

JAVA 2

RMI

Teme

- RMI okvir
- RMI registar
- *Remote interface*
- Korisničko sučelje
- RMI komunikacija

RMI okvir

- RMI – *remote method invocation* – pozivanje udaljenih metoda
- RMI okvir - omogućuje komunikaciju između Java aplikacija koje se izvršavaju unutar različitih Java virtualnih mašina (JVM)
- standardni mehanizam RPC (*remote procedure call*) sustava:
 - stub – lokalna reprezentacija udaljenog objekta na klijentu
 - skeleton – serverski proxy koji prosljeđuje pozive konkretnoj implementaciji
 - protokol komunikacije - *interface*

RMI registar

- u sklopu JDK paketa
- usluga imenovanja pomoću koje poslužitelj registrira objekt čije će metode klijenti pozivati (*remote object*)
- poslužitelj objavljuje objekt u RMI registru, određenog imena
- klijent pretražuje RMI registar u potrazi za objektom, prema imenu
- startanje RMI *registry* servisa:
 - ✓ `c:{path}>rmiregistry`
 - ✓ `LocateRegistry.createRegistry()`

Remote interface

- *marker interface*
- srce RMI arhitekture
- svaki objekt čije metode želimo pozivati iz RMI klijenata mora implementirati sučelje *Remote*
- praksa je takva da se kreira korisničko sučelje koje nasljeđuje sučelje *Remote* i u njemu se deklariraju metode koje će se kasnije pozivati od strane RMI klijenata
- sve metode takvog korisničkog sučelja moraju deklarirati mogućost bacanja iznimke tipa *java.rmi.RemoteException*
- implementacijska klasa je preduvjet kreiranju konkretnih objekata nad kojim će se pozivati udaljene metode

Korisničko sučelje

- kako će se klijentska aplikacija kompajlirati, obzirom da je Java strogo tipizirani jezik?
1. korisničko sučelje poslužiteljske aplikacije arhivira se u *jar*

jar cvf remoteservice.jar {path}/RemoteService.class

2. *jar* se fizički stavlja u klijentsku aplikaciju i u *classpath*

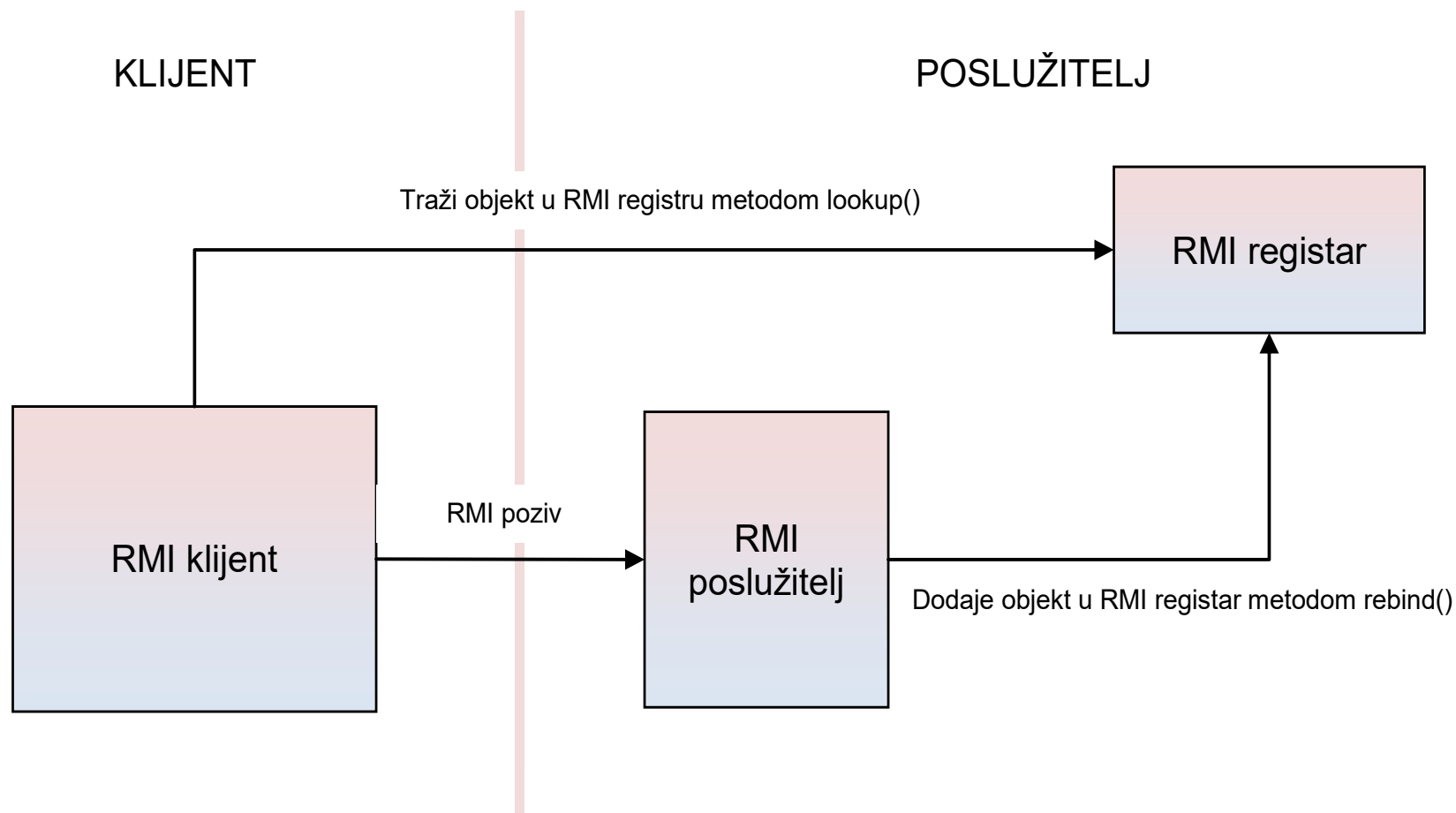
RMI komunikacija

1. start RMI registra ili poslužitelj *kreira* RMI registar na *localhostu*
2. poslužitelj objavljuje objekt u RMI registru
3. klijent pronalazi objekt u RMI registru – dobiva njegovu lokalnu reprezentaciju – *stub*:
 - ima iste metode kao i objekt na poslužitelju, što je osigurano implementiranim sučeljem, ali njegove metode ne implementiraju poslovnu logiku, nego logiku kojom se provodi komunikacija klijenta i poslužitelja
 - skriva serijalizaciju i mrežnu komunikaciju tako da se poziv doima lokalnim

RMI komunikacija

4. klijent poziva metodu – RMI implementacija serijalizira argumente i mrežom dolazi do implementacije *skeletona* na poslužiteljskoj JVM
5. *skeleton* prosljeđuje poziv stvarnoj implementaciji objekta u JVM
6. objekt u JVM izvrši metodu i vrati povratne vrijednosti
7. *skeleton* serijalizira povratne vrijednosti i mrežom ih vraća *stubu*
8. *stub* prosljeđuje podatke klijentu

RMI komunikacija



Demo

- Project



Izvor:<http://www.jnhsolutions.com/contact-us/request-a-demo/>

Hvala na pažnji!

