

УНИВЕРЗИТЕТ У БЕОГРАДУ  
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ



## **СИСТЕМ БЕРЗАНСКОГ ПОСЛОВАЊА**

Пројекат из предмета Конкурентно и дистрибуирано  
програмирање

Лука В. Стојановић 2018/0053

Београд, септембар 2021.

## **Садржај**

<b>1. ТЕКСТ ЗАДАТКА</b> .....	2
<b>2. UML ДИЈАГРАМИ</b> : .....	3
<i>Класни дијаграм централни сервер:</i> .....	3
<i>Класни дијаграм сервер радник:</i> .....	4
<i>Класни дијаграм клијентског програма:</i> .....	4
<i>Класни дијаграм дељеног пакета:</i> .....	4
<i>Дијаграм интеракција значајнијих метода централног сервера:</i> .....	5
<i>Дијаграм интеракција значајнијих метода сервера радника:</i> .....	6
<i>Дијаграм интеракција значајнијих метода сервера радника:</i> .....	6
<b>3. УПУТСТВО ЗА КОРИШЋЕЊЕ ПРОГРАМА:</b> .....	7
<i>Кориснички програм</i> .....	7
<i>Централни сервер:</i> .....	7
<i>Сервер радник:</i> .....	7
<b>4. ПРИМЕРИ РАДА ПРОГРАМА:</b> .....	8
<i>Примери рада сва три програма у нормалним ситуацијама:</i> .....	8
<i>Примери рада програма у ванредним ситуацијама:</i> .....	8
<i>Радни сервер:</i> .....	9

## 1.ТЕКСТ ЗАДАТКА

Пројектовати дистрибуирани рачунарски систем који треба да омогући берзанске трансакције. 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 пријем, праћење цене и власништва над хартијама од вредности, као и за објављивање тренутног стања.
2. Подсервери на којима се обављају конкретне берзанске трансакције, односно куповина, продаја и израчунавање нове цене хартија од вредности.
3. Клијентски програм преко кога клијент прати тренутне цене хартија од вредности, издаје захтеве за куповином и продајом појединих.

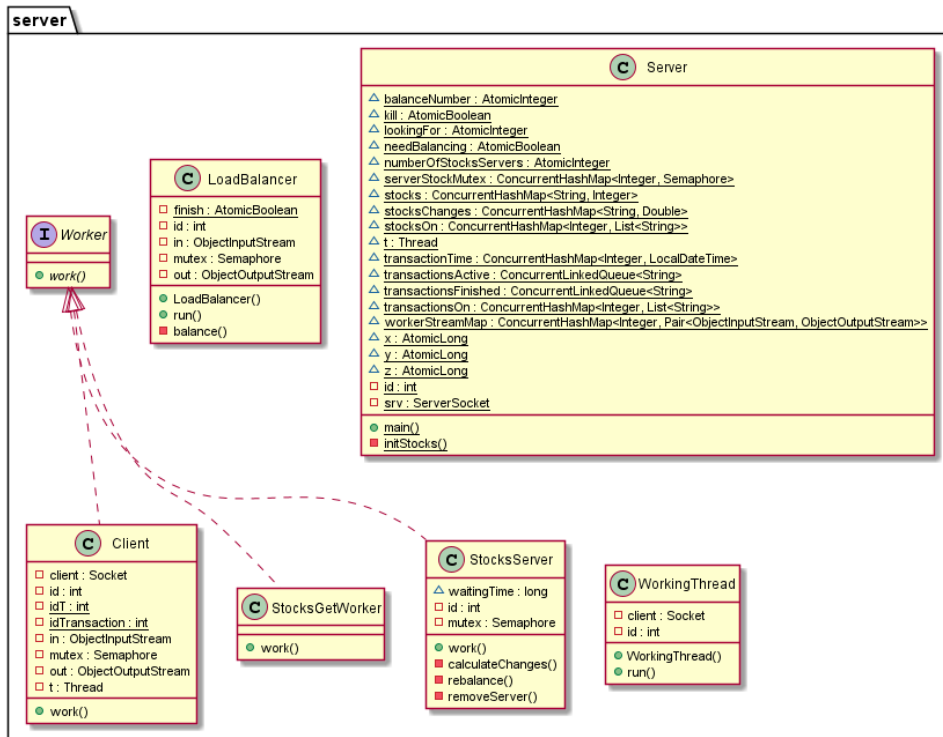
Централни сервер својим претплаћеним корисницима нуди услуге праћења цене хартија од вредности. Ова услуга је таква да сервер периодично шаље свим претплаћеним клијентима тренутну цену свих хартија од вредности. Сервер шаље податке један по један, тако да назначава име хартије, тренутну цену и промену у процентима у односу на претходну вредност (не шаље се податак о свакој појединачно издатој акцији/обвезници него о хартијама које се везују за неку фирму). Ово је поступак који се периодично понавља, а та периода износи  $x$  секунди. Централни сервер податке о цени добија од подсервера који воде рачуна о појединим хартијама од вредности. Централни сервер податке периодично прикупља са подсервера, при чему је тај период износи у секунди. За сваку врсту хартија од вредности постоји тачно један подсервер који води рачуна о њој. Уколико подсервер није доступан, покушава поново за  $y/2$  секунди. Ако три пута за редом податак не успе да се прибави, подсервер се проглашава недоступним, треба урадити балансирање свих клијената на све доступне подсервере (load balancing) и све нове трансакције се пребацују на неки други подсервер коме се додељује да води евиденцију о тој хартији од вредности, док се податак о хартији задржава на последњој читаној вредности. Клијент има могућност да се претплати код сервера тако што пошаље корисничко име и шифру и периодично добија податке о ценама хартија од вредности. Осим прегледа свих цена корисник може да обавља купопродајне трансакције над хартијама од вредности. Приликом купопродајних трансакција корисник шаље захтев централном серверу у ком стоји да жели да купи/прода одређену количину хартија од вредности, као и цену по којој би трансакцију да обави. Када централни сервер прими тај захтев, прослеђује га подсерверу који је задужен за те хартије. Тај подсервер од тог тренутка постаје надлежан за ту трансакцију. Уколико у тренутку приспећа захтева за куповину хартије постоји довољна количина хартија по цени која није већа од оне коју је корисник задао, обавља се трансакција. Ако не постоји онда се чека. Уколико постоји већи број које могу бити купљене, прво се гледају оне које имају најнижу цену, а ако има више оних са најнижом ценом, онда се међу њима прво узима онај који је пре приспео. Уколико у тренутку приспећа захтева за продајом хартије постоји довољна количина хартија по цени која није мања од оне коју је корисник задао, обавља се трансакција. Ако не постоји онда се чека. Уколико постоји већи број које могу бити продате прво се гледају оне које имају највишу цену, а ако има више оних са највишом ценом, онда се међу њима прво узима онај који је пре приспео. Формирање цене поједине хартије коју је потребно проследити централном серверу када је он буде тражио се обавља на следећи начин: Ако је у интервалу између два читавања било трансакција, онда је цена специфицирана последњом обављеном трансакцијом. Ако није било трансакција него је било само понуда за продајом, узима се најнижа активна понуђена цена. Уколико је било само понуда за куповином, узима се највиша активна понуђена цена. Уколико нема активних понуда, онда се узима цена последње обављене трансакције. Уколико се корисник предомисли, може да одустане од затражене трансакције, али понуда мора да буде активна барем  $z$  секунди ( $z > 2 * x$ ,  $x > y$ ). Када се обави трансакција или отказивање корисник добија повратну информацију о томе. Уколико листа подсервера која се води на централном серверу остане празна, централни сервер јавља грешку и обуставља све трансакције док се не појави барем један нови подсервер. Да би се подсервер пријавио у листу активних подсервера на централном серверу мора да пошаље централном серверу захтев за комуникацију. Том приликом централни сервер датом подсерверу додељује за које је хартије задужен. Да би балансирао оптерећење централни сервер може са једног подсервера да на други пребаци операције над неком хартијом од вредности, али тако да у било ком тренутку максимално један води рачуна о једној врсти хартија. Ни корисници ни подсервери не морају да буду у сталној вези са централним сервером. Веза може бити раскинута гашењем програма или затварањем комуникационог канала. Када се кориснички програм повеже може да тражи од сервера статусе свих трансакција. Треба обезбедити да сваки сервер може да у паралели да прима захтеве за већим бројем хартија од вредности. Проблем решити користећи мрежну комуникацију у програмском језику Јава. За сваки од ова три типа рачунара треба да постоји одговарајући графички кориснички интерфејс (GUI треба да буде развијен користећи Java SWING компоненте или JavaFX). Графички кориснички интерфејс мора бити интуитиван. IP адресе и портови потребни за функционисање програма (као и остали подаци који се могу динамички мењати) морају се уносити преко графичког интерфејса (не смеју бити унапред дефинисани у изворном коду). За

сваки од могућих случајева коришћења система поруке и обавештења морају бити прецизно приказани. Потребно је да се кроз графички кориснички интерфејс прикажу детаљни логови, како за централни сервер, тако и за подсервере. Сервери треба да имају могућност покретања и без корисничког интерфејса.

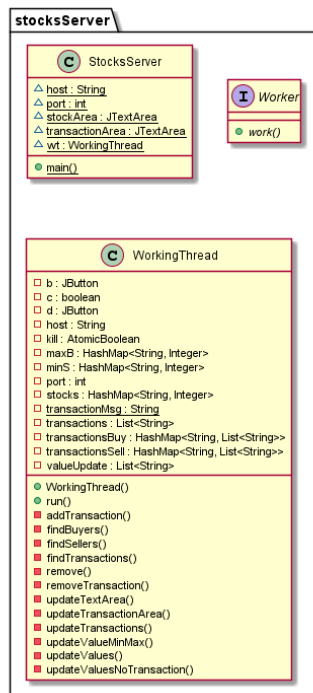
## 2. UML ДИЈАГРАМИ:

У наставку су дати класни дијаграми свих пакета који сачињавају решење, као и значајнији дијаграми секвенци.

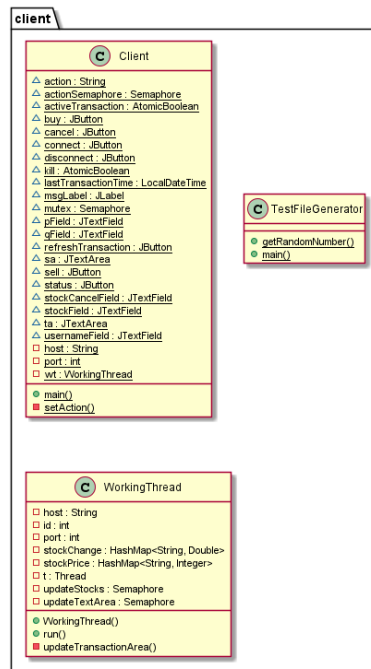
**Класни дијаграм централни сервер:**



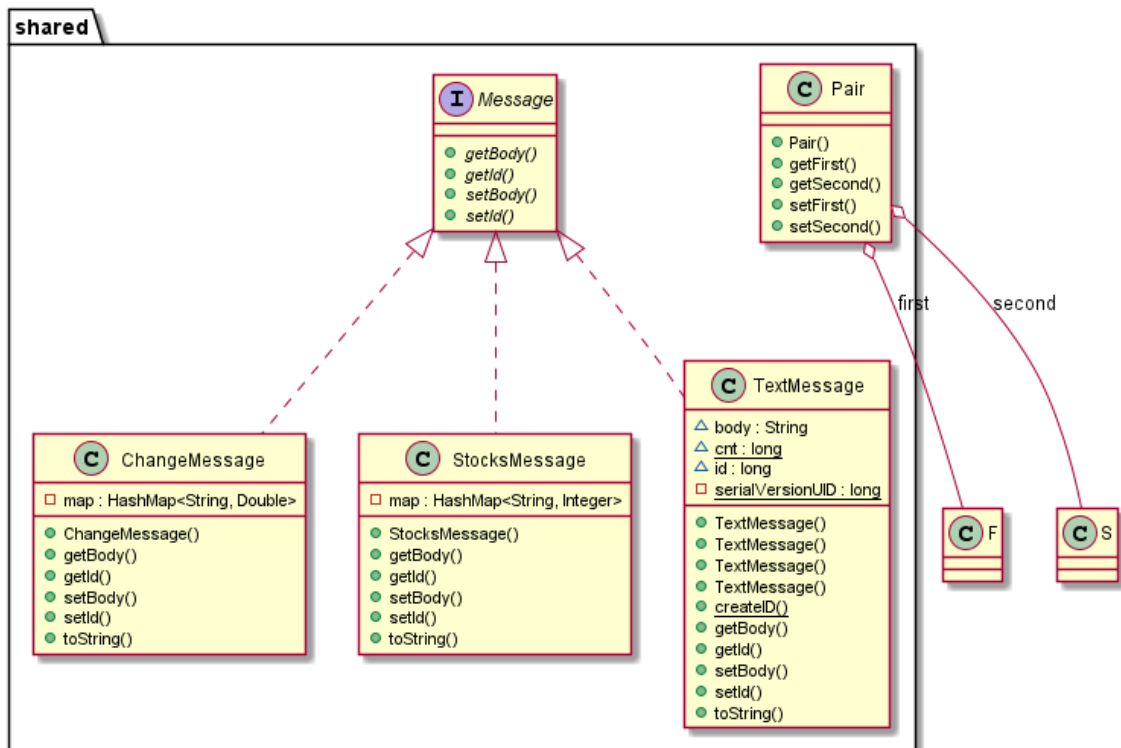
## Класни дијаграм сервер радник:



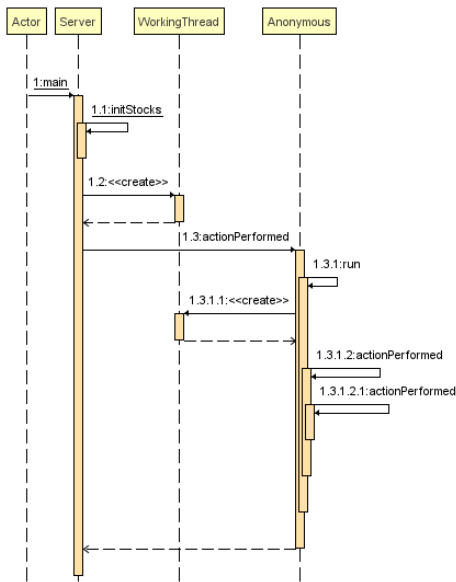
## Класни дијаграм клијентског програма:



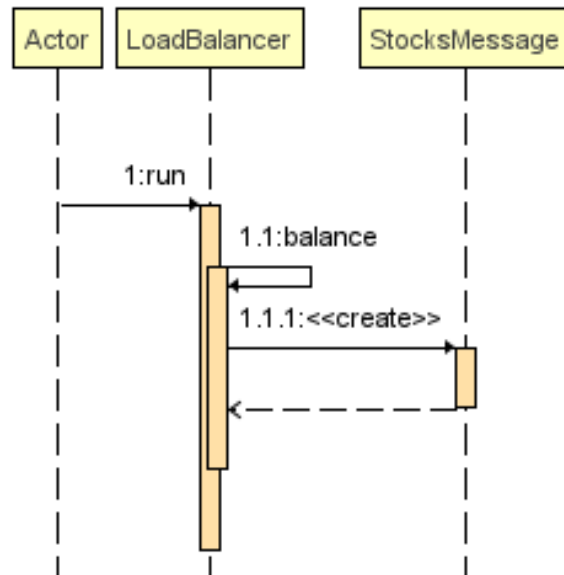
## Класни дијаграм дељеног пакета:



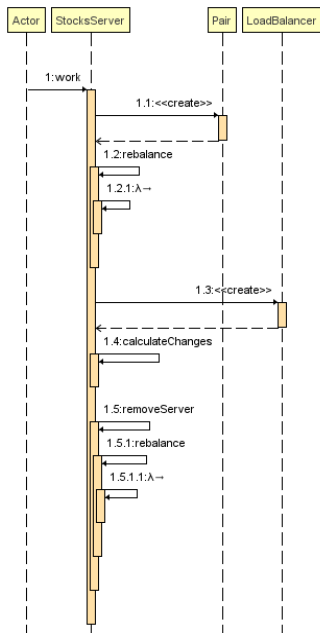
## Дијаграм интеракција значајнијих метода централног сервера:



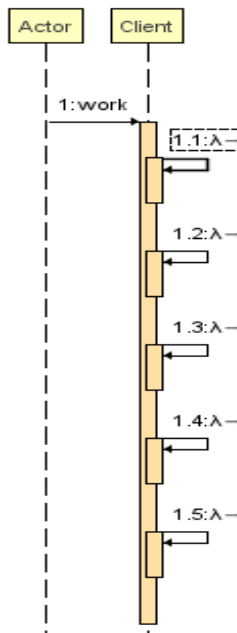
Дијаграм интеракција: 1 Server-main



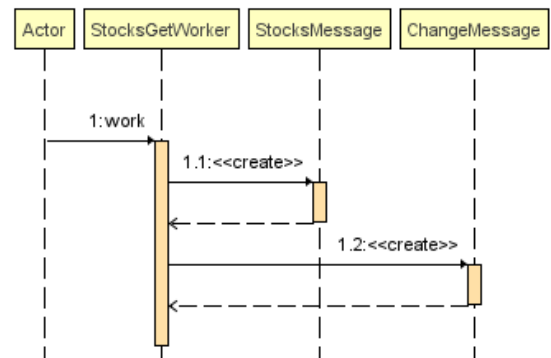
Дијаграм интеракција: 2 Server-LoadBalancer



Дијаграм интеракција: 3 StocksServer-work

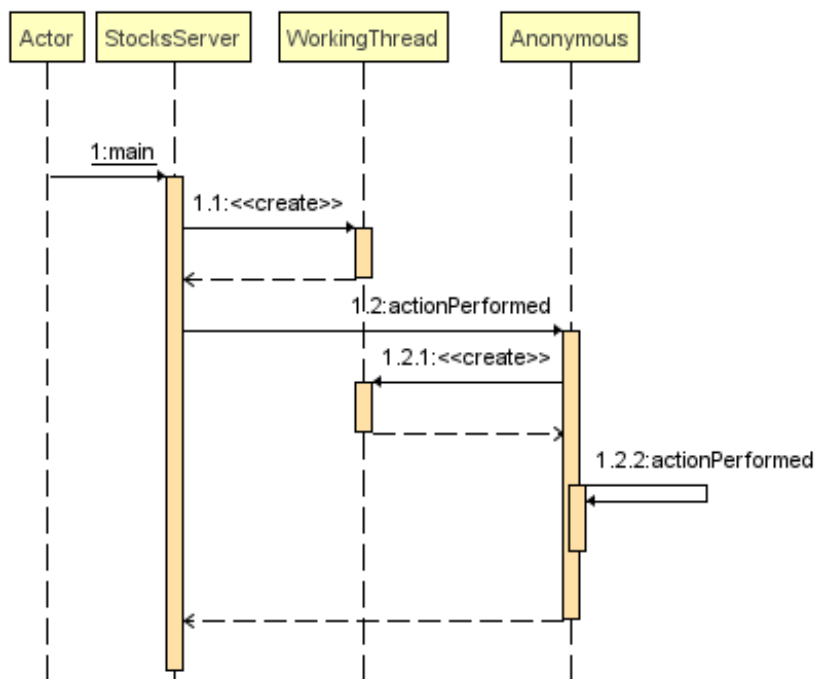


Дијаграм интеракција: 4 Client-work



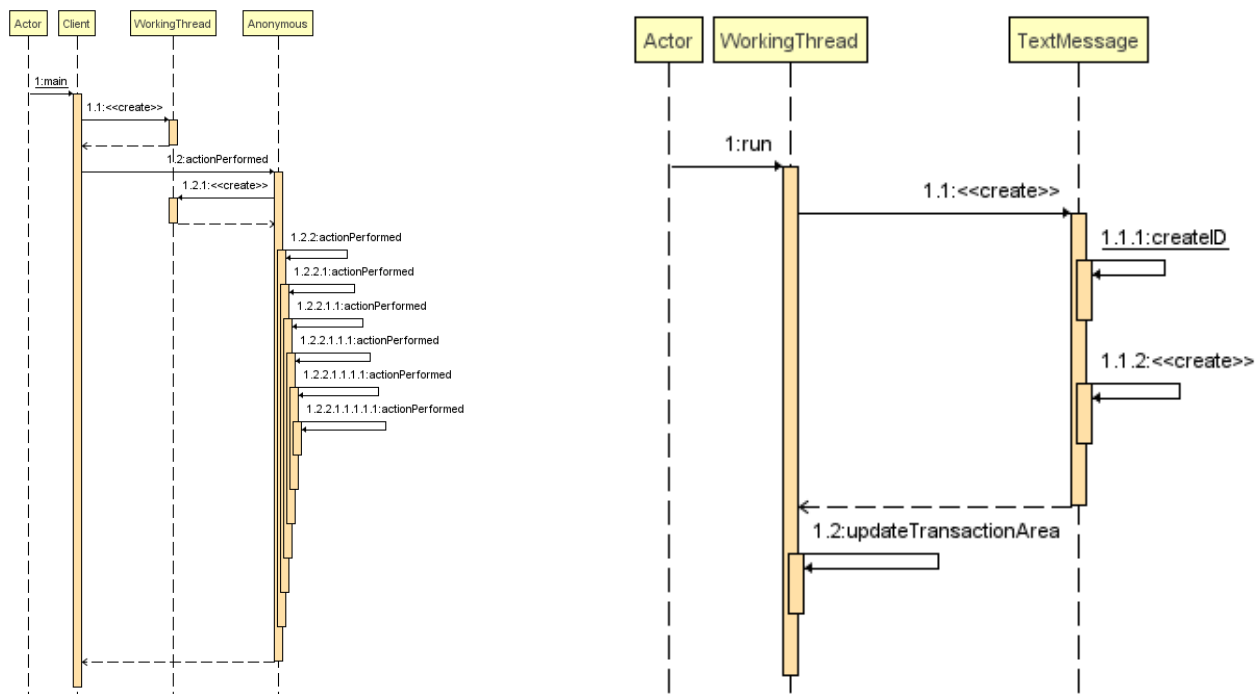
Дијаграм интеракција: 5 StocksGetWorker-work

Дијаграм интеракција значајнијих метода сервера радника:



Дијаграм интеракција: 6 StocksServer-main

Дијаграм интеракција значајнијих метода сервера радника:



Дијаграм интеракција: 7 Client-main

Дијаграм интеракција: 8 Client-WorkingThread-run

### 3. УПУТСТВО ЗА КОРИШЋЕЊЕ ПРОГРАМА:

Сваки програм има могућност покретања у конзолном и графичком режиму. Клијентски програм и сервер радник мају графички интерфејс прилагођен особама са минималним искуством у коришћењу истих или сличних програма. Главни сервер и конзолне верзије осталих програма су намењене особама са напредним искуством. Код конзолних опција корисник је задужен за исправност унесених параметара, у случају неисправних параметара програм се гаси.

#### Кориснички програм

- Графички интерфејс:

Састоји се од поља и лабела јасно назначених намена. Од корисника се захтева да унесе адресу и порт централног сервера, као и своје креденцијале како би се успоставила конекција. Када је конекција успешно успостављена корисник може да уноси назив, количину и цену хартија од вредности коју жели да купи/прода. Избором дугмета за куповину односно продају, шаље се захтев централном серверу са одабраном акцијом. Постоји дугме за освежавање листе трансакција помоћу којег корисник може видети листу свих трансакција. Могуће је отказати или проверити статус трансакција уношењем јединственог идентификатора трансакције и одабиром одговарајућег дугмета. Корисник је у сваком моменту у могућности да прекине конекцију са централним сервером одабиром одговарајућег дугмета.

- Конзолни интерфејс:

Конзолни интерфејс се покреће прослеђивањем параметара при покретању програма у следећем формату:

"port host korisnickoIme с putanjaFajla"

Као 4-ти аргумент позива се наводи слово "с"(латинично ц), а као 5-ти аргумент опционо путања до фајла са командама које ће програм аутоматски извршавати, ради аутоматизованог тестирања. По покретању програма корисник уноси команде преко конзоле које програм извршава. Следи списак могућих команди:

"buy/sell;nazivHartije;Kolicina;Cena" - количина и цена су обавезно целобројне, команда служи за куповину односно продају одређене количине хартија од вредности

"status/cancel;idHartije" - провера статуса односно поништавање трансакције чији се јединствени идентификатор проследи као други аргумент команде

"stocks" - захтев за тренутном ценом свих хартија од вредности

"refresh" - захтев за свим до сада оствареним трансакцијама

"stop" - прекид рада програма

#### Централни сервер:

- Графички интерфејс:

Састоји се од поља за унос порта на којем ће централни сервер прихватати захтеве, као и поља за унос параметара неопходних за покретање програма. Постоји могућност прекида рада сервера одабиром одговарајућег дугмета.

- Конзолни интерфејс:

Конзолни интерфејс се покреће прослеђивањем параметара при покретању програма у следећем формату:

"port x y z"

Сви параметри су обавезни, целобројног типа и треба да важи следећа релација:

$z > 2 * x, x > y$

По покретању програма сервер функционише потпуно независно, и у конзоли приказује поруке од значаја.

#### Сервер радник:

- Графички интерфејс:

Састоји се од поља и лабела јасно назначених намена. Од корисника се захтева да унесе адресу и порт централног сервера, као и своје креденцијале како би се успоставила конекција. По успостављању конекције могуће је видети све хартије од вредности за које је тај радник задужен, као и све трансакције које се тренутно обављају на њему. Могуће је прекинути конекцију са централним сервером одабиром одговарајућег дугмета.

- Конзолни интерфејс:

Конзолни интерфејс се покреће прослеђивањем параметара при покретању програма у следећем формату:

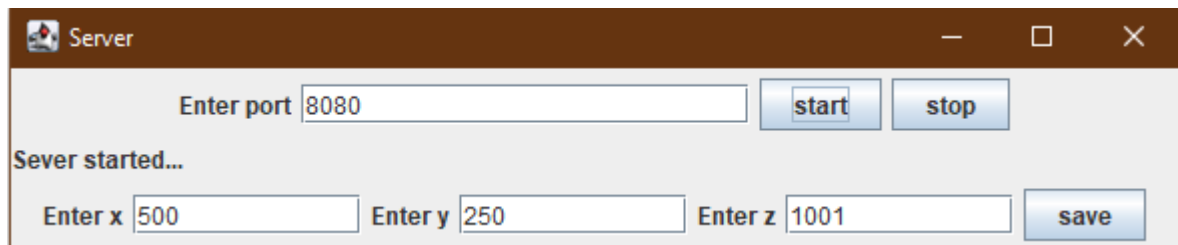
"port host с"

Као 3-ћи аргумент позива се наводи слово "с"(латинично ц). По покретању, програм се извршава потпуно независно, и приказује значајне поруке у конзоли. У конзоли је могуће видети хартије од вредности и трансакције за које је тај сервер задужен.

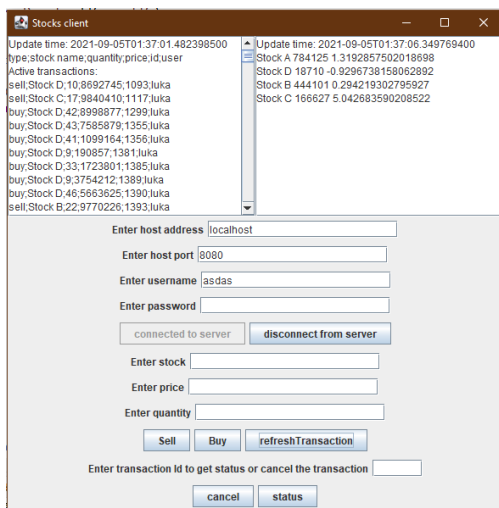


## 4. ПРИМЕРИ РАДА ПРОГРАМА:

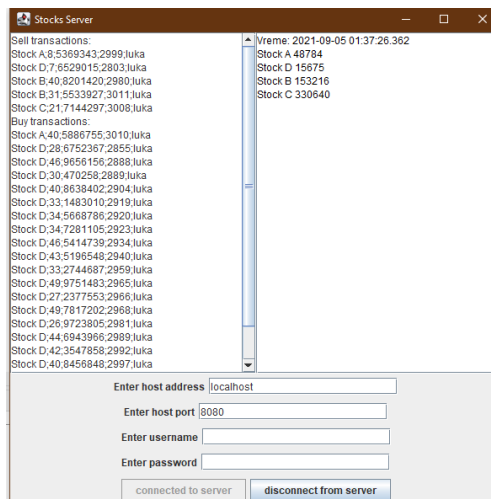
Примери рада сва три програма у нормалним ситуацијама:



Пример рада програма 1 Централни Сервер



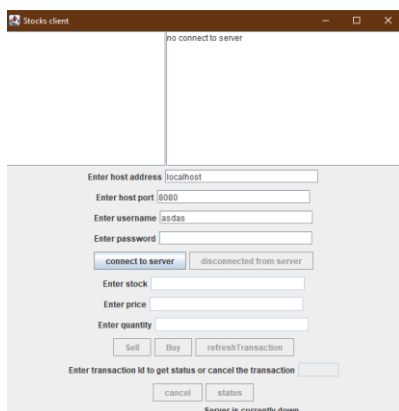
Пример рада програма 2 Клијент



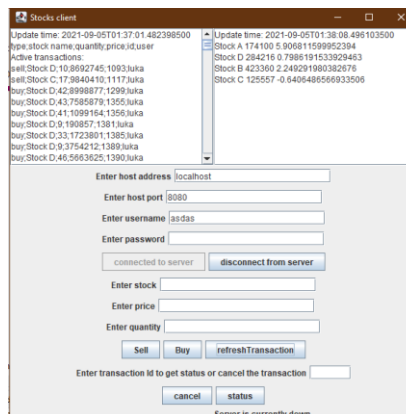
Пример рада програма 3 Радни Сервер

Примери рада програма у ванредним ситуацијама:

Клијент:

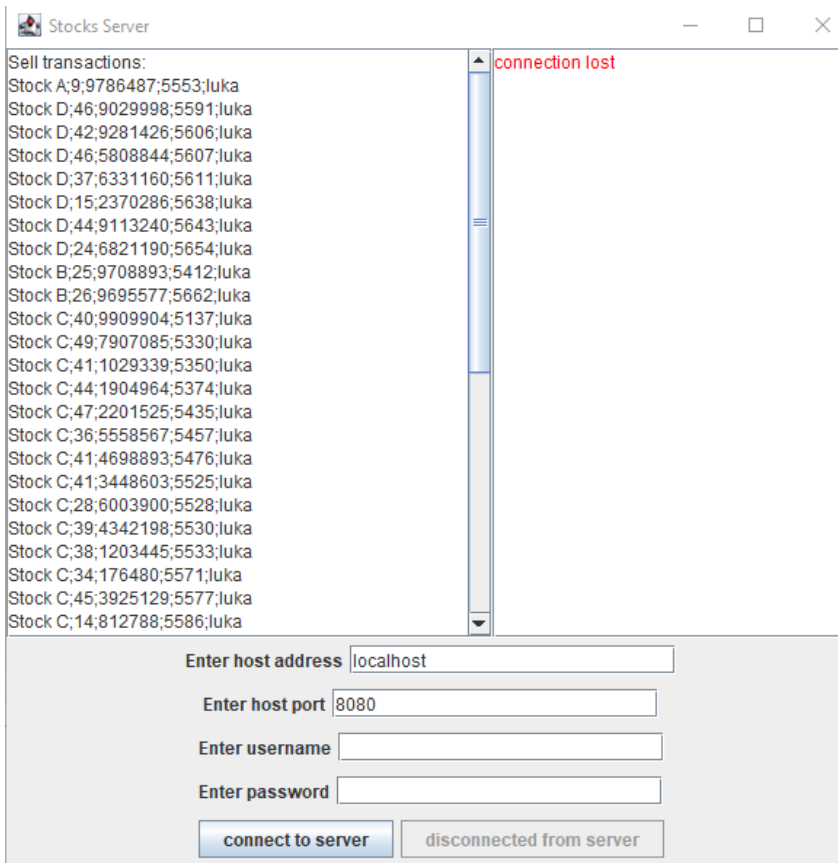


Пример рада програма 5 Непостојећа конекција



Пример рада програма 4 Централни сервер јавља грешку

## Радни сервер:



The screenshot shows a window titled "Stocks Server" with a list of sell transactions and a connection status indicator. The list of transactions is as follows:

Transaction
Sell transactions:
Stock A;9;9786487;5553;luka
Stock D;46;9029998;5591;luka
Stock D;42;9281426;5606;luka
Stock D;46;5808844;5607;luka
Stock D;37;6331160;5611;luka
Stock D;15;2370286;5638;luka
Stock D;44;9113240;5643;luka
Stock D;24;6821190;5654;luka
Stock B;25;9708893;5412;luka
Stock B;26;9695577;5662;luka
Stock C;40;9909904;5137;luka
Stock C;49;7907085;5330;luka
Stock C;41;1029339;5350;luka
Stock C;44;1904964;5374;luka
Stock C;47;2201525;5435;luka
Stock C;36;5558567;5457;luka
Stock C;41;4698893;5476;luka
Stock C;41;3448603;5525;luka
Stock C;28;6003900;5528;luka
Stock C;39;4342198;5530;luka
Stock C;38;1203445;5533;luka
Stock C;34;176480;5571;luka
Stock C;45;3925129;5577;luka
Stock C;14;812788;5586;luka

The connection status is indicated by a red text "connection lost" in the top right corner of the window. Below the list of transactions, there is a form with the following fields:

- Enter host address: localhost
- Enter host port: 8080
- Enter username:
- Enter password:

At the bottom of the form, there are two buttons: "connect to server" and "disconnected from server".

Пример рада програма 6 Неуспешна конекција