

Универзитет у Београду
Електротехнички факултет



Домаћи задатак ЈУН 2009 из предмета
Конкурентно и Дистрибуирано Програмирање

ДИСТРИБУИРАНА ОБРАДА МАТЕМАТИЧКИХ ИЗРАЗА

Прва верзија: 03. 06. 2009.

Студент
Татјана Петровић 94/0394

1. ТЕКСТ ЗАДАТКА	3
2. ФУНКЦИОНАЛНА СПЕЦИФИКАЦИЈА	3
2.1 Сервер	3
2.2 Клијент	4
2.3 Радна станица	4
3. ДЕТАЉАН ОПИС ПАКЕТА	4
3.1 Дијаграми класа	4
3.1.1 Клијентска апликација	4
3.1.2 Серверска апликација	6
3.1.3 Апликација радне станице	7
3.2 Дијаграми интеракција	9
3.2.1 Задавање посла на клијенту	9
3.2.2 Провера стања послова од више клијената	9
3.2.3 Повезивање радних станица са централним сервером. Провера присутности и стања самих радних станица са сервера	10
3.2.4 Контролер примљених послова на серверу је задужен за праћење рада свих послова и њихово поновно стартовање по потреби.	11
3.2.5 Прихват посла на радној станици. Позивање библиотеке за паралелно процесирање математичких израза.	12
3.3 Дијаграм стања посла после пријема на сервер	12
3.4 Опис протокола	12
4. КОРИСНИЧКО УПУТСТВО	14
4.1 Упутство за коришћење апликација система	14
4.1.1. Серверска апликација	14
4.1.2 Клијентска апликација	14
4.1.3 Апликација радне станице	15
4.2 Упутство за коришћење програмских пакета система	16
4.2.1 Серверска апликација	16
4.2.2 Клијентска апликација	16
4.2.3 Апликација радне станице	16

1. ТЕКСТ ЗАДАТКА

Пројектовати дистрибуирани рачунарски систем који треба да омогући дистрибуирану обраду и међусобну удаљену синхронизацију временски захтевних послова математичке обраде.

Програм треба да ради у систему који се састоји од више рачунара повезаних у LAN (Local Area Network) или WAN (Wide Area Network).

У систему постоји три типа програма:

1. Централни сервер који служи за праћење рада извршавања дистрибуиране обраде, чување информација о доступним чворовима у мрежи и могућност поновног стартовања појединих послова.
2. Радна станица која од централног сервера добија послове које треба одрадити.
3. Кориснички програм, који задаје посао који треба урадити и његове параметре.

Процес започиње тако што кориснички програм задаје посао који је потребно извршити. Овај посао представља математички израз који је потребно израчунати. Након тога кориснички програм контактира централни сервер коме прослеђује тај посао и параметре потребне за његову обраду. Када централни сервер прими посао и његове параметре он га прослеђује једној радној станици, чека резултат обраде и враћа комплетан резултат клијенту. Када радна станица прими посао, креира нит намењену извршавању тог посла, и у тој нити започиње са његовим извршавањем на основу примљених параметара. Посао се извршава на радној станици тако што покренута нит радне станице позива одговарајућу методу инстанце класе за паралелно процесирање математичких израза (имплементира интерфејс *Equations*). Да би се обављало паралелно процесирање потребно је прво иницијализовати ову инстанцу адресама и портovima преосталих класа за паралелно израчунавање који се налазе на осталим радним станицама (интерфејс *Connector*), програму за паралелно израчунавање је потребно доделити један слободни серверски порт (*setPort*) и покренути их свакој радној станици (метод *connect()*). Инстанцирање класа које раде математичку обраду и повезивање на радној станици обезбеђује се користећи статичке методе класе *Creator*. Када се заврши израчунавање математичких израза, радна станица прикупља излаз рачунања, пакује га у одговарајућу датотеку коју прослеђује серверу, као и резултате извршавања. Да би се обезбедило прикупљање информација о активним радним станицама централни сервер на сваких *x* секунди проверава да ли је нека радна станица исправна. Уколико сервер утврди да нека радна станица која је до тог тренутка обављала математичка израчунавања није више исправна, посао који је обављала та радна станица прослеђује некој слободној радној станици. Након слања захтева за обраду кориснички програм може да раскине везу са централним сервером. Веза може бити раскинута гашењем програма или затварањем комуникационог канала. Када се следећи

пут повеже кориснички програм може да тражи резултате претходно задате обраде. Треба обезбедити да централни сервер може да у паралели да прима већи број послова које је потребно обрадити. Кориснички програм може од централног сервера да тражи информације о статусу посла, а може да тражи и резултате. Одмах по стартовању радне станице шаљу централном серверу информацију о томе да су стартоване и број послова које могу у паралели да обрађују. Број послова које радне станице могу да приме је аргумент који им се поставља приликом покретања.

Параметри посла које клијент задаје су: математичка операција коју је потребно извршити, датотека у којој се налази корисникова матрица коју је потребно обрадити, датотека са низом вредности које је потребно још применити у операцији (параметар је опциони), име датотеке у коју је потребно сместити резултате операције коју је са радне станице треба пребацити на клијентски рачунар да које представљају резултате. Ови параметри могу да се задају или путем корисничког интерфејса или путем текстуалне датотеке.

Централни сервер у лог уписује време када је пристигао сваки посао, број под којим је посао сачуван, име рачунара коме је посао прослеђен, време када је посао завршен и његов тренутни статус. Статус посла може да буде: Ready – приспео на сервер али није никоме прослеђен, Scheduled – тренутно се прослеђује радној станици, Running – извршавање је у току, Done – посао се успешно извршио, Failed – посао није могао да се изврши (емитовао је неки изузетак), Aborted – корисник је одустао од извршавања посла.

Проблем решити користећи мрежну комуникацију у програмском језику Јава. Решење треба да буде независно од посла који се обавља. За сваки од ова три типа рачунара треба да постоји одговарајући графички кориснички интерфејс (GUI треба да буде развијен користећи Java **SWING** компоненте). Радна станица треба да има могућност покретања и без корисничког интерфејса.

2. ФУНКЦИОНАЛНА СПЕЦИФИКАЦИЈА

Пројектовани систем се састоји од три подсистема – Сервер, Клијент и Радна станица. Ова три подсистема могу бити покренута како на једном, тако и на више рачунара. Сваки од њих има посебан графички интерфејс за интеракцију са корисником.

2.1 Сервер

Серверска апликација прима послове од клијената, додељује примљеним пословима јединствене идентификационе бројеве и чува све информације о пословима све до њиховог завршетка. Сервер је задужена за прослеђивање послова одговарајућим

радним станицама. Такође је задужен за вођење евиденције о расположивим радним станицама у систему, као и о њиховим карактеристикама (броју послова које свака од расположивих радних станица може да обавља у паралели). По завршеном послу, сервер чува, локално, на хард диску, резултат посла све док га клијент не затражи. Сервер води рачуна о статусима посла и рестартује посао уколико је радна станица, којој је он додељен, постала недоступна. Сервер пише у лог када неки посао стигне или промени статус, као и радну станицу којој је тај посао прослеђен на извршавање. На тај начин је могуће вратити систем у првобитно стање и након рестартовања сервера. Због ових карактеристика сервера, на серверу су увек отворена два порта, један за комуникацију са клијентом и један на коме сервер ослушкује појављивање нове радне станице (пријаву радне станице на сервер) и врши комуникацију са свим радним станицама у систему.

3.1.1 Клијентска апликација

2.2 Клијент

Клијент шаље серверу посао који је потребно извршити са свим потребним подацима при чему му се враћа јединствен идентификациони број посла од стране сервера ако је посао прихваћен на серверу. Клијентска апликација има могућност праћења статуса свих послова које је послала серверу, а он их прихватио. По завршеном послу, клијент тражи од сервера резултат обраде. Такође, клијент има могућност заустављања посла који је послао серверу, а у том тренутку још увек није завршен.

2.3 Радна станица

Радна станица је апликација која је задужена за извршавање посла, од његовог прихватања од сервера па све до његовог успешног или неуспешног завршетка. При њеном стартовању се врши пријава на сервер после чега је она у стању чекања на посао од стране сервера. Када прими посао, она поставља неке статусе посла, у зависности од тога у којој је фази обраде у том тренутку посао. По завршетку посла, она шаље серверу резултат посла, уколико је успешно извршен посао, или одговарајућу поруку у супротном. Пре старовања сваке радне станице, потребно је задати број послова које може та радна станица да извршава истовремено у паралели.

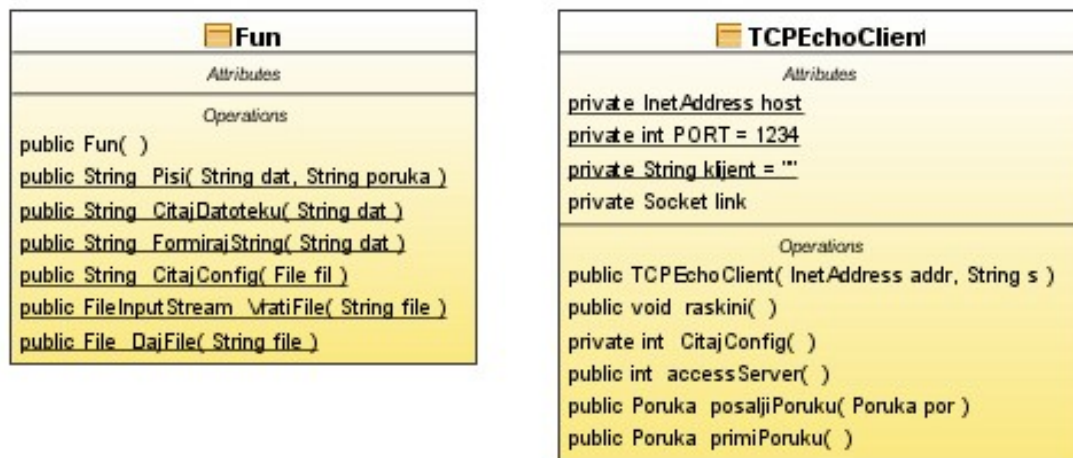
3. ДЕТАЉАН ОПИС ПАКЕТА

3.1 Дијаграми класа

Дијаграми класа су генерисани уз помоћ *Netbeans 6.5* окружења.

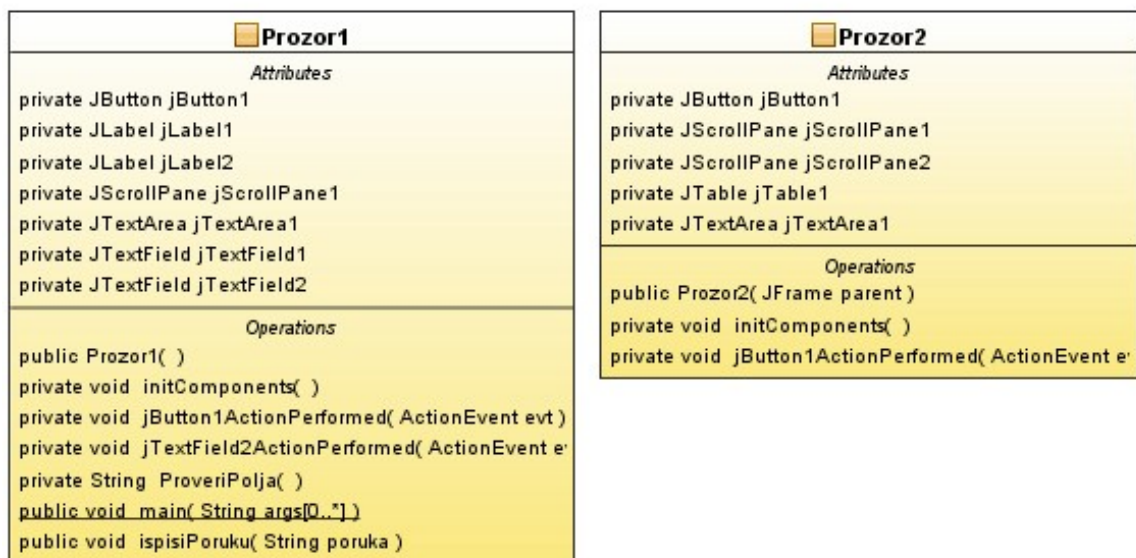
Prozor1	Prozor2
<p>Attributes</p> <pre> private JButton jButton1 private JLabel jLabel1 private JLabel jLabel2 private JLabel jLabel3 private JLabel jLabel4 private JTextField jTextField1 private JTextField jTextField2 </pre>	<p>Attributes</p> <pre> private JButton jButton1 private JButton jButton2 private JButton jButton3 private JButton jButton4 private JButton jButton5 private JLabel jLabel1 private JLabel jLabel2 private JLabel jLabel3 private JLabel jLabel4 private JLabel jLabel5 private JLabel jLabel6 private JPanel jPanel1 private JPanel jPanel2 private JPanel jPanel3 private JPanel jPanel4 private JScrollPane jScrollPane1 private JScrollPane jScrollPane2 private JScrollPane jScrollPane3 private JScrollPane jScrollPane4 private JScrollPane jScrollPane5 private JScrollPane jScrollPane6 private JTabbedPane jTabbedPane1 private JTable jTable1 private JTextArea jTextArea1 private JTextArea jTextArea2 private JTextArea jTextArea3 private JTextArea jTextArea4 private JTextArea jTextArea5 private JTextField jTextField1 private JTextField jTextField2 private JTextField jTextField3 private JTextField jTextField4 private JTextField jTextField5 private JTextField jTextField6 </pre>
<p>Operations</p> <pre> public Prozor1() private void initComponents() private void jButton1ActionPerformed(ActionEvent evt) private String ProveriPolja() public void main(String args[0..*]) </pre>	<p>Operations</p> <pre> public Prozor2(JFrame parent, TCPEchoClient client) private void initComponents() private void jButton1ActionPerformed(ActionEvent evt) private void zatvaranje(WindowEvent evt) private void jButton2ActionPerformed(ActionEvent evt) private void jButton3ActionPerformed(ActionEvent evt) private void jTextField5ActionPerformed(ActionEvent evt) private void jButton4ActionPerformed(ActionEvent evt) private void jButton5ActionPerformed(ActionEvent evt) </pre>

Слика 1 – пакет *Prozori*

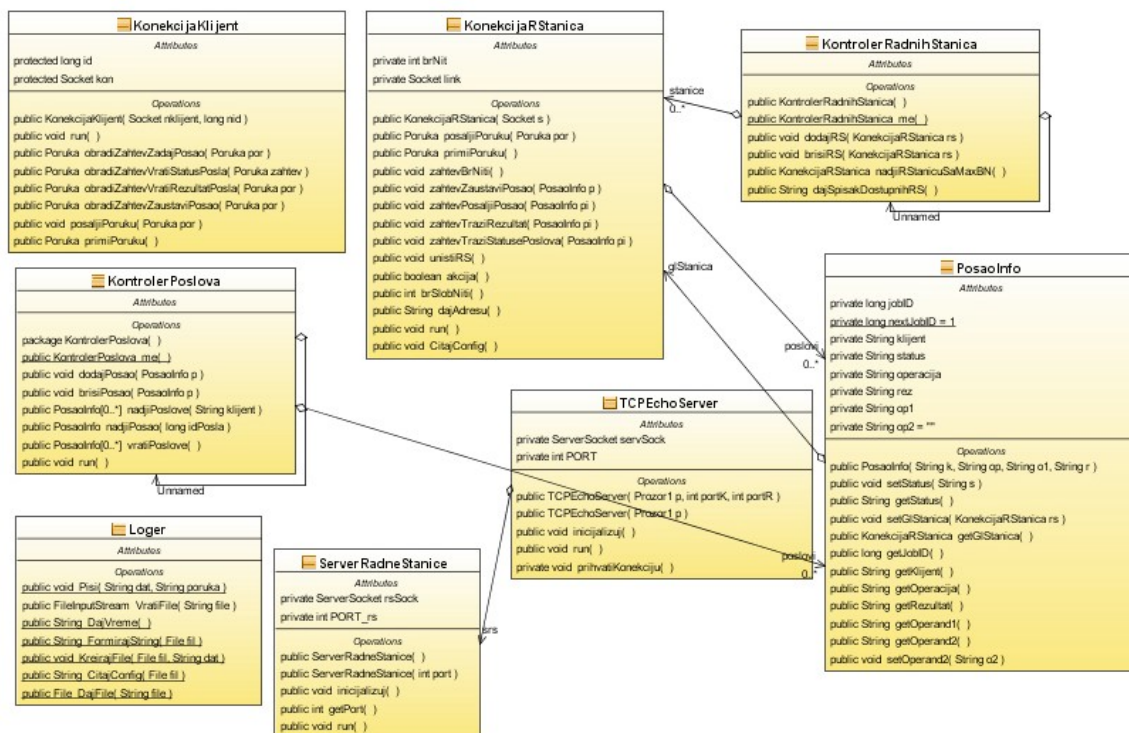


Слика 2 – пакет *Klase*

3.1.2 Серверска апликација

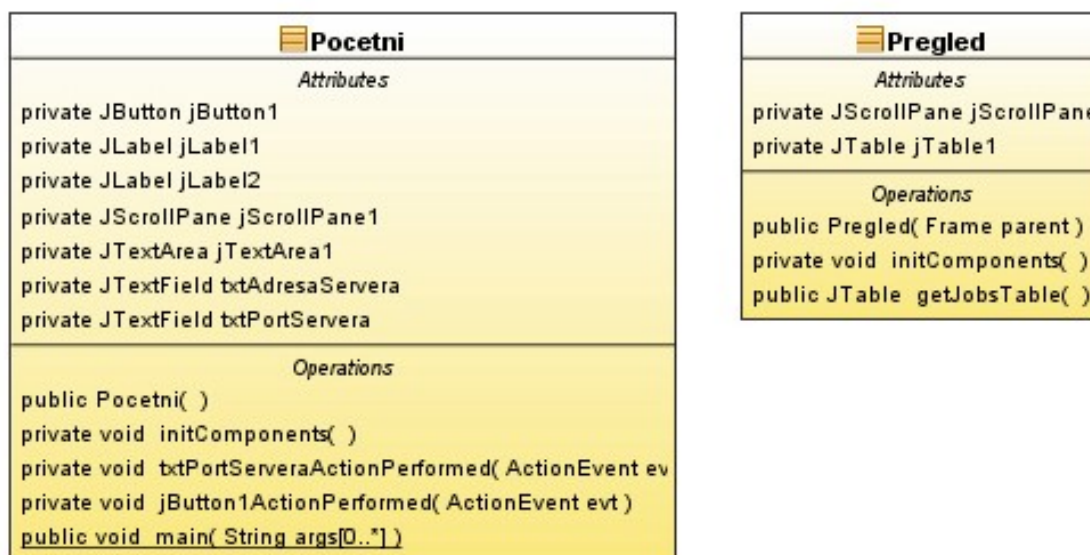


Слика 3 – пакет *Prozori*

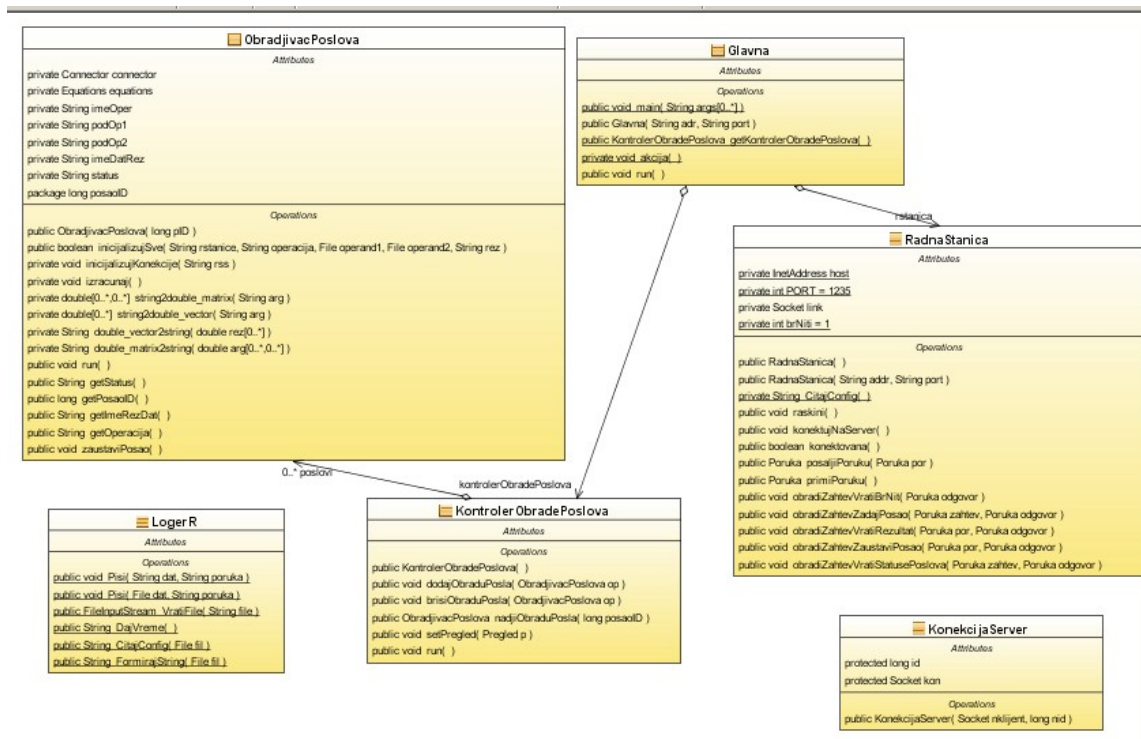


Слика 4 – пакет *paketServer*

3.1.3 Апликација радне станице



Слика 5 – пакет *GUIStanica*

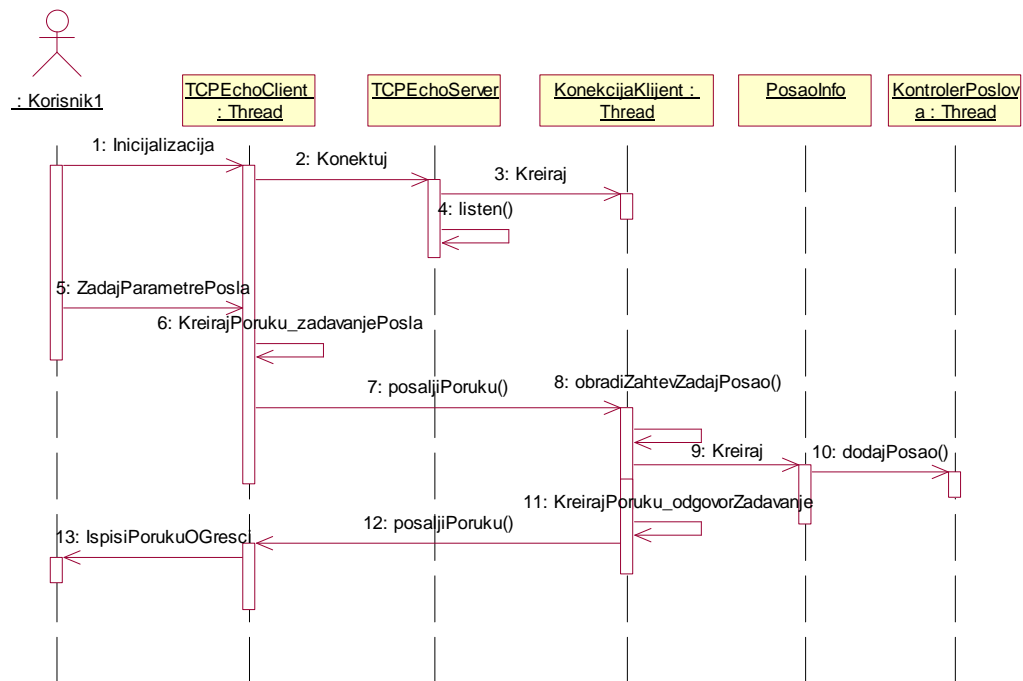


Слика 6 – пакет *RadnaStanica*

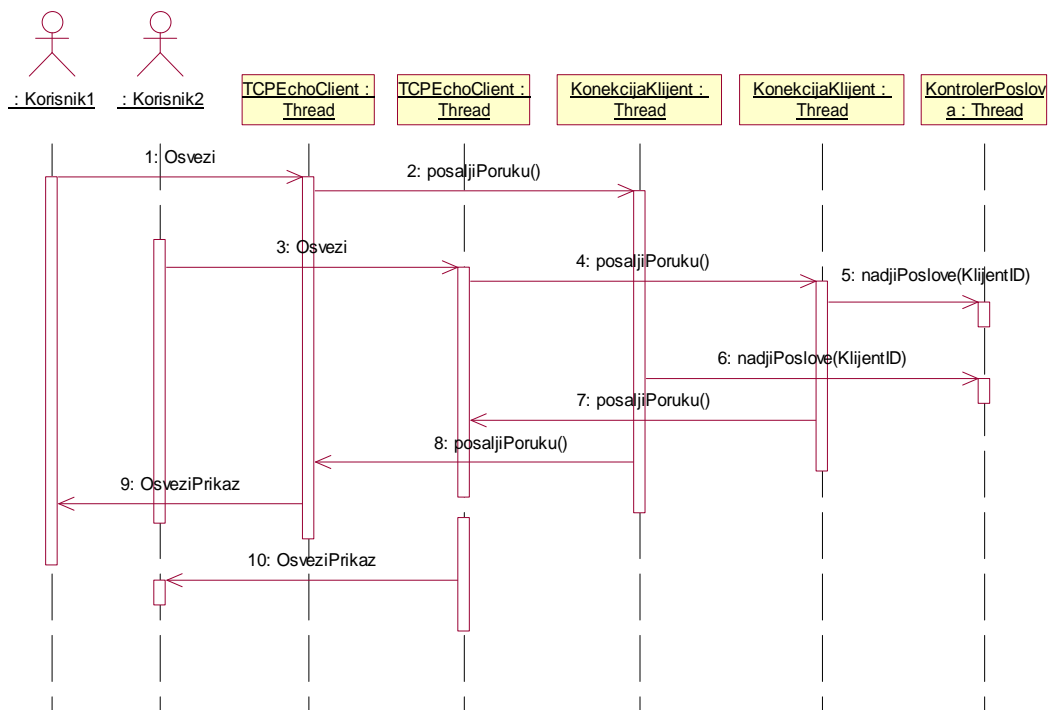
3.2 Дијаграми интеракција

Дијаграми класа су генерисани уз помоћ *RationalRose* окружења.

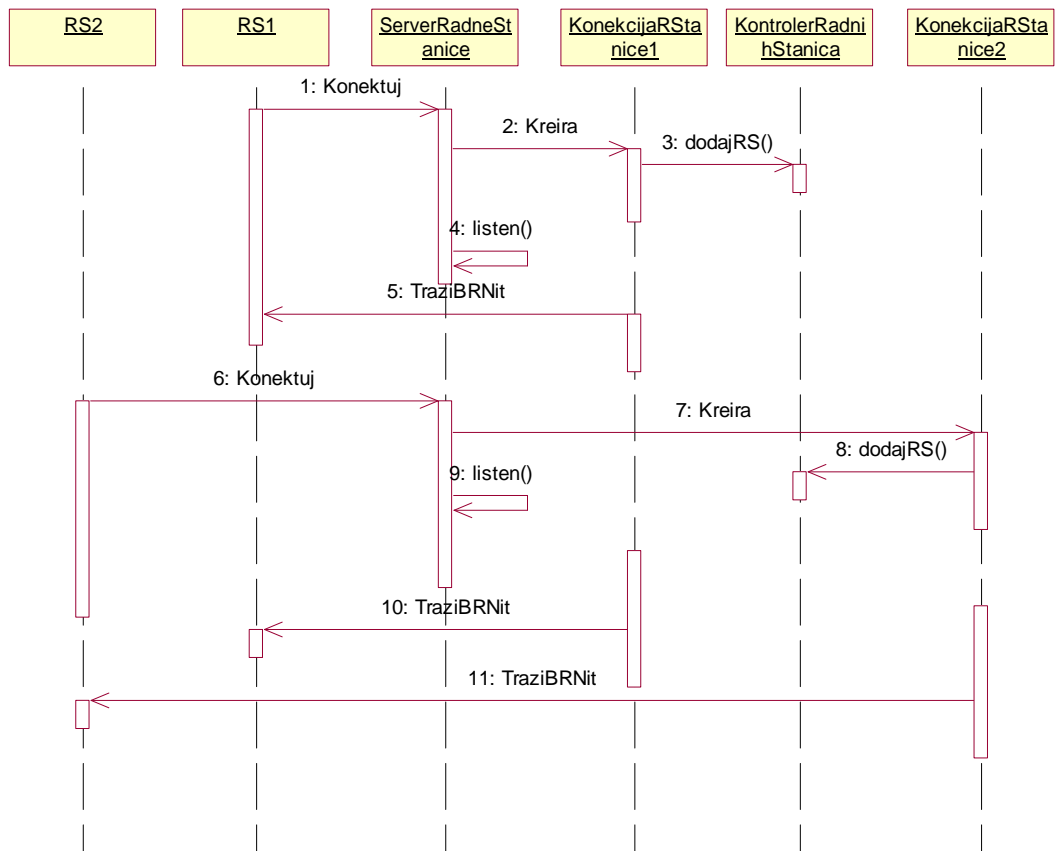
3.2.1 Задавање посла на клијенту



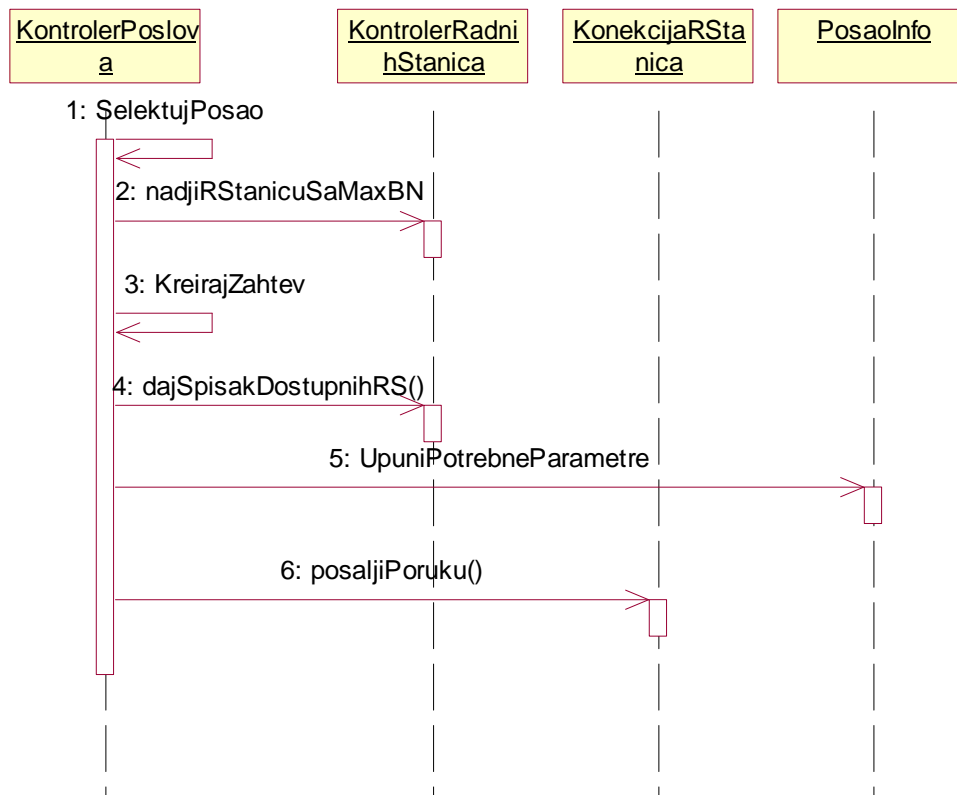
3.2.2 Провера стања послова од више клијената



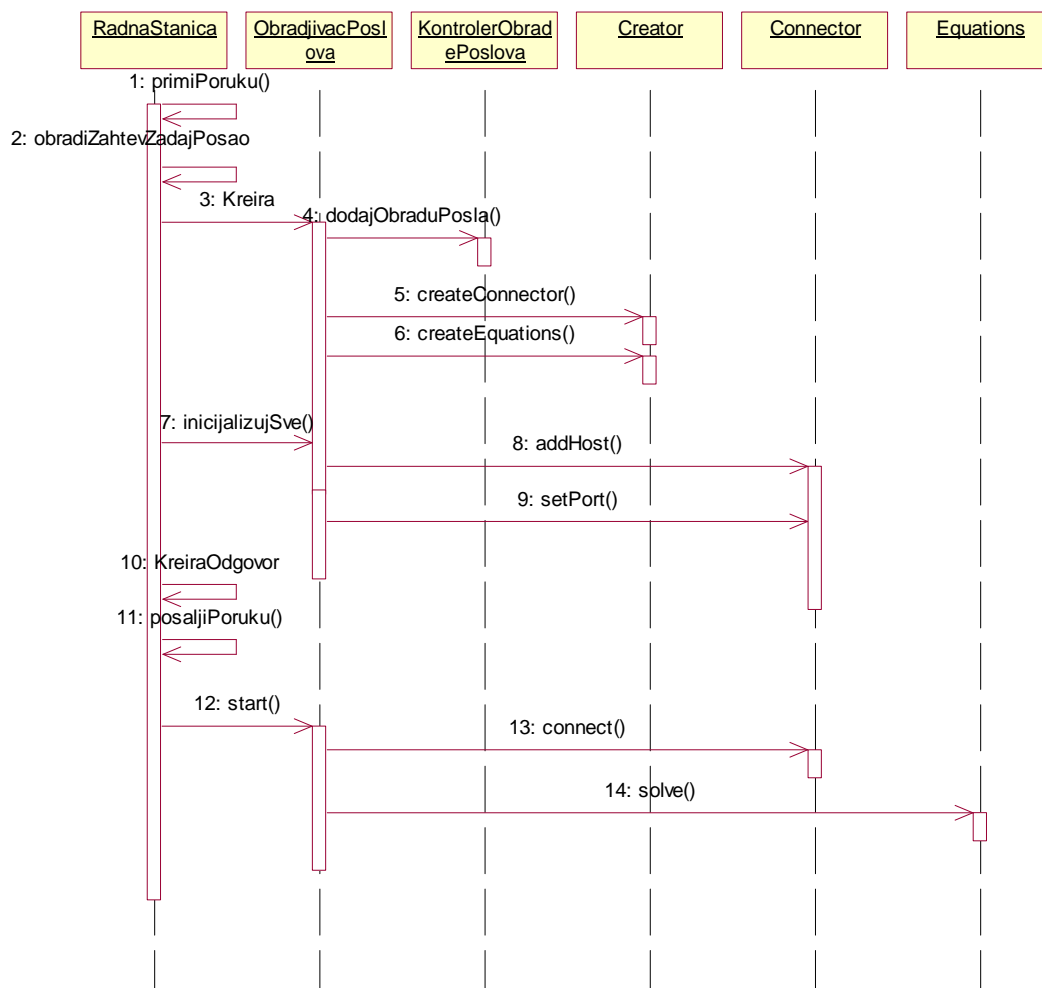
3.2.3 Повезивање радних станица са централним сервером. Провера присутности и стања самих радних станица са сервера.



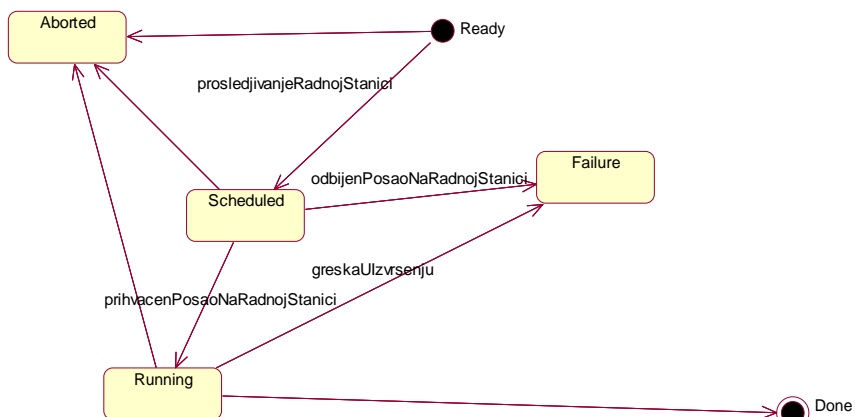
3.2.4 Контролер примљених послова на серверу је задужен за праћење рада свих послова и њихово поновно стартовање по потреби.



3.2.5 Прихват посла на радној станици. Позивање библиотеке за паралелно процесирање математичких израза.



3.3 Дијаграм стања посла после пријема на сервер



3.4 Опис протокола

Опис протокола за комуникацију са сервером је дат у табели доле.

Захтев	Опис	Параметри
-1	Грешка у пријему поруке	

5	Захтев за задавање посла од клијента	<ul style="list-style-type: none"> • Идентификатор клијента • Операција • Датотеке операнада • Име датотеке резултата
6	Одговор на задавање посла од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Порука о грешци
7	Захтев за стање посла од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Идентификатор клијента
8	Одговор на стање посла од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Стање посла на серверу • Порука о грешци
9	Захтев за преузимање резултата од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Идентификатор клијента
10	Одговор на преузимање резултата од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Датотека резултата • Стање посла • Порука о грешци
11	Захтев за заустављање посла од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Идентификатор клијента
12	Одговор на заустављање посла од клијента	<ul style="list-style-type: none"> • Идентификатор посла • Стање посла • Порука о грешци
20	Захтев за бројем слободних нити на радној станици	<ul style="list-style-type: none"> •
21	Одговор на захтев за бројем слободних нити радној станици	<ul style="list-style-type: none"> • Број расположивих нити • Порука о грешци
22	Захтев за задавање посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла • Операција • Датотеке операнада • Име датотеке резултата • Списак доступних радних станица
23	Одговор на захтев за задавање посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла • Број расположивих нити • Порука о грешци
24	Захтев за преузимање резултата посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла
25	Одговор на захтев за преузимање резултата посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла • Датотека резултата • Име датотеке резултата • Стање посла • Порука о грешци
26	Захтев за заустављање посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла
27	Одговор на захтев за заустављање посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла • Стање посла • Порука о грешци
28	Захтев за статусима посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла
29	Одговор на захтев за статусима посла на радној станици	<ul style="list-style-type: none"> • Идентификатор посла • Стање посла • Порука о грешци

4. КОРИСНИЧКО УПУТСТВО

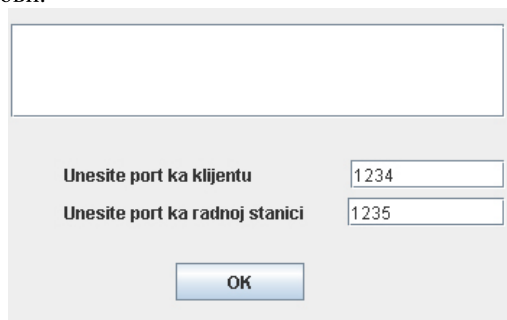
Упутство за коришћење је обрађено у два случаја, као упутство за корисника апликација и као упутство за коришћење програмских пакета од стране програмера.

4.1. Упутство за коришћење апликација система

4.1.1. Серверска апликација

Сервер се стартује кликом на бат фајл *start_server.bat* при чему се отвара прозор као на слици 7. Потребно је да корисник упише два порта, један за комуникацију сервера са клијентима и један за комуникацију сервера са радним станицама. Кликном на дугме *ОК*, сервер је у могућности да прима конекцију на та два порта, а уколико није могуће да се накачи на те портове, појавиће се порука о томе. Након успешног качења сервера на портове отвара се нови прозор као на слици 8. Кликном на дугме *Osvezi*, приказује се статус извршавања послова (приказује се податак о клијенту који је задао посао, јединствен идентификациони број посла, тренутни статус посла и радна станица која је примила тај посао на обраду). Уколико се јави нека грешка приликом извршавања посла, појавиће се порука о томе.

Током рада сервера се креира фолдер *DokumentaServer* у коме се налази фајл *Logfile.txt* у коме сервер бележи ток извршавања послова и фолдер *PosloviServer* у коме се чувају потребни операнди за извршавање послова (под именом: *idklijenta_imеfile*) и резултати успешног извршеног посла (под именом: *idklijenta_imеfile*). За исправан рад сервера потребно је да се не модификују наведени фајлови.



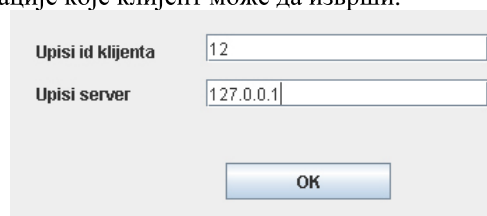
Слика 7



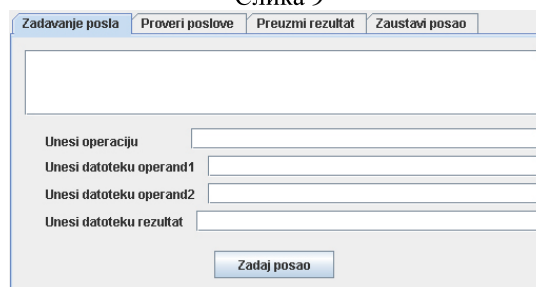
Слика 8

4.1.2. Клијентска апликација

Да би корисничка апликација радила потребно је у њеном конфигурационом фајлу уписати порт на коме сервер очекује конекцију клијента. Тај фајл се зове *configC.txt*, садржи само један податак (порт) и апликација очекује да се налази на *C* партицији. Корисничка апликација се стартује кликом на бат фајл *start_klijent.bat* при чему се отвара прозор као на слици 9. Потребно је унети *id klijenta* и *IP* адресу сервера и кликнути на тастер *ОК*. Уколико конекција прође успешно, отвара се нови прозор као на слици 10. У овом прозору постоје 4 таба, за 4 могуће операције које клијент може да изврши.



Слика 9



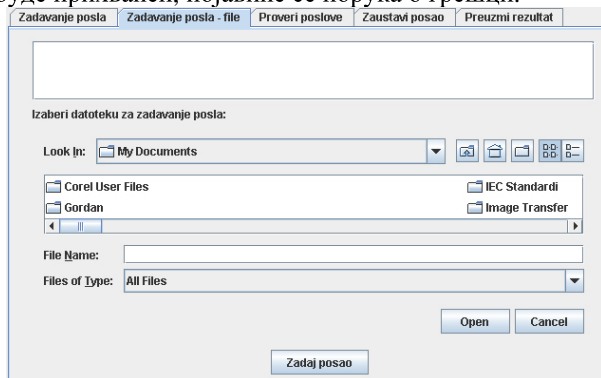
Слика 10

Креирање посла:

Да би клијент задао посао, потребно је да у табу *Zadavanje posla* који је приказан на слици 10 упише потребне податке: имена фајлова у којима се налазе операнди (при чему је операнд2 опциони), име операције коју је потребно извршити над њима и име фајла у који ће се уписати резултат успешно извршеног посла и да кликне на дугме *Zadaj posao*. Подразумева се да је екстензија тих фајлова *.txt*, па је није потребно уносити. Потребно је да се ови фајлови налазе у фолдеру *Dokumenta* на *C* партицији. Уколико сервер прихвати посао, клијент ће бити обавештен о јединственом идентификационом броју (*Id*) посла који је задао. Уколико посао не може да буде прихваћен, појавиће се порука о грешци.

Креирање посла путем датотеке:

Клијент има могућност задавања посла избором одговарајуће датотеке у табу *Zadavanje posla* - *file* који је приказан на слици 11. Потребно је да одговарајућа датотека има садржај као на слици 12. Уколико сервер прихвати посао, клијент ће бити обавештен о јединственом идентификационом броју (*Id*) посла који је задао. Уколико посао не може да буде прихваћен, појавиће се порука о грешци.



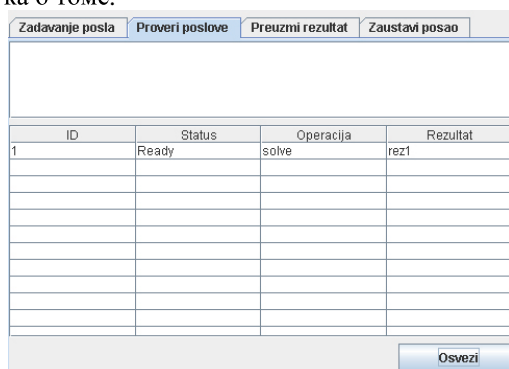
Слика 11

```
operacija=solve
op1=qwer
op2=b
rezultat=r1
```

Слика 12

Преглед статуса послова:

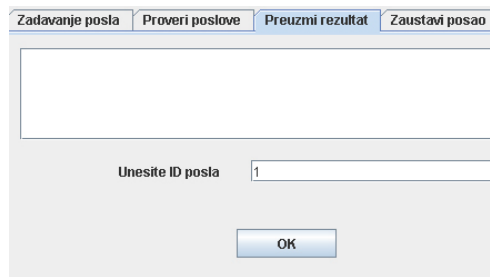
Клик на таб *Proveri poslove* је приказан на слици 13. Кликком на дугме *Osvezi*, приказује се статус извршавања послова које је задао тај клијент (приказује се јединствен идентификациони број посла, тренутни статус посла, операција која се извршава и име фајла у који се смешта резултат успешно извршеног посла). Уколико се јави нека грешка приликом прегледа статуса, појавиће се порука о томе.



Слика 13

Преузимање резултата

Клик на таб *Preuzmi rezultat* је приказан на слици 14. Уколико клијент жели да преузме резултат неког посла који се извршио до краја, онда је потребно да упише *Id posla* и кликне на дугме *OK*. Уколико се јави нека грешка приликом извршавања преузимања резултата, појавиће се порука о томе.

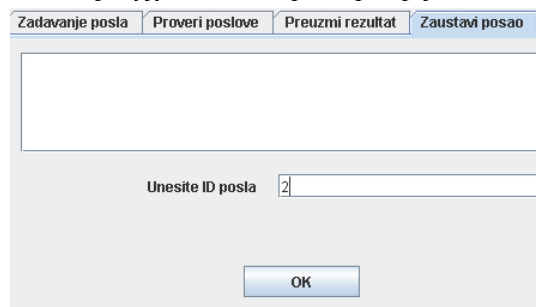


Слика 14

Прекидање посла

Клик на таб *Zaustavi posao* је приказан на слици 15. Уколико клијент жели да прекине неки посао који је задао а још увек се није извршио до краја, онда је потребно да упише *Id posla* и кликне на дугме *OK*. Уколико се јави нека грешка приликом прекидања посла, појавиће се порука о томе.

За исправан рад клијентске апликације потребно је да се не модификују наведени фолдер и фајлови.



Слика 15

4.1.3 Апликација радне станице

Да би апликација радне станице радила потребно је у њеном конфигурационом фајлу уписати порт на коме сервер очекује конекцију радне станице, *IP* адресу сервера и број послова који радна станица може да обавља у паралели. Тај фајл се зове *configRS.txt*, садржи само један ред у коме су уписани *port;IP_adresa_servera;broj_niti* и апликација очекује да се налази на *C* партицији. Апликација радне станице може бити покренута из командне линије или покретањем графичког окружења и у зависности од тога се стартује кликом или на бат фајл *start_rs.bat* или на бат фајл *start_rsG.bat* при чему се отвара прозор као на слици 16. Ако се покреће из командне линије, радна станица чита потребне податке из конфигурационог фајла. Ако се покреће из графичког окружења, радна станица чита податке из конфигурационог фајла, али се *IP* адреса сервера и порт по коме ће се вршити комуникација са сервером могу накнадно променити из прозора са слике 16. Да би, у том случају, радна станица остварила комуникацију са сервером, потребно је кликнути на дугме *OK* Уколико се јави нека грешка приликом остваривања комуникације са сервером, појавиће се порука о томе.

Током рада радне станице се креира фолдер *DokumentaRS* у коме се чувају резултати успешно извршеног посла (под именом: *idkljenta_imfile*). За

исправан рад радне станице потребно је да се не модификују наведени фајлови.



Слика 16

4.2 Упутство за коришћење програмских пакета система

Ово упутство се састоји из кратких описа сваке од класа апликација.

4.2.1 Серверска апликација

- + Prozori
 - Prozor1.java
 - Prozor2.java
- + paketServer
 - KonekcijaKlijent.java
 - KonekcijaRStanica.java
 - KontrolerPoslova.java
 - KontrolerRadnihStanica.java
 - Logger.java
 - PosaoInfo.java
 - ServerRadneStanice.java
 - TCPEchoServer.java

Prozori.Prozor1

Графички интерфејс серверске апликације.

Prozori.Prozor2

Графички интерфејс серверске апликације.

paketServer.KonekcijaKlijent

Класа за конекцију сервера са клијентском апликацијом.

paketServer.KonekcijaRStanica

Класа за конекцију сервера радном станицом.

paketServer.KontrolerPoslova

Класа за чување информација о свим пословима који су на серверу и прослеђивање послова радним станицама по потреби.

paketServer.KontrolerRadnihStanica

Чува информације о свим радним станицама у систему.

paketServer.Logger

Класа која обрађује рад са фајловима.

paketServer.ServerRadneStanice

Прима конекције од радних станица.

paketServer.TCPEchoServer

Прима конекције од клијената.

4.2.2 Клијентска апликација

- + Prozori
 - Prozor1.java
 - Prozor2.java
- + Klase
 - Fun.java
 - TCPEchoClient.java

Prozori.Prozor1

Графички интерфејс клијентске апликације.

Prozori.Prozor2

Графички интерфејс клијентске апликације.

Klase.Fun

Класа која обрађује рад са фајловима.

Klase.TCPEchoClient

Класа за конекцију клијента на сервер.

4.2.3 Апликација радне станице

- + GUIRStanica
 - Pocetni.java
 - Pregled.java
- + RadnaStanica
 - Glavna.java
 - KontrolerObradePoslova.java
 - LoggerR.java
 - ObradljivacPoslova.java
 - RadnaStanica.java

GUIRStanica.Pocetni

Графички интерфејс апликације радне станице.

GUIRStanica.Pregled

Графички интерфејс апликације радне станице.

RadnaStanica.Glavna

Класа за покретање апликације без графичког окружења.

RadnaStanica.KontrolerObradePoslova

Контрола обраде свих послова који се извршавају на радној станици.

RadnaStanica.LoggerR

Класа која обрађује рад са фајловима.

RadnaStanica.ObradljivacPoslova

Позивање библиотеке за паралелно израчунавање математичких израза.

RadnaStanica.RadnaStanica

Класа за конекцију радне станице ка серверу.