

רשתות ומחשוב – תרגיל רביעי

הוראות להגשת המטלה:

1. יש לתת שם שמכיל ת.ז. של הסטודנט/ים, לקובץ מכוון.
2. שימו לב, חובה שהקוד ירוץ בכל מחשב ולכן כל החלטה מקומית (path) חייבת להיות כללית.
3. כתיבת קוד נכונה כולל שמות משתנים, אובייקטים ופונקציות קטנות שצריך, אין מספרי קסם.
4. יש לכוון את כל הקבצים באחד מהפורמטים הבאים, ZIP, RAR, 7-ZIP, בלבד.

a. יש להגיש Java Doc

b. יש להגיש קובץ pcap

c. יש להגיש מסמך class diagram

d. חובה להראות שהשתמשם ב-JUNIT

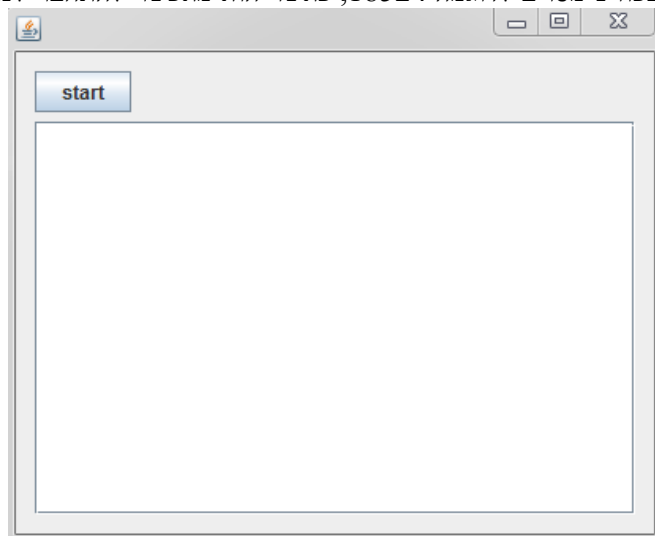
e. יש להגיש הן קבצי java וגם jar לשרת ו-jar ללקוח

תיאור התרגיל:

בתרגיל זה עליכם לבנות מערכת מסרים מיידים פרימיטיבית (בדומה ל-messenger) מבוססת על תקשורת TCP/IP תוך שימוש ב-JAVA. ספריות רלוונטיות ניתן למצוא באינטרנט. חובה לציין מקור הספריות אשר בשימוש.

המערכת תהיה בנויה משרת ולקוחות כאשר:

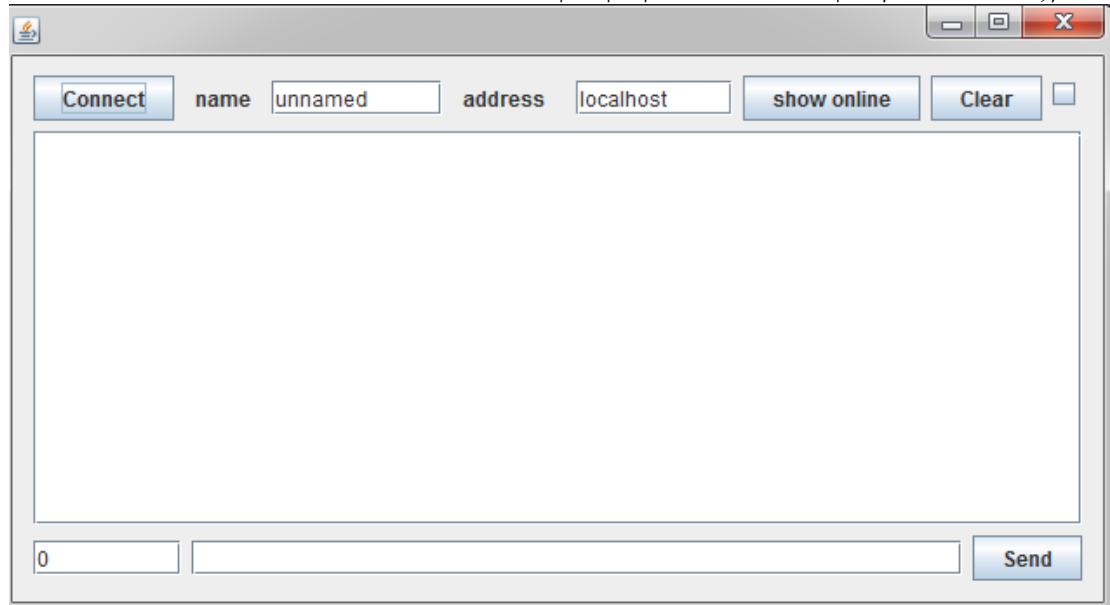
השרת (server) מאתחל את עצמו ו"מקשיב". השרת רץ על גבי שרת IP מסוים ומקשיב ללקוחות בפורט מסוים לדוגמה 18524, כאשר הוא מאפשר להתחבר למספר לקוחות בו זמנית (לפחות 5).



הלקוח (client) יכול לבצע את הפעולות הבאות:

- (1) להתחבר לשרת
- (2) להתנתק מהשרת
- (3) לשלוח הודעה ללקוח אחר
- (4) לשלוח הודעה לכל הלקוחות המחוברים לשרת כעת
- (5) לקבל את שמות הלקוחות המחוברים לשרת

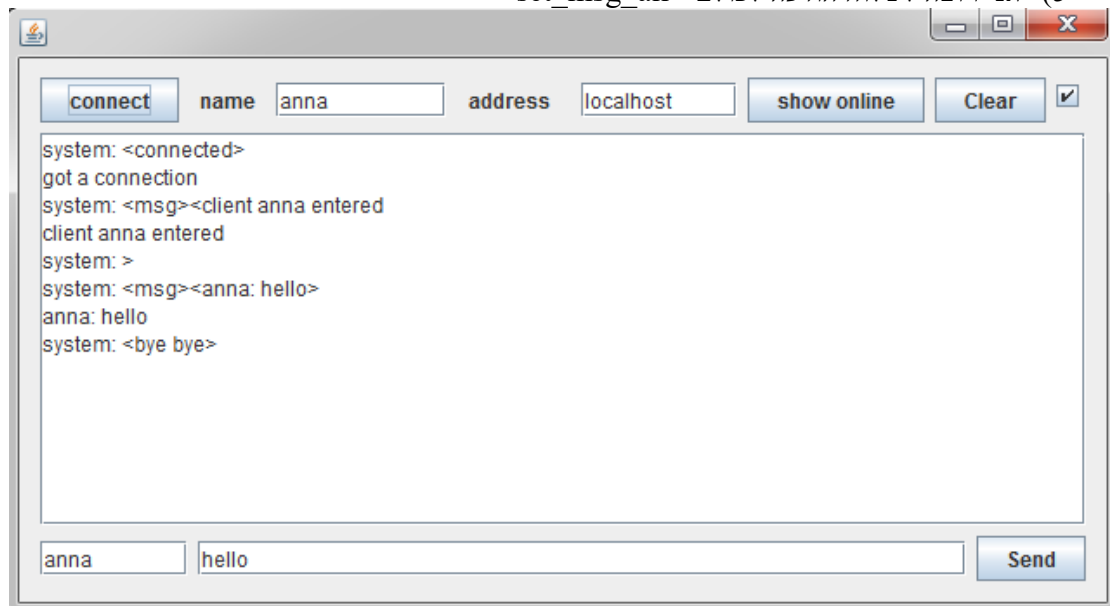
כמו כן, במידה והצטרף לקוח חדש או התנתק לקוח קיים יש לשלוח הודעה מתאימה לכל המשתתפים.



מומלץ להגדיר פרוטוקול (שפה משותפת) בין השרת ללקוח. לדוגמה:

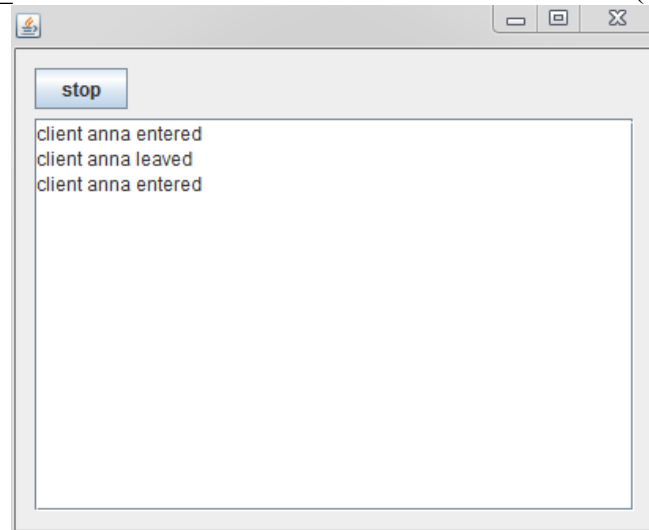
הודעות הלקוח:

- (1) אני רוצה להתחבר - `<connect><name>`
- (2) אני רוצה לקבל את רשימת המחוברים `<get_users>`
- (3) אני רוצה להתנתק `<disconnect>`
- (4) אני רוצה לשלוח הודעה ללקוח `<set_msg><Name>`
- (5) אני רוצה לשלוח הודעה לכולם `<set_msg_all>`



הודעות השרת

- (1) התחברת <connected>
- (2) התנתקת <disconnected>
- (3) ההודעות עבורך <msg_lst><num_of_msgs><"...">....<"..."><end>
- (4) רשימת המחוברים <users_lst><num_of_users><"...">...<"..."><end>



הערות כלליות:

- ניתן להניח שכמות ההודעות של כל לקוח לא תעלה על 100 הודעות.
- קודם יש לנסות על מחשב אחד גם את הלקוח וגם את שרת. כתובת מקומית של כל מחשב היא 127.0.0.1 או localhost.

בהצלחה!