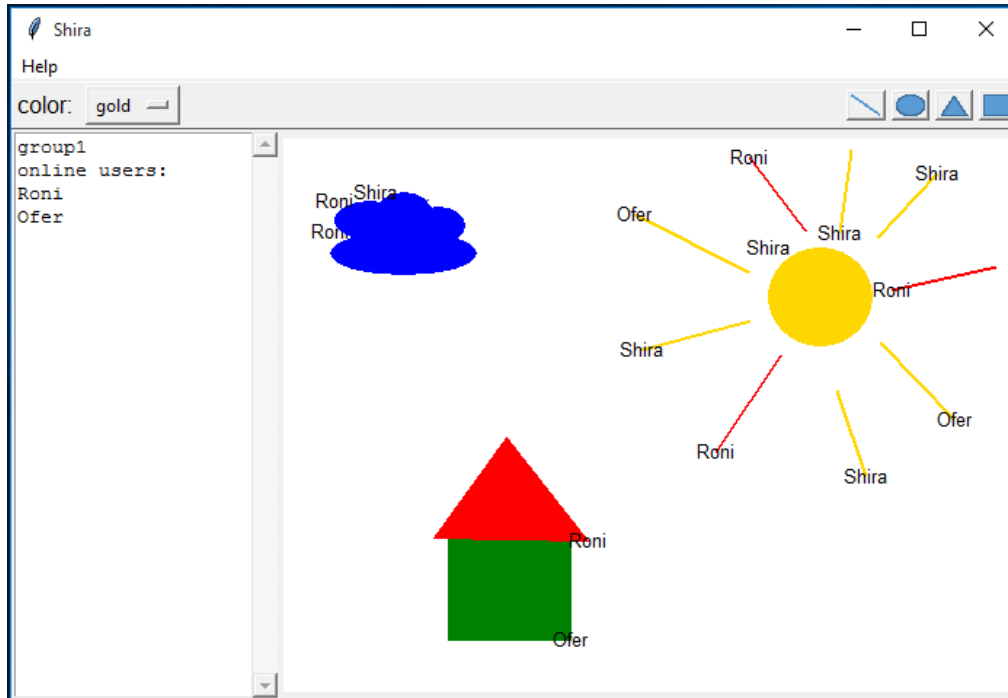


מבוא למדעי המחשב 67101  
תרגיל 12 – תקשורת וממשק למשתמש  
להגשה בתאריך 20/01/2016 בשעה 20:15

**תרגיל זה הוא להגשה בזוגות.**

בתרגיל זה נממש אפליקציה של לוח ציור משותף בין מספר משתמשים. האפליקציה מאפשרת ליצור קבוצות שונות של משתמשים, כאשר לכל קבוצה לוח ציור משותף. כל משתמש מודיע בעת ההתחברות מה שם המשתמש שלו ולאילו קבוצה הוא מעוניין להתחבר. לאחר ההתחברות נפתח חלון שבו מוצג לוח הציור עם הצורות שקיימות על הלוח של הקבוצה אליה הצטרף, ורשימת המשתמשים האחרים המחוברים באותה קבוצה. בנוסף, הלוח מאפשר למשתמש להוסיף צורה ללוח ע"י בחירה של הצורה, צבעה, ומיקומה על הלוח. כל פעולה שקורית אצל אחד המשתמשים המחוברים באותה קבוצה, כמו התחברות לקבוצה, הוספת צורה ללוח או עזיבת הקבוצה, צריכה להתעדכן אצל כל המשתמשים האחרים באותה קבוצה. דוגמא לממשק הגרפי:



שם המשתמש ושם הקבוצה חייבים להיות מחרוזות באורך עד 20 תווים, וצריכים להכיל רק מספרים ואותיות לועזיות. לצורך המימוש נשתמש בארכיטקטורת שרת לקוח.

## הלקוח:

את הלקוח עליכם לממש בקובץ client.py .

## **הלקוח מקבל את הפרמטרים הבאים לפי הסדר:**

1. כתובת השרת.  
עבור השרת שרץ באוניברסיטה הכתובת היא e-intro.cs.huji.ac.il (אם אתם עובדים על מחשבים של האוניברסיטה הכתובת היא e-intro).  
אם ברצונכם להריץ את השרת מקומית על המחשב שלכם, הכתובת תהייה localhost
2. הפורט בשרת אליו ניתן להתחבר.  
עבור השרת באוניברסיטה הפורט הוא אחד מהבאים: 8000, 8001, 8002, 8003. עבור הרצה מקומית של השרת הזינו פורט כלשהו, למשל 5678.
3. שם המשתמש
4. שם הקבוצה אליה המשתמש מעוניין להתחבר

דוגמא לפקודת ההרצה של הלקוח:

```
python client.py e-intro.cs.huji.ac.il 8000 Shira group1
```

## **פעולות הלקוח:**

1. הרצת הממשק הגרפי:  
את הממשק הגרפי תממשו ע"י שימוש בסיפריית tkinter ועליו לכלול את האלמנטים הבאים:
  1. שם המשתמש ושם הקבוצה אליה הוא מחובר
  2. כפתור Help, שלחיצה עליו תציג חלון עם טקסט שמתאר כיצד להשתמש באפליקציה
  3. אפשרות לבחור צבע מתוך הרשימה "blue", "red", "green", "yellow", "black", "violet", "orange"
  4. אפשרות לבחור את אחת הצורות הבאות: קו, מרובע, משולש, או אליפסה
  5. רשימה של המשתמשים המחוברים השייכים לקבוצה
  6. לוח עליו מצויירות הצורות שהוסיפו המשתמשים השונים כאשר ליד כל צורה מופיע שם המשתמש שהוסיף אותה. גובה הלוח הוא 500 ורוחבו 500. (width=500, height=500)
  7. הצגה של הודעת שגיאה שנשלחת מהשרת בחלון הודעה מיוחד (פרוטוקול ההודעה מפורט בהמשך).

אופן השימוש בממשק:

המשתמש בוחר צורה וצבע. לאחר מכן הוא מסמן על הלוח את הקואורדינטות בהן הוא רוצה שהצורה תופיע:

- עבור קו, יש לסמן שתי נקודות
- עבור מלבן, יש לסמן שתי נקודות שמהוות את האלכסון של המרובע
- עבור אליפסה, יש לסמן שתי נקודות שמהוות את האלכסון של המלבן שחוסם את האליפסה
- עבור משולש יש לסמן שלוש נקודות עבור שלושת הקודקודים

עם סימון הנקודה האחרונה תופיע הצורה בצבע שנבחר, כאשר ליד אחת מהקואורדינטות יופיע שם המשתמש.

2. התחברות לשרת.
3. שליחת הודעות לשרת (עפ"י הפרוטוקול שמפורט בהמשך):
  1. מיד לאחר ההתחברות לשרת שליחת הודעת התחברות שם המשתמש והקבוצה אליה הוא מתחבר.
  2. לאחר הוספה של צורה ללוח שליחת הודעה עם פירוט הצורה שנוספה
  3. כאשר הלקוח מתנתק (כלומר סוגר את הממשק הגרפי) שליחה של הודעת התנתקות
4. קבלת הודעות מהשרת:
  1. לאחר ההתחברות של הלקוח לשרת השרת שולח ללקוח:
    - א. הודעה עם שמות המשתמשים המחוברים
    - ב. הודעה עם הצורות השונות הקיימות על הלוח
  2. הודעות הצטרפות של משתמשים חדשים לאותה קבוצה
  3. הודעות על צורות שמשתמשים אחרים בקבוצה הוסיפו ללוח על מנת שהלקוח יוכל לעדכן את הממשק הגרפי בהתאם
  4. הודעה על התנתקות של אחד המשתמשים המחוברים על מנת ששמו יוסר מרשימת המשתמשים המחוברים.
  5. הודעה על שגיאה

### השרת:

השרת ממומש בקובץ server.py המצורף.

### פרמטרים שיש לתת לשרת בזמן הרצת הקובץ (לפי הסדר):

1. פורט.  
עבור השרת באוניברסיטה הפורט הוא אחד מהפורטים 8000, 8001, 8002, 8003. עבור הרצה מקומית של השרת הזינו פורט כלשהו, למשל 5678.

עבור הרצה מקומית על המחשב שלכם:

python server.py 5678

### פעולות השרת:

1. השרת מאזין ומקבל הודעות שונות מהלקוחות השונים בהתאם לפרוטוקול ההודעות המפורט בהמשך.
2. השרת שולח הודעות ללקוחות המחוברים השייכים לקבוצה מסוימת כאשר נעשה עידכון במצב הלוח או ברשימת המשתמשים המחוברים של אותה קבוצה
3. השרת מחזיק את רשימת הקבוצות הפעילות. ברגע שקבוצה כלשהי ריקה (כלומר כל המשתמשים שהיו בה אינם מחוברים יותר) הקבוצה תימחק.
4. בכל קבוצה השרת מחזיק את רשימת המשתתפים המחוברים ואת רשימת הצורות שמופיעות על הלוח המשותף לאותה קבוצה.
5. ברגע שמתחבר משתמש חדש השרת שולח אליו את רשימת המשתמשים המחוברים ורשימת הצורות הקיימות על הלוח המשותף של הקבוצה אליה הצטרף

6. כאשר לקוח מתחבר לקבוצה בעלת שם שאינו קיים ברשימת הקבוצות, תיווצר קבוצה חדשה עם השם הנתון.
7. השרת שולח הודעת שגיאה כאשר מתרחשת שגיאה כל שהיא, לדוגמא:
- א. שם לא תיקני (לא מכיל רק אותיות ומספרים ו.או אורכו גדול מ 20)
  - ב. הודעה באורך גדול מ 200 תווים
  - ג. ניסיון לשלוח הודעה שאינה הודעת התחברות עם שם קבוצה שאינו קיים
  - ד. שליחה יותר מפעם אחת של הודעת התחברות של לקוח
  - ה. שליחת הודעה של לקוח שאינו מחובר

### ההודעות ופרוטוקול ההודעות:

ישנם חמישה סוגי הודעות. כל הודעה שמועברת בין השרת והלקוח מכילה מספר סוגי מידע. כל סוג מידע שמועבר בהודעה אחת מופרד ע"י התו ';'

**חשוב: כל הודעה מסתיימת תמיד בתו '\'**

הודעות שנשלחות ע"י הלקוח:

סוג ההודעה	מבנה ההודעה
הצטרפות	join;username;group
צורה	shape;line;coordinates;color
עזיבה	leave

הודעות שנשלחות ע"י השרת:

סוג ההודעה	מבנה ההודעה
הצטרפות	join;username
צורה	shape;username;line;coordinates;color
עזיבה	leave:username
רשימת משתמשים	users;username1,username2
שגיאה	error;error_message

### סוגי ההודעות:

1. הודעת הצטרפות. במימוש הלקוח עליכם:
1. לשלוח הודעה זו עם התחברותו לשרת

2. לקבל הודעה זו מהשרת בכל פעם שמתחבר משתמש חדש לקבוצה. (השרת יודע לקבל הודעה זו ולהפיץ אותה לכל המשתמשים המחוברים לקבוצה).

פרוטוקול ההודעה:

ההודעה תופיע כמחרוזת אחת, ותכיל את המידע הבא (לפי הסדר):

- זוהי הודעת הצטרפות, לכן ההודעה זו תתחיל במילה 'join'.
- שם המשתמש
- כאשר ההודעה נשלחת מהלקוח, הקבוצה אליה הוא מצטרף

לדוגמא:

הודעה מהלקוח:

"join;Roni;group1"

הודעה מהשרת:

"join;Roni"

2. הודעה על הוספת צורה ללוח

במימוש הלקוח עליכם:

1. לשלוח הודעה זו עם הוספת צורה חדשה לשרת

2. לקבל הודעה זו מהשרת בכל פעם שמשתמש בקבוצה מוסיף צורה חדשה. (השרת יודע לקבל הודעה זו ולהפיץ אותה לכל המשתמשים המחוברים לקבוצה). רצף של הודעות צורה יישלחו למשתמש חדש עם התחברותו לשרת על מנת שיוכל לעדכן אצלו את מצב הלוח של הקבוצה אליה התחבר.

פרוטוקול ההודעה:

ההודעה תופיע כמחרוזת אחת, ותכיל את המידע הבא (לפי הסדר):

- זוהי הודעה של הוספת צורה, לכן תתחיל במילה 'shape'
- כאשר ההודעה נשלחת מהשרת, שם המשתמש שהוסיף את הצורה
- הצורה שנוספה. הצורות השונות מיוצגות באופן הבא: **המילה 'rectangle' למלבן, המילה 'triangle' למשולש, המילה 'oval' לאליפסה והמילה 'line' לקו.**
- קואורדינטות הצורה. כל זוג מספרים עוקבים מהווה קואורדינטת x וקואורדינטת y של נקודה. כל קואורדינטה מופרדת בתו ';'.
- צבע הצורה

לדוגמא:

הודעה שנשלחת מהשרת:

"shape;Roni;line;237,87,145,190;red"

הודעה שנשלחת מהלקוח:

"shape;line;237,87,145,190;red"

### 3. הודעה על התנתקות של משתמש

במימוש הלקוח עליכם:

1. לשלוח הודעה זו עם התנתקות הלקוח מהשרת (סגירת החלון הגרפי)
2. לקבל הודעה זו מהשרת בכל פעם שמשתמש בקבוצה עוזב. (השרת יודע לקבל הודעה זו ולהפיץ אותה לכל המשתמשים המחוברים לקבוצה).

פרוטוקול ההודעה:

ההודעה תופיע כמחרוזת אחת, ותכיל את המידע הבא (לפי הסדר):

- זוהי הודעה של הוספת עזיבת משתמש, לכן תתכיל במילה 'leave'
- שם המשתמש במידה וההודעה נשלחה מהשרת

לדוגמא:

הודעה שנשלחת מהשרת:

"leave:Roni"

הודעה שנשלחת מהלקוח:

"leave"

### 4. הודעה על רשימת משתמשים.

הודעה זו נשלחת ע"י השרת בלבד מיד לאחר התחברות הלקוח, ולכן במימוש הלקוח עליכם לדעת לקבל את ההודעה מהשרת.

פרוטוקול ההודעה:

ההודעה תופיע כמחרוזת אחת, ותכיל את המידע הבא (לפי הסדר):

- זוהי הודעה של רשימת משתמשים, לכן תתכיל במילה 'users'
- רשימת המשתמשים המחוברים בקבוצה, כאשר כל שם מופרד בתו ';'

לדוגמא:

"users;Shira,Ofer,Sharon"

### 5. הודעת שגיאה.

גם הודעה זו נשלחת ע"י השרת בלבד במקרה ומתרחשת אחת מהשגיאות שהוזכרו בחלק המתאר את פעולות השרת (סעיף 7).

פרוטוקול ההודעה:

ההודעה תופיע כמחרוזת אחת, ותכיל את המידע הבא (לפי הסדר):

- זוהי הודעה של רשימת משתמשים, לכן תתכיל במילה 'error'
- המחרוזת של הודעת השגיאה

לדוגמא:

"error;Not member of any group"

## הערות והכוונות:

1. מצורף מימוש של לקוח נוסף בקובץ `online_info_client.py`.  
כאשר לקוח זה מתחבר לשרת הוא מקבל חזרה את הקבוצות השונות ורשימת המשתתפים המחוברים בכל קבוצה.  
לקוח זה מקבל כפרמטרים את כתובת השרת והפורט.  
על מנת להריצו יש להשתמש בפקודה  
`python online_info_client.py e-intro.cs.huji.ac.il 8000`  
(אפשר להשתמש ב `localhost` ופורט כלשהו אחר עבור הרצה מקומית של השרת).  
לאחר ההתחברות לשרת הלקוח שולח הודעה מסוג `group`. השרת שולח בחזרה הודעה מסוג זה, עם רשימת הקבוצות המחוברות.  
לאחר מכן שולח הלקוח הודעה מסוג `who` אם שם הקבוצה ומקבל בחזרה הודעה מסוג `users` עם שמות המשתתפים המחוברים.  
רשימת הקבוצות והמשתתפים בהם מוחזרת כמילון שהמפתחות שלו הם שמות הקבוצות והערכיה הם רשימת המשתתפים בכל קבוצה.

2. כאשר מריצים את הממשק הגרפי לאחר הקריאה ל `mainloop()` אנחנו נמצאים למעשה בלולאה.  
בנוסף, על מנת להיות מסוגלים לקבל הודעות מהשרת עלינו לרוץ בלולאה שכל כניסה אליה בודקת האם התקבלו הודעות חדשות. זה עלול לגרום לכך שריצת לולאה אחת מתנע את ריצתה של השנייה.  
דרך אחת שמאפשרת להתמודד עם מצב כזה היא השימוש בפונקציה `after` של `tkinter`:

```
class App:
    def __init__(self, master):
        self.master = master
        self.poll() # start polling

    def poll(self):
        ... do something ...
        self.master.after(100, self.poll)
```

<http://effbot.org/tkinterbook/widget.htm#Tkinter.Widget.after-method>

3. כאשר רוצים לסגור את האפליקציה יש ללחוץ על הכפתור המובנה `X` שנמצא בפינה הימנית העליונה של הממשק הגרפי. כאשר המשתמש מבצע את הפעולה הזו הוא למעשה מתנתק מהשרת, ולכן צריכה להשלח לשרת הודעת עזיבה, ובנוסף ה `socket` של הלקוח צריך להסגר. ניתן להוסיף לממשק הגרפי את הקריאה הבאה  
`root.protocol("WM_DELETE_WINDOW",on_closing)`  
כאשר `root` הוא אובייקט ה `GUI` של `tkinter`, ו `on_closing` היא פונקציה שממושת עלי ידי הלקוח, ומכילה את המימוש של שליחת הודעת עזיבה עפ"י הפרוטוקול שהוגדר, וסגירה של ה `socket` של הלקוח.

4. שימו לב כי העיצוב הגרפי במימוש שלכם יכול להיות על פי שיקולכם ואינו מחוייב כלל לזה שבדוגמא, אך הפונקציות חייבת להשמר.

5. על מנת להריץ את השרת ולקוחות שונים מאותו מחשב עליכם לפתוח טרמינל נפרד עבור כל הרצה (cmd בווינדוס) או להריץ את פקודת ההרצה עם התו & בסופה (בלינוקס)

## הגשת התרגיל:

את התרגיל יש להגיש בזוגות.

עליכם להגיש קובץ zip הנקרא ex12.zip ומכיל את הקבצים הבאים:

1. client.py

2. README

3. קובץ AUTHORS עם שמותיכם

4. כל קובץ אחר שהוספתם ונחוץ להרצת התוכנית. כמובן שעליכם לציין את שמות הקבצים בקובץ README ה

 **בהצלחה**