka *DATE*, koji označava kada je rezervacija napravljena.
* *TABREZSTAT* - Status rezervacije, pohranjen kao cjelobrojna vrijednost koja označava trenutni status rezervacije (npr. aktivna, otkazana).

Za tablicu *REZERV* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* rezervacije. Također, kreiran je i pogled (view) pod nazivom *VIEW\_REZERV*, koji omogućuje jednostavan pristup ključnim podacima o rezervacijama, kao što su *ID*, *ID* knjige, *ID* člana, datum rezervacije i status rezervacije. Tablica *REZERV* koristi strane ključeve kako bi se osigurala konzistentnost podataka između tablica *KNJIGE* i *CLANOVI*. Definirani su strani ključevi *FK\_REZERV\_KNJ* i *FK\_REZERV\_CLAN*, koji osiguravaju da se svaka rezervacija odnosi na postojeću knjigu i člana u odgovarajućim tablicama.

Programski kod 7 – SQL za kreiranje tablice REZERV

|  |
| --- |
| /\* Kreiranje tablice REZERV s primarnim i stranim ključevima \*/  CREATE TABLE REZERV (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID rezervacije \*/      TABIDKNJ INTEGER NOT NULL, /\* TABIDKNJ - ID knjige (strani kljuc) \*/      TABIDCLAN INTEGER NOT NULL, /\* TABIDCLAN - ID clana (strani kljuc) \*/      TABDATREZ DATE, /\* TABDATREZ - Datum rezervacije \*/      TABREZSTAT INTEGER, /\* TABREZSTAT - Status rezervacije \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_REZERV PRIMARY KEY (TABID),      /\* Definiranje stranog kljuca za knjige \*/      CONSTRAINT FK\_REZERV\_KNJ FOREIGN KEY (TABIDKNJ)          REFERENCES KNJIGE(TABID)          ON DELETE NO ACTION,      /\* Definiranje stranog kljuca za članove \*/      CONSTRAINT FK\_REZERV\_CLAN FOREIGN KEY (TABIDCLAN)          REFERENCES CLANOVI(TABID)          ON DELETE NO ACTION  ); |

### Tablica DOBAVLJ

Tablica *DOBAVLJ* (Programski kod 8 – SQL za kreiranje tablice *DOBAVLJ*) koristi se za pohranu podataka o dobavljačima koji surađuju s knjižnicom. Ova tablica omogućuje praćenje osnovnih informacija o dobavljačima, uključujući njihov naziv i kontakt broj. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svakog dobavljača.

Struktura tablice *DOBAVLJ* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svakog dobavljača.
* *TABNAZIV* - Naziv dobavljača, obavezan podatak, pohranjen kao tekst do 60 znakova.
* *TABKONTBR* - Kontakt broj, pohranjen kao cjelobrojna vrijednost, koji služi za kontaktiranje dobavljača.

Za tablicu *DOBAVLJ* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* dobavljača. Također, kreiran je i pogled (view) pod nazivom *VIEW\_DOBAVLJ,* koji omogućuje jednostavan pristup ključnim podacima o dobavljačima, kao što su ID, naziv i kontakt broj.

Programski kod 8 – SQL za kreiranje tablice DOBAVLJ

|  |
| --- |
| /\* Kreiranje tablice DOBAVLJ s primarnim ključem \*/  CREATE TABLE DOBAVLJ (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID dobavljača \*/      TABNAZIV VARCHAR(60) NOT NULL, /\* TABNAZIV - Naziv dobavljača \*/      TABKONTBR INTEGER, /\* TABKONTBR - Kontakt broj \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_DOBAVLJ PRIMARY KEY (TABID)  ); |

### Tablica NABAVA

Tablica *NABAVA* (Programski kod 9 - SQL za kreiranje tablice *NABAVA*) koristi se za pohranu podataka o nabavama knjiga koje knjižnica nabavlja od dobavljača. Ova tablica omogućuje praćenje informacija o tome koja je knjiga nabavljena, tko je dobavljač te kada je nabava obavljena. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake nabave.

Struktura tablice *NABAVA* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake nabave
* *TABIDKNJ* - ID knjige, strani ključ koji se povezuje s tablicom *KNJIGE* radi osiguranja referencijalnog integriteta
* *TABIDDOB* - ID dobavljača, strani ključ koji se povezuje s tablicom *DOBAVLJ* radi osiguranja referencijalnog integriteta
* *TABDATNAB* - Datum nabave, pohranjen kao tip podataka *DATE*, koji označava datum kada je nabava obavljena

Za tablicu *NABAVA* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* nabave. Također, kreiran je i pogled (view) pod nazivom *VIEW\_NABAVA*, koji omogućuje jednostavan pristup ključnim podacima o nabavama, kao što su *ID* nabave, *ID* knjige, *ID* dobavljača i datum nabave. Tablica *NABAVA* koristi strane ključeve kako bi se osigurala konzistentnost podataka između tablica *KNJIGE* i *DOBAVLJ*. Definirani su strani ključevi *FK\_NABAVA\_KNJ* i *FK\_NABAVA\_DOB*, koji osiguravaju da se svaka nabava odnosi na postojeću knjigu i dobavljača u odgovarajućim tablicama.

Programski kod 9 - SQL za kreiranje tablice NABAVA

|  |
| --- |
| /\* Kreiranje tablice NABAVA s primarnim i stranim ključevima \*/  CREATE TABLE NABAVA (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID nabave \*/      TABIDKNJ INTEGER NOT NULL, /\* TABIDKNJ - ID knjige (strani kljuc) \*/      TABIDDOB INTEGER NOT NULL, /\* TABIDDOB - ID dobavljača (strani kljuc) \*/      TABDATNAB DATE, /\* TABDATNAB - Datum nabave \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_NABAVA PRIMARY KEY (TABID),      /\* Definiranje stranog kljuca za knjige \*/      CONSTRAINT FK\_NABAVA\_KNJ FOREIGN KEY (TABIDKNJ)          REFERENCES KNJIGE(TABID)          ON DELETE NO ACTION,      /\* Definiranje stranog kljuca za dobavljače \*/      CONSTRAINT FK\_NABAVA\_DOB FOREIGN KEY (TABIDDOB)          REFERENCES DOBAVLJ(TABID)          ON DELETE NO ACTION  ); |

### Tablica POSUDBA

Tablica *POSUDBA* (Programski kod 10 - SQL za kreiranje tablice *POSUDBA*) koristi se za pohranu podataka o posudbama knjiga koje su članovi knjižnice posudili. Ova tablica omogućuje praćenje informacija o tome koja je knjiga posuđena, tko je član koji je posudio knjigu te kada je posudba obavljena. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake posudbe.

Struktura tablice *POSUDBA* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake posudbe
* *TABIDKNJ* - ID knjige, strani ključ koji se povezuje s tablicom *KNJIGE* radi osiguranja referencijalnog integriteta
* *TABIDCLAN* - ID člana, strani ključ koji se povezuje s tablicom *CLANOVI* radi osiguranja referencijalnog integriteta
* *TABDATPOS* - Datum posudbe, pohranjen kao tip podataka *DATE*, koji označava datum kada je posudba obavljena

Za tablicu *POSUDBA* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* posudbe. Također, kreiran je i pogled (Eng., view) pod nazivom *VIEW\_POSUDBA*, koji omogućuje jednostavan pristup ključnim podacima o posudbama, kao što su *ID* posudbe, *ID* knjige, *ID* člana i datum posudbe.

Tablica *POSUDBA* koristi strane ključeve kako bi se osigurala konzistentnost podataka između tablica *KNJIGE* i *CLANOVI*. Definirani su strani ključevi *FK\_POSUDBA\_KNJIGA* i *FK\_POSUDBA\_CLAN*, koji osiguravaju da se svaka posudba odnosi na postojeću knjigu i člana u odgovarajućim tablicama.

Programski kod 10 - SQL za kreiranje tablice POSUDBA

|  |
| --- |
| /\* Kreiranje tablice POSUDBA s primarnim i stranim ključevima \*/  CREATE TABLE POSUDBA (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID posudbe \*/      TABIDKNJ INTEGER NOT NULL, /\* TABIDKNJ - ID knjige (strani kljuc) \*/      TABIDCLAN INTEGER NOT NULL, /\* TABIDCLAN - ID clana (strani kljuc) \*/      TABDATPOS DATE, /\* TABDATPOS - Datum posudbe \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_POSUDBA PRIMARY KEY (TABID),      /\* Definiranje stranog kljuca za knjige \*/      CONSTRAINT FK\_POSUDBA\_KNJIGA FOREIGN KEY (TABIDKNJ)          REFERENCES KNJIGE(TABID)          ON DELETE NO ACTION,      /\* Definiranje stranog kljuca za članove \*/      CONSTRAINT FK\_POSUDBA\_CLAN FOREIGN KEY (TABIDCLAN)          REFERENCES CLANOVI(TABID)          ON DELETE NO ACTION  ); |

### Tablica ZAKAS

Tablica *ZAKAS* (Programski kod 11 - SQL za kreiranje tablice *ZAKAS*) koristi se za pohranu podataka o zakasnini koju članovi knjižnice moraju platiti za kasno vraćanje knjiga. Ova tablica omogućuje praćenje informacija o tome koji je član kasnio s vraćanjem knjige, koja je posudba u pitanju, datum zakasnine, te iznos zakasnine. Primarni ključ tablice je stupac *TABID*, koji osigurava jedinstvenu identifikaciju svake zakasnine.

Struktura tablice *ZAKAS* sastoji se od sljedećih stupaca:

* *TABID* - Primarni ključ, automatski generiran za jednoznačnu identifikaciju svake zakasnine
* *TABIDCLAN* - *ID* člana, strani ključ koji se povezuje s tablicom *CLANOVI* radi osiguranja referencijalnog integriteta
* *TABIDPOS* - *ID* posudbe, strani ključ koji se povezuje s tablicom *POSUDBA* radi osiguranja referencijalnog integriteta
* *TABZAKDAT* - Datum zakasnine, pohranjen kao tip podataka *DATE*, koji označava datum kada je zakasnina nastala
* *TABIZNOS* - Iznos zakasnine, pohranjen kao cjelobrojna vrijednost koja označava iznos zakasnine koju član mora platiti

Za tablicu *ZAKAS* definiran je i jedinstveni indeks kako bi se omogućila brža pretraga podataka po *ID-u* zakasnine. Također, kreiran je i pogled (view) pod nazivom *VIEW\_ZAKAS*, koji omogućuje jednostavan pristup ključnim podacima o zakasninama, kao što su *ID* zakasnine, *ID* člana, *ID* posudbe, datum zakasnine i iznos zakasnine. Tablica *ZAKAS* koristi strane ključeve kako bi se osigurala konzistentnost podataka između tablica *CLANOVI* i *POSUDBA*. Definirani su strani ključevi *FK\_ZAKAS\_CLAN* i *FK\_ZAKAS\_POS*, koji osiguravaju da se svaka zakasnina odnosi na postojećeg člana i posudbu u odgovarajućim tablicama.

Programski kod 11- SQL za kreiranje tablice ZAKAS

|  |
| --- |
| /\* Kreiranje tablice ZAKAS s primarnim i stranim ključevima \*/  CREATE TABLE ZAKAS (      TABID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY          (START WITH 1 INCREMENT BY 1 NO MAXVALUE ORDER), /\* TABID - ID zakasnine \*/      TABIDCLAN INTEGER NOT NULL, /\* TABIDCLAN - ID clana (strani kljuc) \*/      TABIDPOS INTEGER NOT NULL, /\* TABIDPOS - ID posudbe (strani kljuc) \*/      TABZAKDAT DATE, /\* TABZAKDAT - Datum zakasnine \*/      TABIZNOS INTEGER, /\* TABIZNOS - Iznos zakasnine \*/        /\* Definiranje primarnog kljuca s imenom \*/      CONSTRAINT PK\_ZAKAS PRIMARY KEY (TABID),      /\* Definiranje stranog kljuca za članove \*/      CONSTRAINT FK\_ZAKAS\_CLAN FOREIGN KEY (TABIDCLAN)          REFERENCES CLANOVI(TABID)          ON DELETE NO ACTION,      /\* Definiranje stranog kljuca za posudbe \*/      CONSTRAINT FK\_ZAKAS\_POS FOREIGN KEY (TABIDPOS)          REFERENCES POSUDBA(TABID)          ON DELETE NO ACTION  ); |

## Postavljanje PL/I, CICS i SQL kompajlera

Prije početka programiranja PL/I koda, potrebno je osposobiti kompajler za isti. PL/I kompajler je IBM-ov modul za kompilaciju koda, koji u sebi sadrži instrukcije za pretvaranje teksta u izvršivi modul. Pokretanjem modula dobiva se rad koji je opisan u tekstnim instrukcijama samoga koda. Za potrebe sustava knjižnice, potreban je kompajler za PL/I programski jezik, SQL kompajler za ugniježđene SQL izraze, te CICS translator, koji prevodi sve CICS instrukcije u ostatak koda gdje se iste nalaze [24].

PL/I kompajler, u praktičnom smislu, je niz JCL (Job Control Language) instrukcija u kojima se nalaze informacije odakle z/OS operacijski sustav pronalazi napisani kod, gdje sprema kompilirani modul PL/I koda, iz kojih setova podataka dodaje stvari u kod i što dodatno u kodu mora obraditi. Ovisno o sadržaju programa koji pišemo potrebno je složiti odgovarajuće kompajlere za:

* PL/I kod - Kompajler koji samo obrađuje čisti PL/I kod bez ikakvih ostalih dodataka
* PL/I + SQL – Kompajler koji obrađuje PL/I kod i ugradbeni SQL unutar koda. Prekompajliranjem SQL koda, kompajler Db2 bazi kreira Database Request Module, zahtijevani modul Db2 bazi podataka (DBRM).

## Izrada CICS ekrana za 3270 terminale

## Analiza i razvoj PL/I aplikacija

## Kompiliranje aplikacija

## Kreiranje, instaliranje i testiranje transakcija

# MOGUĆA POBOLJŠANJA SUSTAVA

# ZAKLJUČAK

U ovom poglavlju treba ukratko rezimirati rad. Osvrnuti se na postavljene ciljeve i, držeći se sadržaja rada, ukratko opisati najznačajnije rezultate. U ovom je poglavlju poželjno navesti i ograničenja ili područja moguće primjene dobivenih rezultata. **U zaključku se ne citira literatura niti se spominju grafički prilozi i tablice.** Zaključak (kao i uvod) ne treba biti predug (jedna do dvije stranice teksta).

# LITERATURA

|  |  |
| --- | --- |
| [1] | IBM, »CICS Transaction Server for z/OS,« 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=whats-new. [Pokušaj pristupa 11 lIPANJ 2024]. |
| [2] | IBM, »CICS Transaction Server for z/OS - Physical and symbolic map sets,« IBM, 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=map-physical-symbolic-sets. [Pokušaj pristupa 12 Lipnja 2024]. |
| [3] | IBM, »CICS Transaction Server for z/OS - EXEC interface block (EIB),« IBM, 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=applications-exec-interface-block-eib. [Pokušaj pristupa 12 Lipnja 2024]. |
| [4] | IBM, »CICS Transaction Server for z/OS - The Db2 address spaces,« IBM, 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=db2-address-spaces. [Pokušaj pristupa 12 Lipnja 2024]. |
| [5] | IBM, »CICS Transaction Server for z/OS - Enabling CICS Db2 applications to use OTE through threadsafe programming,« IBM, 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=stopccda-enabling-cics-db2-applications-use-ote-through-threadsafe-programming. [Pokušaj pristupa 12 Lipnja 2024]. |
| [6] | Customer Information Control System (CICS) General Information Manual, IBM. |
| [7] | IBM, »Db2 12 for z/OS - What is Db2 for z/OS?,« IBM, 2024. [Mrežno]. Available: https://ibm.com/docs/en/db2-for-zos/12?topic=getting-started-db2-zos. [Pokušaj pristupa 07 Srpanj 2024]. |
| [8] | IBM, »Enterprise PL/I for z/OS,« IBM, 2021. [Mrežno]. Available: https://www.ibm.com/docs/en/epfz/5.3. [Pokušaj pristupa 7 Srpanj 2024]. |
| [9] | IBM, »z/OS 2.4,« IBM, 2021. [Mrežno]. Available: https://www.ibm.com/docs/en/zos/2.4.0. [Pokušaj pristupa Srpanj 2024]. |
| [10] | IBM, »ACBs of z/OS System Programming Volume 1,« IBM RedBooks, 2017. [Mrežno]. Available: https://www.redbooks.ibm.com/redbooks/pdfs/sg246981.pdf. [Pokušaj pristupa Srpnja 2024]. |
| [11] | Wikipedia, »Wikipedia,« [Mrežno]. Available: https://en.wikipedia.org/wiki/Z/OS. [Pokušaj pristupa 7 Srpnja 2024.]. |
| [12] | IBM, »z/OS Basic Skills - Hardware resources used by z/OS,« IBM, 2010. [Mrežno]. Available: https://www.ibm.com/docs/en/zos-basic-skills?topic=1960s-hardware-resources-used-by-zos. [Pokušaj pristupa Srpanj 2024]. |
| [13] | IBM, »z/OS Basic Skills - Reusable JCL Collection,« IBM, 2010. [Mrežno]. Available: https://www.ibm.com/docs/en/zos-basic-skills?topic=collection-basic-jcl-concepts. [Pokušaj pristupa 11 Srpnja 2024]. |
| [14] | IBM, »z/OS 2.4.0 Documentation,« IBM, 2021. [Mrežno]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=mp-syntax-6. [Pokušaj pristupa 11 Srpnja 2024]. |
| [15] | IBM, »z/OS 2.4.0 DOcumentation,« IBM, 2021. [Mrežno]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=mp-subparameter-definition-6. [Pokušaj pristupa 11 Srpnja 2024]. |
| [16] | IBM, »z/OS 2.4.0 Documentation,« IBM, 2021. [Mrežno]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=user-sysuid. [Pokušaj pristupa 11 Srpnja 2024]. |
| [17] | Wikipedia, »IBM Db2,« Wikipedia, 12 06 2024. [Mrežno]. Available: https://en.wikipedia.org/wiki/IBM\_Db2. [Pokušaj pristupa 12 07 2024]. |
| [18] | IBM, »Db2 for z/OS - Determining z/OS Workload Manager velocity goals,« 14 06 2024. [Mrežno]. Available: https://www.ibm.com/docs/en/db2-for-zos/12?topic=db2-determining-zos-workload-manager-velocity-goals. [Pokušaj pristupa 12 07 2024]. |
| [19] | IBM, »Db2 for z/OS - Buffer Pools,« IBM, 18 06 2024. [Mrežno]. Available: https://www.ibm.com/docs/en/db2-for-zos/12?topic=objects-buffer-pools. [Pokušaj pristupa 12 07 2024]. |
| [20] | »MEDIUM,« [Mrežno]. Available: https://medium.com/codex/walk-through-on-ibm-db2-schema-97aa428f9f32. [Pokušaj pristupa 10 09 2024]. |
| [21] | Wikipedia, »PL/I - Wikipedia,« Wikipedia, 18 Rijna 2024. [Mrežno]. Available: https://en.wikipedia.org/wiki/PL/I. [Pokušaj pristupa 23 Rujna 2024]. |
| [22] | IBM, »Starting Db2,« IBM, 2024. [Mrežno]. Available: https://www.ibm.com/docs/en/db2-for-zos/12?topic=db2-starting. [Pokušaj pristupa 9 Listopad 2024]. |
| [23] | IBM, »What happens when you start CICS,« IBM, 14 Kolovoza 2024. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/6.x?topic=startup-what-happens-when-you-start-cics. [Pokušaj pristupa 9 Listopada 2024]. |
| [24] | IBM, »IBM Documentation,« 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/cics-ts/5.6?topic=routing-basic-mapping-support-bms. [Pokušaj pristupa 11 Lipanj 2024]. |
| [25] | IBM, »CICS Transaction Server for z/OS,« 10 Rujna 2023. [Mrežno]. Available: https://www.ibm.com/docs/en/SSGMCP\_5.6.0/pdf/api-reference\_pdf.pdf. [Pokušaj pristupa 12 Lipnja 2024]. |

# OZNAKE I KRATICE

AOR – Aplikacijska regija, „Application Owning Region“

API – Programsko sučelje aplikacije „Application Programming Interface“

CICS – Customer Information Control System

COBOL – COmmon Business Oriented Language

DASD – Direct Access Storage Device

Db2 – Database Two

DBMS – Database Managment Systems

DBRM – Database Request Module

DFS – Distributed File System

HLASM – High Level Asembler

IMS – Information Managment System

ISPF – Interactive System Productivity Facility

JCL – Jezik Kontrole Posla, „Job Control Language“

JES – Podsustav za unos posla, „Job Entry Subsystem“

PL/I – Programming Language One

RACF – Resource Access Control Facility

SDSF – System Display and Search Facility

SQL – Strukturni Upitni Jezik, „Structured Query Language“

STC – Started Tasks

TOR – Terminalna regija, „Terminal Owning Region“

TSO – Time Sharing Option

z/OS – Operacijski sustav Z, „Z Operating System“

ZFS – z/OS UNIX File System

# SAŽETAK

**Naslov**:

Sažetak vrlo kratko i jezgrovito izražava svrhu i bit rada. **Sažetak nije zaključak** i ne smije biti predug (**od 100 do najviše 300 riječi**). U njemu treba u najkraćim crtama pojasniti glavne dijelove rada (o čemu se u radu govorilo, na koji su se način prikupili podaci, gdje, kakvi su rezultati dobiveni). Svrha je sažetka **dati jasan prikaz nekog rada** na temelju kojeg potencijalni čitatelj odlučuje je li mu taj rad koristan i treba li ga pročitati.

**Ključne riječi**: riječ1, riječ2, riječ3, riječ4.

# ABSTRACT

**Title**:

Tekst sažetka na engleskom.

**Keywords**: word1, word 2, word3, word4.

# PRILOZI

Ako je potrebno, u rad možete dodati i priloge koji nisu neophodni za razumijevanje rada, ali smatrate da bi ih bilo dobro prikazati (tablice, grafikoni, nacrti i sl.).Bitno je da je svaki prilog **jasno opisan i razumljiv** i da je **radu potreban** (u priloge ne treba stavljati bespotrebne stvari).

**IZJAVA O AUTORSTVU ZAVRŠNOG RADA**

Pod punom odgovornošću izjavljujem da sam ovaj rad izradio/la samostalno, poštujući načela akademske čestitosti, pravila struke te pravila i norme standardnog hrvatskog jezika. Rad je moje autorsko djelo i svi su preuzeti citati i parafraze u njemu primjereno označeni.

|  |  |  |
| --- | --- | --- |
| Mjesto i datum | Ime i prezime studenta/ice | Potpis studenta/ice |
| U Bjelovaru,\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |  |

Ovu stranicu isprintajte, **ispunite i potpišite** te skenirajte (prije nego što uvežete u rad) kako biste ju ispunjenu mogli umetnuti u elektroničku inačicu rada.

U skladu s čl. 58, st. 5 Zakona o visokom obrazovanju i znanstvenoj djelatnosti, Veleučilište u Bjelovaru dužno je u roku od 30 dana od dana obrane završnog rada objaviti elektroničke inačice završnih radova studenata Veleučilišta u Bjelovaru u nacionalnom repozitoriju.

Suglasnost za pravo pristupa elektroničkoj inačici završnog rada u nacionalnom repozitoriju

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ime i prezime studenta/ice*

Dajem suglasnost da tekst mojeg završnog rada u repozitorij Nacionalne i sveučilišne knjižnice u Zagrebu bude pohranjen s pravom pristupa (zaokružiti jedno od ponuđenog):

1. Rad javno dostupan
2. Rad javno dostupan nakon \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (upisati datum)
3. Rad dostupan svim korisnicima iz sustava znanosti i visokog obrazovanja RH
4. Rad dostupan samo korisnicima matične ustanove (Veleučilište u Bjelovaru)
5. Rad nije dostupan

Svojim potpisom potvrđujem istovjetnost tiskane i elektroničke inačice završnog rada.

U Bjelovaru, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*potpis studenta/ice*

Obrazac najprije ispuniti i skenirati (da biste mogli umetnuti u elektroničke verzije rada), a potom umetnuti i u papirnatu verziju. Preuzeti ga možete na internetskoj stranici VUB-a.

1. Spool – Prostor SDSF-a za pregled aktivnih i neaktivnih programa [↑](#footnote-ref-2)
2. Z – IBM Mainframe instalacije [↑](#footnote-ref-3)