

# REALIZACIJA KLIJENT - SERVER ARHITEKTURE U PROGRAMSKOM JEZIKU JAVA NA PRIMERU INFORMACIONOG SISTEMA BIBLIOTEKE<sup>1</sup>

## REALISATION OF CLIENT - SERVER ARCHITECTURE USING JAVA PROGRAMING LANGUAGE IN THE CASE OF LIBRARY IFORMATION SYSTEM

Milan Zdravkovi<sup>2</sup>, Miroslav Trajanovi<sup>3</sup>, Nikola Korunovi<sup>4</sup>

*Sadržaj: Zahvaljujući razvoju Internet tehnologija, otkriveno je novo okruženje za korišćenje distribuiranih baza podataka primenom klijent-server arhitekture. Korišćenjem programskog jezika JAVA, omogučen je razvoj pouzdane, sigurne i brze klijent-server aplikacije koja se koristi u heterogenom TCP/IP okruženju, kao što je pokazano na primeru informacionog sistema biblioteke Mašinskog fakulteta u Nišu.*

*Abstract: Due to a development of Internet technologies, a new environment has been discovered, for using a distributed databases in a client-server architecture. By using JAVA programming language, it's easy to develop robust, secure and fast client-server application that will work in heterogeneous TCP/IP environment as shown in an example of information system of a Faculty of Mechanical Engineering library.*

### 1. UVOD

**Informacioni sistem biblioteke Mašinskog fakulteta u Nišu** je nastao na bazi potrebe interaktivnog pristupa resursima biblioteke od strane predavača i saradnika fakulteta, u cilju bržeg nalaženja potrebne informacije. Razvoj se odvijao u 3 faze: *projektovanje administratorske aplikacije*, uz pomoć Microsoft Access alata; *unos podataka o resursima biblioteke* sa papirnatih dokumentacije; *projektovanje mrežne aplikacije* i njen probni rad.

U ovom radu su prikazane mogućnosti implementacije klijent-server arhitekture u programskom jeziku JAVA, i dat je opis

realizacije informacionog sistema biblioteke Mašinskog fakulteta u Nišu.

### 2. KLIJENT-SERVER ARHITEKTURA I DISTRIBUIRANE BAZE PODATAKA

**Obrada podataka po modelu klijent - server** se odvija u mrežnom (LAN, WAN, Internet) okruženju, putem definisanih mrežnih protokola (Novell, TCP/IP), pri čemu se svaki tip mrežne arhitekture karakteriše nekim specifičnostima primene. Baze podataka u klijent - server sistemima se nazivaju još i *distribuirane baze podataka*.

Klijent - server sistem se sastoji od tri komponente: *server baze podataka*, *klijent aplikacija* i *mreža*. Na slici 2.1, prikazana je uprošćena struktura klijent - server arhitekture. **Server baze podataka** predstavlja složenu aplikaciju koja se stara o očuvanju integriteta podataka i poštovanju "pravila poslovanja", koja se zadaju deklarativno. Ovi restriktivni uslovi se zadaju *deklarativnim sredstvima*, odnosno bez programiranja, što olakšava i ubrzava administriranje servera baze podataka.

Pored toga, server mora da obezbedi paralelnu obradu transakcija (multi-threading), odnosno, upravljanje bazom kojoj istovremeno pristupa više korisnika. Budućnost baza podataka predstavljaju otvoreni sistemi; u tom smislu, server baza podataka mora posedovati interfejs za povezivanje sa drugim DBMS. *Interfejs prema drugim bazama podataka* ili DBC (Database Connectivity) omogućava komunikaciju između različitih DBMS.

<sup>1</sup> Ovaj rad je deo inovacionog projekta I.1.1280 pod nazivom: "Razvoj Web servera sa informacijama o proizvodima i standardnim delovima" koga finansira Ministarstvo za nauku i tehnologiju Republike Srbije.

<sup>2</sup> Milan Zdravković, dipl.maš.inž., saradnik Mašinskog fakulteta u Nišu

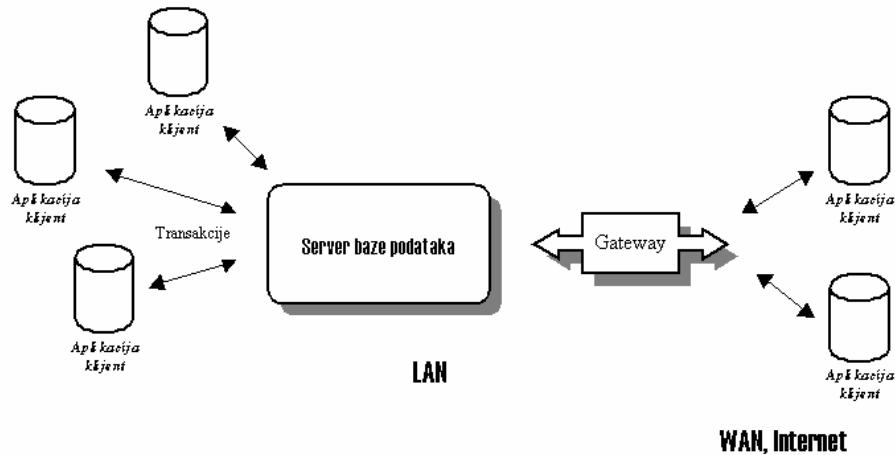
<sup>3</sup> dr Trajanović Miroslav, docent Mašinskog fakulteta u Nišu

<sup>4</sup> Nikola Korunović, dipl.maš.inž., saradnik Mašinskog fakulteta u Nišu

Komunikacija se vrši posredstvom SQL rešenica.

**Klijent aplikacija** je deo DBMS koji korisnicima omogućava rad sa podacima. Sastoji se od korisničkog interfejsa i modula za ostvarivanje veze sa serverom baza podataka. Zahtev za željenom transakcijom se programski definiše u obliku SQL rešenica.

Zavisno od implementacije mreže i tipa klijent-server arhitekture, veza sa serverom baza podataka uspostavlja odgovarajući programski modul. U TCP/IP mreži, povezivanje sa serverom se vrši putem soketa (sockets), datagrama i HTTP funkcijama.



Slika 2.1 Arhitektura klijent - server sistema

SQL (structured query language) predstavlja univerzalni jezik interpreter za manipulaciju i modifikaciju podataka u DBMS. Skoro svi DBMS podržavaju neki od nivoa ANSI/ISO standarda za SQL, što omogućava korisnicima i aplikacijama da rade na istom načinu sa različitim serverima baza podataka.

**Mreža** i njena implementacija uslovljavaju transportne protokole za razmenu informacija. Preovladavajući tip mreže, uslovljen naglim usponom Interneta je TCP/IP (Transport Control Protocol/Internet Protocol).

### 3. IMPLEMENTACIJA KLIJENT-SERVER ARHITEKTURE U PROGRAMSKOM JEZIKU JAVA

**Programski jezik JAVA** je projektovan sa ciljem da odgovori izazovu razvoja aplikacija u heterogenom okruženju, nastalom naglim razvojem mrežnih tehnologija i ekspanzijom Interneta. Osnovne osobine JAVA razvojnog okruženja su:

- ✓ *Objektno-orijentisano programiranje* - razvojno okruženje JAVA je nastalo na temelju programskih jezika Eiffel, SmallTalk, Objective C, i poseduje sintaksu koja se vrlo malo razlikuje od sintakse primenjene u jeziku C++.
- ✓ *Interpretirani kod* - razvojni put do gotove aplikacije se dramatično skratio, jer se kod samo kompajlira.

- ✓ *Portabilnost* - projektovane aplikacije su prenosive, nezavisno od operativnog sistema i hardverske platforme. U poslednje vreme, uočena su izvesna neslaganja u deklaracijama JDK standarda koje nezavisno nameću kompanije Microsoft i Sun Microsystems.

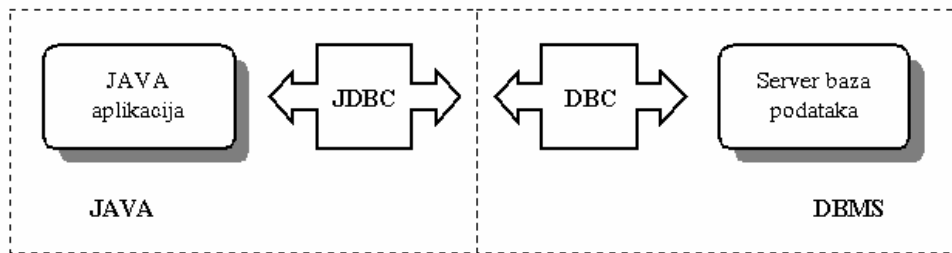
- ✓ *Pouzdanost* - JAVA poseduje sopstveni run-time sistem za upravljanje memorijom.

- ✓ *Visoke performanse* - multithreading osobina JAVA programskog jezika omogućava istovremen i nezavistan rad nekoliko konkurentnih niti aplikacije.

JAVA je programski jezik, projektovan za upotrebu u distribuiranom, klijent - server okruženju. Iz tog razloga, predstavlja logičan izbor za razvojnu platformu pri izradi kompletnog informacionog sistema u kojem se podaci razmenjuju posredstvom TCP/IP protokola.

Interfejs koji služi za komunikaciju između JAVA programskog koda i DBMS se naziva JDBC (JAVA Database Connectivity).

Na slici 3.1, prikazana je uprošćena šema povezivanja JAVA programskog koda i sistema za upravljanje bazama podataka. Komunikacija između JAVA aplikacije i baze podataka se vrši putem JDBC - DBC mosta i to posredstvom SQL rešenica.



Slika 3.1 Povezivanje JAVA aplikacije i proizvoljnog servera baza podataka

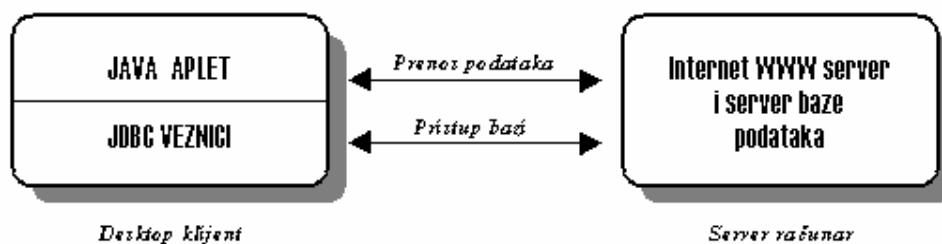
JDBC je još uvek call-level interfejs, što znači da unutar JAVA klasa ne postoji DBMS jezgro sa sopstvenom SQL sintaksom, već se JDBC koristi samo za uspostavljanje veze sa bazom podataka koja interpretira SQL rečenicu, koja zato, mora biti napisana po specifikaciji koju podržava DBMS.

JDBC, dakle, dopušta da se ka DBMS interfejsu (DBC) pošalje bilo koji string. Ispravnost njegove sintakse zavisi od implementacije DBMS.

U realizaciji klijent - server arhitekture primenom programskog jezika JAVA, može se primeniti jedna od tri arhitekture: aplet - klijent, aplikacija - klijent i troslojna (three-tier) arhitektura.

### Aplet - klijent

Najširu upotrebu Jave predstavljaju apleti, koji se prenose preko Interneta, kao delovi web stranica. U okviru apleta se nalazi konfigurisan JDBC interfejs prema definisanom DBMS. Ostvaruje se direktna veza aplet - server baze podataka.



Slika 3.2. JAVA implementacija Klijent-server arhitekture sa direktnim pristupom

Ovakva upotreba je pogodna u uslovima kada se serveru baze podataka pristupa preko Interneta (preko gateway-a lokalne mreže, na kojoj se nalazi server). Medjutim, aplet se može pokupiti i na lokalnoj mreži, na kojoj, sigurnost klijent računara može biti problematična.

Tipični apleti se razlikuju od klasičnih aplikacija za upravljanje bazama podataka u sledećim tačkama:

1. Apleti su vrlo ograničeni u operacijama koje mogu izvršavati. Konkretno, oni nemaju pristup lokalnim datotekama (na disku klijent-računara) i ne mogu uspostaviti vezu sa proizvoljnim host računarima.
2. vreme pristupa preko Interneta je mnogo veće nego u lokalnoj mreži.

### Aplikacija - klijent

Ako se serveru baze podataka pristupa uz pomoć aplikacije koja se nalazi na klijent računaru, JAVA kod je poverljiv (trusted) i moguće je pristup lokalnim datotekama i uspostavljanje veze sa proizvoljnim host računarima.

Ovakva primena je preporučena u lokalnim mrežama preduzeća (intranet). Npr. ako se baza podataka nalazi na serveru preduzeća, upotrebom JAVA alata za kreiranje GUI, mogu se napraviti forme za pristup serveru iz Intranet-a ili WAN preduzeća. Medjutim, korišćenjem ovih aplikacija, pristup serveru je moguće i putem Interneta.



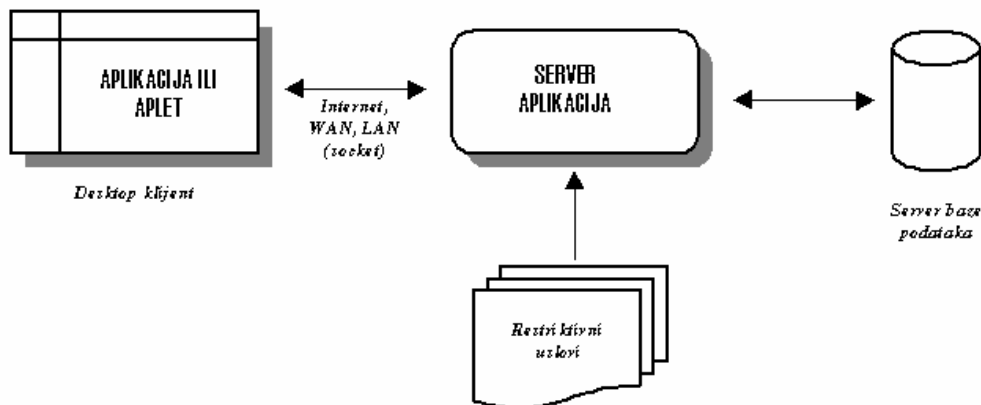
Slika 3.3 JAVA implementacija klijent-server arhitekture sa distribuiranom aplikacijom

### Troslojna arhitektura

- Three Tier pristup - JAVA aplikacije ili apleti pristupaju server aplikaciji (srednji sloj), koja pristupa bazi. Pristup server aplikaciji može biti ostvaren i kroz RPC (remote procedure call) ili ORB (object request broker), gde se pozivaju metode u programima napisanim u bilo kom programskom jeziku koji podržava CORBA standard.

Koncept koji ovo omogu•ava se naziva RMI ili Remote Method Invocation i predstavlja osnovu distribuiranog programiranja.

Dobra strana troslojnog pristupa je da se u server aplikaciji mogu definisati restriktivna pravila za pristup serveru baze podataka. Takodje, u mnogim slu•ajevima, performanse ovog sistema u pogledu brzine su najbolje.



Slika 3.4 JAVA implementacija klijent-server troslojne arhitekture

## 4. REALIZACIJA INFORMACIONOG SISTEMA BIBLIOTEKE MAŠINSKOG FAKULTETA U NIJU

Informacioni sistem biblioteke Mašinskog fakulteta je nastao iz potrebe interaktivnog koriš•enja resursa biblioteke u lokalnoj mreži fakulteta i to na temelju Microsoft Access baze podataka, koja se koristi pri unosu, pregledu i izdavanju inventarskih jedinica. Ova baza se nalazi na PC ra•unaru u prostorijama biblioteke.

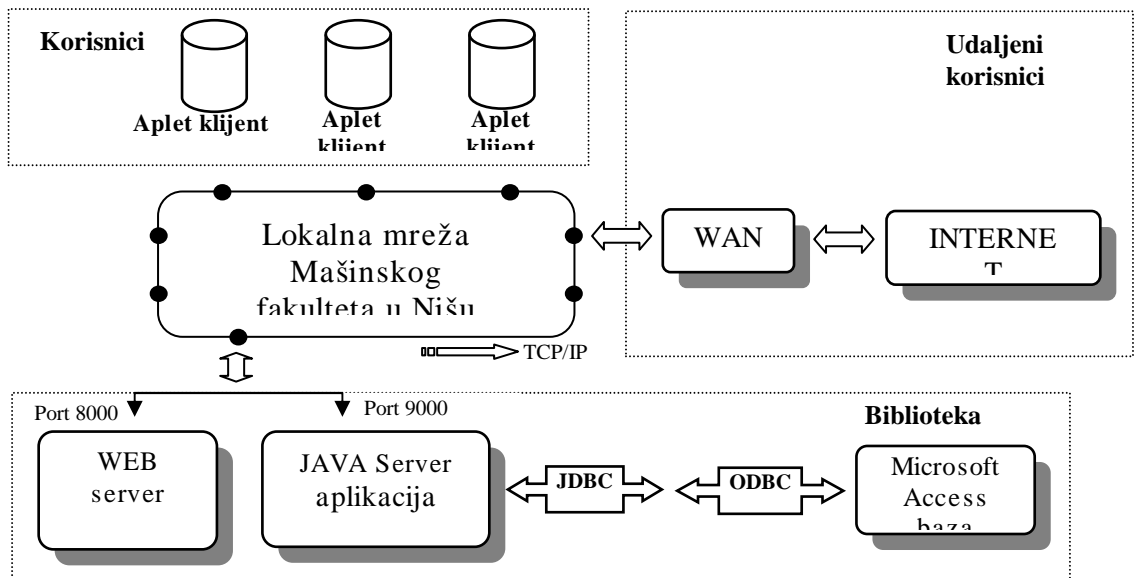
Zbog karakteristika brzine pristupa, jednostavnog administriranja i distribuiranja, za programsku realizaciju klijent - server arhitekture, izabran je troslojni (three-tier) pristup, implementiran uz pomo• JAVA razvojnog alata.

Na slici 4.1. prikazana je blok šema programske realizacije informacionog sistema biblioteke i to na bazi TCP/IP protokola.

Kao hardverska osnova za projektovanje informacionog sistema, koristi se lokalna mreža Mašinskog fakulteta, zasnovana na TCP/IP protokolu.

Server baze podataka je PC ra•unar sa procesorom Pentium/133MHz, koji se nalazi u prostorijama biblioteke. Kontrolu protoka informacija i transakcija obavljaju dve server aplikacije:

- ✓ *HTTP server* - vrši distribuciju aplet - klijenta preko WEB stranice biblioteke Mašinskog fakulteta. Osluškujе na portu 8000 - standardnom za HTTP zahteve.
- ✓ *JAVA server* - obavlja transakcije. Na portu 9000, preko otvorenog soketa, osluškujе zahteve, koji sa klijenta stižu putem TCP/IP



protokola i to u obliku SQL rećenica, koji se u slu•aju zadovoljenja restriktivnih uslova, izvršavaju posredstvom JDBC-ODBC mosta.

Klijente predstavljaju svi ra•unari povezani na Internet. Klijent-ra•unar mora imati instaliran WEB pretraživa• ( Netscape 3.0 ili Microsoft

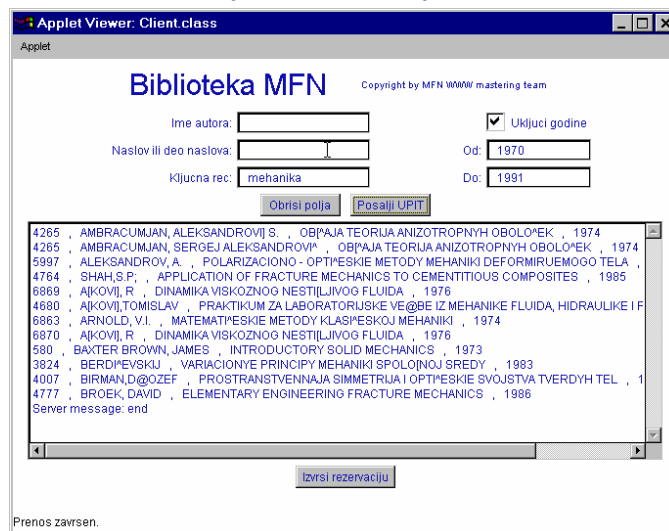
Internet Explorer 3.0) sa ugradjenim JAVA interpretrom (JAVA VM - Virtual Machine).

**Aplet-klijent** se sastoji iz korisni•kog interfejsa, modula za generisanje SQL iskaza i modula za komunikaciju sa JAVA server aplikacijom.

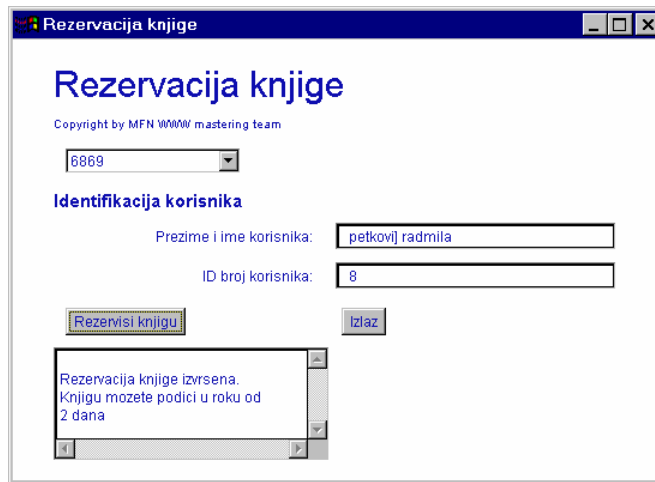
#### Sl.4.1 Programska realizacija informacionog sistema biblioteke Mašinskog fakulteta u Nišu

✓ *Korisni•ki interfejs* objedinjava niz elemenata potrebnih za jednozna•no definisanje korisni•kih zahteva. Projektovani su korisni•ki interfejs za

pregled baze podataka, pomo•u kojeg se definišu kriterijumi pretraživanja i korisni•ki interfejs za rezervaciju inventarske jedinice.



4.2 Korisni•ki interfejs za pregled baze knjiga



#### 4.3 Korisni•ki interfejs za autorizaciju korisnika i rezervaciju knjige

- ✓ *Modul za generisanje SQL iskaza* koristi parametre unete u odgovaraju•im ulaznim elementima korisni•kog interfejsa kao argumente SQL iskaza SELECT (pretraživanje) i INSERT INTO (rezervacija).
- ✓ *Modul za komunikaciju* vrši povezivanje sa JAVA server aplikacijom preko otvorenog soketa na portu 9000 i kontrolu transfera SQL re•enica i rezultata njihovog izvršavanja.

**Aplikacija-server** vrši funkcije komunikacije sa aplet-klijentom, izvršavanja transakcija preko JDBC-ODBC mosta i preusmeravanja rezultata ka klijentu uz zadovoljenje *restriktivnih uslova*. JAVA server je multi-threading aplikacija i omogu•ava istovremeni pristup bazi podataka ve•em broju korisnika.

- ✓ *Restriktivni uslovi* predstavljaju niz uslova koji moraju biti zadovoljeni da bi se transakcija uspešno izvršila.
  - ◆ *Uslovi integriteta* baze podataka su definisani tipom podatka u strukturi baze podataka.
  - ◆ Restriktivnim uslovima se definišu i *sintaksna ograni•enja* mogu•eg SQL iskaza tako da se onemogu•ava izvršenje destruktivnih SQL iskaza (DROP TABLE, DELETE ROW).
  - ◆ Vršiti se kontrola *prava pristupa i izmene* tabela u bazi podataka. Prava pristupa su definisana u okviru kategorizacije korisnika.

**Baza podataka** predstavlja jezgro informacionog sistema. Podaci dostupni aplet klijentu se nalaze u tabelama *Knjige, Clanci, Diplomski radovi, Magistarski radovi i Doktorske disertacije*. Kategorizacija korisnika

sa odgovaraju•im pravima pristupa i izmene je obavljena u tabeli *Korisnici*.

Pri rezervaciji inventarske jedinice, najpre se vrši provera prava pristupa i izmene, na osnovu autorizacije korisnika preko imena i prezimena i ID broja, definisanog u tabeli *Korisnici*. U slu•aju autorizovanog pristupa, vrši se upis odgovaraju•ih podataka u tabelu *Rezervacije*. Administriranje baze podataka vrši operater u biblioteci.

## 5. ZAKLJU^AK

Navedene osobine programskog jezika JAVA, name•u ga kao jedini izbor pri razvoju sistema za upravljanje distribuiranim bazama podataka i uopšte, aplikacija koje rade u heterogenom mrežnom okruženju. U projektovanju i eksperimentalnom radu informacionog sistema biblioteke Mašinskog fakulteta u Nišu, uo•ene su slede•e karakteristike:

- ✓ **Malo angažovanje resursa servera•unara**
  - ◆ Efikasni run-time sistem za upravljanje memorijom, integrisan u kod JAVA aplikacije, obezbedjuje visoku pouzdanost u radu.
  - ◆ Multithreading karakteristike JAVA interpretera omogu•uju ravnomernu raspodelu resursa medju korisnicima koji istovremeno pristupaju bazi podataka.
- ✓ **Jednostavnost**
  - ◆ Definisane ciljeva, projektovanje i programiranje informacionog sistema je obavljeno u kratkom roku.

- ◆ Aplet klijent se sastoji od dve datoteke sa ukupno 420 linija programskog koda. Nakon kompajliranja, ukupna veličina apleta iznosi 10 KB.

✓ **Sigurnost**

- ◆ Ugradjeni sigurnosni sistemi omogućavaju poverljivost transakcija.
- ◆ Konceptija troslojne (three-tier) arhitekture dozvoljava definisanje restriktivnih uslova, kojima se baza podataka zaštićuje od neautorizovanog pristupa.

**5. LITERATURA**

1. Bobrowski S., "Oracle 7 i obrada podataka po modelu klijent/server", Mikro Knjiga, SYBEX, Beograd, 1995
2. JDK 1.1.4 Dokumentacija, HTML, Sun Microsystems, Inc.
3. Mitrović I., "JDBC", članak, časopis "PC PRESS", oktobar, Beograd, 1997
4. Rogers Cadenhead, "Java 1.1 Programming", Sams net, Indianapolis 1997
5. <http://www.javaworld.com>
6. <http://www.javashareware.com>