תרגיל בית 8

תכנות מונחה עצמים

הנחיות כלליות:

- קראו בעיון את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ ex8_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז. שלכם, כל 9 הספרות כולל ספרת ביקורת.
- מועד אחרון להגשה: כמפורסם באתר.
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון).
- אין לשנות את שמות הפונקציות והמשתנים שכבר מופיעים בקובץ השלד של התרגיל. כמו כן אין לשנות את חתימת הפונקציות (כלומר, הארגומנטים אותן הן מקבלות כקלט).
- היות ובדיקת התרגילים עשויה להיות אוטומטית, **יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).**
- אופן ביצוע התרגיל: שימו לב, בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף. קראו היטב(!!!) את ההוראות ואת קובץ השלד המצורף ועקבו אחר התיעוד המופיע בו. מלאו את ההוראות בהתאם. שימו לב מתי **להחזיר** מתי **להדפיס** ומתי **לזרוק שגיאה**.

מטרות התרגיל: הבנת תכנות מונחה עצמים וזרימה של קוד. כתיבת פונקציות/מתודות כחלק *בלתי תלוי* בשאר הקוד ושימוש בהן.

אפליקציית צ'אט קבוצתי

בתרגיל זה תפתחו אפליקציה משוכללת בשם "UpWhats" המאפשרת צ'אט קבוצתי! האפליקציה מאפשרת לכל משתמש (מתוך רשימת משתמשים מוגדרת מראש) לשלוח הודעות טקסט קצרות לשאר חברי הקבוצה.

החדשנות באפליקציה שלנו (בניגוד למתחרים) הינה שכל משתמש רשאי למחוק הודעות גם שלו וגם של משתמשים אחרים! וכן היא אינה "מציפה" את המסך בהודעות ללא הרף אלא רק לבקשת המשתמש. בנוסף, מעת לעת, תוכן השיחה מגובה ב"cloud" (בתפקיד הcloud יהיה לא אחר מאשר קובץ רגיל על המחשב!).

לצורך הפשטות, השימוש באפליקציה יבוצע על ידי כל המשתמשים המורשים מאותו המכשיר (המחשב שלנו) ולא כל אחד ממכשירו האישי. לאחר יצירת השיחה/קבוצה, המשתמש יזין את שמו ויוכל לבצע פעולות במידה והינו חבר בקבוצה ע"י הקלדה את הפעולה הרצויה מתוך תפריט טקסטואלי מוגדר מראש. תפריט האפליקציה מאפשר יציאה מהתוכנית, הצגת כל ההודעות בשיחה הקבוצתית, שליחת הודעה חדשה ומחיקה של הודעה קיימת לפי מספרה הסידורי (הייחודי לה). האפליקציה תנהל את רשימת ההודעות ברקע. בסוף דף התרגיל מובא פלט לדוגמא של הרצה של התוכנית. להלן התפריט הראשי של האפליקציה:

```
##################################################
Welcome to UpWhats! What would you like to do?
[0] End conversation
[1] Show full conversation
[2] Send new message
[3] Remove existing message

Please type your choice and press ENTER
```

הערה: בראש הקובץ עם שלד הקוד מופיעות מספר שורות המייבאות פונקציות ממודולים חיצוניים וכן מספר חיוויים של האפליקציה. פונקציות וחיוויים אלה ישמשו אתכם בפתרון (כל אחד יובהר בזמנו). **אל** תמחקו אותם....

כיצד תשמור האפליקציה את נתוני השיחה הקבוצתית בזיכרון ?

עליכם להגדיר מחלקות אשר ייצגו את הישויות הנדרשות ואח"כ ניצור מהמחלקות אובייקטים אשר יכילו את כל האינפורמציה של השיחה ויידעו לתפעל אותה. יש למלא את הקוד המתאים בהתאם להוראות על מנת שהאפליקציה תפעל כראוי (פרטים בהמשך). כבר עכשיו, אם תנסו להריץ את האפליקציה (F5 על קובץ השלד) תראו שהיא עובדת אבל באופן לא תקין. בקובץ השלד כתוב pass (שמסמן לפייתון לדלג הלאה) בכל אחד מהמקומות שעליכם לממש ולכן התוכנית ניתנת להרצה. עליכם להחליף את הpass בקוד המתאים ואז האפליקציה תפעל כראוי.

1. ממשו מחלקה מסוג Date (דומה לזו שראינו בתרגול אבל **לא** זהה).

למחלקה שלושה שדות:

- hour – שעה מסוימת ביממה המיוצגת ע"י טיפוס מסוג מחרוזת.
- minute – דקה מסוימת בשעה המיוצגת ע"י טיפוס מסוג מחרוזת.
- second – שניה מסוימת בדקה המיוצגת ע"י טיפוס מסוג מחרוזת.

למחלקה שתי מתודות:

- __init__(self, current_time) המקבלת מחרוזת של זמן מסוים ביממה המופרד ע"י פסיקים. למשל ניתן להניח שהקלט תקין ושאת השעה 16:23:07 תקבלו כמחרוזת "16,23,07". על המתודה לעבד את מחרוזת הקלט ולאתחל את השדות של self בהתאם.
- __str__(self) מחזירה ייצוג מחרוזת של הזמן. עבור השעה לעיל תוחזר המחרוזת "16:23:07".

2. ממשו מחלקה מסוג Message.

למחלקה ארבעה שדות:

- sender – שם השולח המיוצג ע"י טיפוס מסוג מחרוזת.
- content – תוכן ההודעה המיוצג ע"י טיפוס מסוג מחרוזת.
- date – זמן השליחה המיוצג ע"י טיפוס מסוג Date (כן כן, זה שהרגע הגדרנו!).

- id – מזהה הודעה (ייחודי) המיוצג ע"י טיפוס מסוג int (באמצעות מזהה זה נוכל, אם נרצה, למחוק את ההודעה, פרטים בהמשך).

למחלקה שלוש מתודות:

- __init__(self, sender, content, date, msg_id) המקבלת את האובייקט וערכים עבור שדותיו ומאתחלת אותו בהתאם (למען הסר ספק, ההשֵׁמות צריכות להתבצע As is, אין צורך ב"עיבוד" הקלטים).

  הערה: msg_id זו לא שגיאה. id זה שם שמור ולכן הפרמטר לא מועבר כid.
- __len__(self) המקבלת את האובייקט ומחזירה את אורכו כאשר אורך ההודעה הינו אורך התוכן שלה בלבד (ללא שם השולח/זמן השליחה/מזהה ההודעה, כל אלה, "עלינו").
- __str__(self) מחזירה ייצוג מחרוזת של ההודעה. לדוגמא, David שלח Howdy! בשעה 16:23:05, ונניח שמזהה ההודעה הינו 1. אזי ייצוג ההודעה כמחרוזת ייראה כך: "(1) 16:23:05 David: Howdy!"

3. ממשו מחלקה מסוג Conversation.

למחלקה שבעה שדות:

- members – שמות חברי הקבוצה המיוצג ע"י tuple של מחרוזות.
- size_limit – כמות האחסון המקסימלית המוקצית לשיחה המיוצגת ע"י int.
- backup_policy – מדיניות הגיבוי (כל כמה הודעות תגובה השיחה בחשבון הcloud?) המיוצגת ע"י int.
- cloud_account - חשבון בו ניתן לגבות את תוכן השיחה המיוצג ע"י מחרוזת שהינה נתיב ל**תיקייה** חוקית/קיימת.
- size – כמות האחסון הנוכחית ב**תווי הודעה** (ולא במספר ההודעות, בכל זאת, אין סיבה שקבוצות השולחות הודעות תמציתיות תצטרכנה לשלם [כן, אנחנו בונים על השלמת הכנסה מהסיפור] עבור האחסון כמו קבוצות השולחות הודעות "חופרות") המיוצגת ע"י int.
- total_messages_sent – מספר ההודעות שנשלחו בשיחה עד כה המיוצג ע"י int. במספר זה נעשה שימוש כמזהה הודעה (בעת יצירת הודעה).

- content – תוכן השיחה עצמה המיוצג ע"י רשימה של הודעות (כל אחת מטיפוס Message כמובן).

למחלקה שלוש עשרה מתודות:

- __init__(self, members, size_limit, backup_policy, cloud_account_prefix)
המקבלת את האובייקט וערכים עבור שדותיו ובמידה והם תקינים, ע"פ המתודה is_valid (ראו מטה), מאתחלת אותו בהתאם. שלושת הראשונים As is, בחשבון הcloud יש לשים את cloud_account_prefix ולשרשר לו את שם הadmin (החבר הראשון בקבוצה) ולאחר מכן סיומת '.txt' (לקובץ הנמצא בנתיב זה תגובה השיחה בהתאם למדיניות הגיבוי, פרטים בהמשך).
הערה 1: יש להעביר את הערכים לis_valid **לפני** אתחול שדות האובייקט (נסו לחשוב על "קלט זדוני" בו אתחול האובייקט **לפני** בדיקת הערכים ע"י is_valid עלול לייצר שגיאה...).
הערה 2: מומלץ לתת כתחילית (prefix) כתובת הcloud את התיקיה בה נמצא קובץ הקוד שלנו. זה ימנע באגים מיותרים הקשורים לענייני נתיבים. המחרוזת המסמנת את הנתיב של התיקיה הנוכחית הינה './' (נקודה ואחריה סלאש רגיל, שנמצא בד"כ על אותו הכפתור במקלדת של "?"). תהיה התיקיה שבחרתם אשר תהיה, **לא לשכוח לשרשר** לתחילית זו את מה שדרוש לעיל. לגבי מספר הסלאשים, גם אם יהיו שניים ברצף זה יעבוד. אל דאגה.
עבור שלושת השדות הנותרים עליכם לבצע אתחול כרצונכם. רצוי שיהיה הגיוני...
- __len__(self) המקבלת את האובייקט ומחזירה את אורכו כאשר אורך השיחה הינו מספר ההודעות שנשלחו בשיחה (בלי קשר לאורך של כל אחת מהן).
- __str__(self) מחזירה ייצוג מחרוזת של השיחה. לדוגמא, David שלח Howdy! בשעה 16:23:05, ונניח שזו היתה ההודעה הראשונה מאז נוצרה השיחה. דקה וחצי אחריו, Steve שלח Gaudi אזי ייצוג השיחה כמחרוזת ייראה כך:
"(1) 16:23:05 David: Howdy!\n(2) 16:24:35 Steve: Gaudi"
- is_valid(self, member, size_limit, cloud_account, backup_policy) המקבלת את האובייקט ואת ערכי הפרמטרים שהתקבלו מהמשתמש וזורקת שגיאה מסוג ValueError במידה ואחד מהם אינו חוקי: (אחרת לא מחזירה דבר)
  - מספר החברים בקבוצה חייב להיות 2 ומעלה

- כמות האחסון המקסימלית חייבת להיות יותר מ10
- התיקייה לגיבוי חייבת להיות קיימת. על מנת לבדוק שהתיקייה קיימת תשתמשו בפונקצייה path_ok(path) המקבלת נתיב ומחזירה True אם הוא קיים וFalse אחרת. שימו לב אין צורך לממש את הפונקציה, נשתמש בפונקציה קיימת מהמודול os (ה"ייבוא" המתאים כבר בוצע עבורכם. הפונקציה מוכנה ומזומנה לשימוש).
- מדיניות הגיבוי צריכה להיות לפחות 1 (כלומר גיבוי השיחה לאחר כל שליחת הודעה, אין פחות מזה...).

- is_member(self, username) המקבלת את האובייקט ושם של משתמש ומחזירה True אם ורק אם שם המשתמש הינו אחד מחברי הקבוצה.
- enough_space(self, msg) המקבלת את האובייקט והודעה פוטנציאלית לשליחה (מסוג message) ומחזירה True אם ורק אם כמות האחסון בתווי הודעה לאחר השליחה לא תחרוג מכמות האחסון המקסימלית שהוקצתה לשיחה.
- is_empty(self) המקבלת את האובייקט ומחזירה True אם ורק אם השיחה ריקה, קרי, מספר ההודעות בשיחה כרגע הינו 0. שימו לב, השיחה יכולה להיות ריקה גם בעת יצירת השיחה וגם לאחר שליחת מספר הודעות ומחיקתן.
- time_for_backup(self) המקבלת את האובייקט ומחזירה True אם ורק אם זה הזמן לגבות את השיחה, כמובן בהתאם למדיניות הגיבוי ולמספר ההודעות שנשלחו **עד כה** (בין אם חלקן נמחקו ובין אם לאו).
- backup_content(self) המקבלת את האובייקט ומגבה את תוכן השיחה בחשבון הcloud, קרי, כותבת את מחרוזת השיחה לקובץ הרלוונטי. כנהוג בכתיבה לקבצים, נעטוף את ניסיון הכתיבה בבלוק של try. אם הפעולה הצליחה **נדפיס** למסך **שרשור** של ההודעה השמורה במשתנה backup_succeeded ושל חשבון הcloud אליו גובתה השיחה, אם הפעולה נכשלה בגלל שגיאה מסוג IOError נמנע את קריסת התוכנית ו**נדפיס** למסך את ההודעה השמורה במשתנה backup_failed (שימו לב שהמשתנים backup_succeeded ו-backup_failed לא מועברים לפונקציה כפרמטרים ואין לשנות זאת. איך בכל זאת ניגש אליהם? רמזים בקוד...). בכל מקרה, נוודא שחרור משאבים לאחר הפעולה. הפונקציה לא מחזירה דבר.

- get_conversation(self) המקבלת את האובייקט ואם השיחה לא ריקה מחזירה ייצוג מחרוזת של השיחה. אחרת, **מחזירה** את ההודעה השמורה במשתנה empty_conversation (שימו לב שהוא לא מועבר לפונקציה כפרמטר ואין לשנות זאת. איך בכל זאת ניגש אליו? רמזים בקוד...).
- send_msg(self, username, msg_content, msg_time) המקבלת האובייקט, שם משתמש, תוכן ההודעה (מטיפוס מחרוזת) ומועד שליחת ההודעה (מטיפוס Date). אם כמות האחסון בתווי הודעה לאחר השליחה תחרוג מכמות האחסון המקסימלית שהוקצתה לשיחה יש לזרוק שגיאה מסוג MemoryError (ולא לשלוח את ההודעה כמובן). אחרת, יש לשלוח את ההודעה. שליחת ההודעה תתבצע ע"י:
  - עדכון שדות השיחה הרלוונטיים-
    - מספר ההודעות שנשלחו בשיחה עד כה.
    - תוכן השיחה עצמה (ע"י הוספת הודעה מתאימה חדשה מטיפוס Message כמובן).
    - כמות האחסון הנוכחית בתווי הודעה.
  - גיבוי תוכן השיחה בcloud במידה והגיע הזמן על פי מדיניות הגיבוי.
  - **החזרת** ההודעה השמורה במשתנה sending_succeeded_msg (שימו לב שהוא לא מועבר לפונקציה כפרמטר ואין לשנות זאת. איך בכל זאת ניגש אליו? רמזים בקוד...).
- find_msg_index(self, msg_id) המקבלת את האובייקט ומזהה הודעה (המיוצג כ-int) ואם קיימת הודעה בשיחה עם מזהה זה תחזיר את האינדקס שלה ברשימה (החל מ-0 כמובן), אחרת תחזיר 1-. שימו לב, בגלל היכולת למחוק הודעות, המזהה לא בהכרח תואם לאינדקס ההודעה בשיחה! למשל אם ההודעה הראשונה (אינדקס 0, מזהה 1) נמחקה אזי ההודעה השניה (עם מזהה 2) הופכת להיות במיקום הראשון בשיחה (אינדקס 0).
- delete_msg(self, msg_id_str) המקבלת את האובייקט ומזהה הודעה (המיוצג כ-str). אם אין הודעה בשיחה אם מזהה כזה (אולי כי היא כבר נמחקה ואולי כי עוד לא נשלחו מספיק הודעות) נזרוק שגיאה מסוג ValueError (ולא למחוק דבר כמובן). אחרת, יש למחוק את ההודעה. מחיקת ההודעה תתבצע ע"י:
  - עדכון שדות השיחה הרלוונטיים:

- תוכן השיחה עצמה (ע"י מחיקת ההודעה המתאימה).
- כמות האחסון הנוכחית בתווי הודעה.

○ **החזרת** ההודעה השמורה במשתנה removing_succeeded_msg (שימו לב שהוא לא מועבר לפונקציה כפרמטר ואין לשנות זאת. איך בכל זאת ניגש אליו? רמזים בקוד...).

<u>רשות:</u> הקובץ CodeTest.py מכיל מספר בדיקות שתעזורנה לכם לוודא את תקינות הקוד שלכם. כל שורה עם המילה assert היא בעצם טסט (test). הפקודה assert זורקת שגיאה בכל פעם שהביטוי שאחריה אינו True. כלומר, אם המימושים שלכם תקינים, הביטוי אחרי <u>כל</u> assert ישתערך לTrue וכל הטסטים יעברו. אם קיבלתם שגיאה מאחד הטסטים, זה מאוד ימקד אתכם בחיפוש אחר הבאג. על מנת להריץ את הבדיקות עליכם להעתיק את התוכן של הקובץ הנ"ל ולהדביקו בתחתית הקובץ ex8_012345678.py במקום המתאים (ראו הערה מתאימה בקוד). אין זה אומר שאלו כל הבדיקות האפשריות. יכול (מאוד) להיות שנבצע בדיקות נוספות. בכל אופן, לאחר שתסיימו, הגישו את הקוד שלכם ללא בדיקות הרשות הללו.

לבסוף, אם מימשתם הכל כראוי תוכלו להפעיל את האפליקציה ע"י לחיצת F5.

איך בפועל הכל מתחבר בסוף? בקצרה, בתחתית הקובץ מאותחלת האפליקציה (המחלקה Application כבר מומשה עבורכם) ומורצת (באמצעות המתודה run). התפריט מודפס וניתן לבצע פעולות באופן אינטראקטיבי. אם הפונקציות שנתבקשתם לממש עובדות כדרוש, האפליקציה תעבוד חלק! לפרטים נוספים- מוזמנים לעיין בקוד...

בהצלחה!

☺

```
>>>
============== RESTART: D:/Py4Eng/1617a/Ex8/Ex8_012345678.py ==============
Please enter the number of members in the group. Please enter the number of
members in the group.
3
Please enter member number 1:
Lena
Please enter member number 2:
Ilan
Please enter member number 3:
Dvir
Please enter your storage limit (int):
20
Please a desired backup policy (int):
2
Please enter a file path for backing up your data:
./

##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
1
Please enter username (only conversation's members are allowed to send/read
messages).
Lena
Conversation either didn't start or all messages were removed.


##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
```

```
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
2
Please enter username (only conversation's members are allowed to send/read
messages).
Ilan
Please type your message.
Hi!
Message was sent successfully!

2
##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
2
Please enter username (only conversation's members are allowed to send/read
messages).
Dvir
Please type your message.
Bye
Data was backed up successfully to:
.//Lena.txt
Message was sent successfully!


##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
```

```
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
1
Please enter username (only conversation's members are allowed to send/read
messages).
Lena
(1) 13:28:17 Ilan: Hi!
(2) 13:28:42 Dvir: Bye

##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
3
Please enter username (only conversation's members are allowed to send/read
messages).
Ilan
Please enter message id.
2
Message was removed successfully!


##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message
```

```
Please type your choice and press ENTER
1
Please enter username (only conversation's members are allowed to send/read
messages).
Lena
(1) 13:28:17 Ilan: Hi!

##################################################
Welcome to UpWhats! What would you like to do?
        [0] End conversation
        [1] Show full conversation
        [2] Send new message
        [3] Remove existing message

Please type your choice and press ENTER
2
Please enter username (only conversation's members are allowed to send/read
messages).
Dvir
Please type your message.
123456789012345678
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Anaconda\lib\site-
packages\spyderlib\widgets\externalshell\sitecustomize.py", line 714, in
runfile
    execfile(filename, namespace)
  File "D:\Anaconda\lib\site-
packages\spyderlib\widgets\externalshell\sitecustomize.py", line 74, in
execfile
    exec(compile(scripttext, filename, 'exec'), glob, loc)
  File "D:/Py4Eng/1617a/Ex8/Ex8_012345678.py", line 294, in <module>

  File "D:/Py4Eng/1617a/Ex8/Ex8_012345678.py", line 205, in run
    response = self.coversation.send_msg(username, msg, msg_time)
  File "D:/Py4Eng/1617a/Ex8/Ex8_012345678.py", line 108, in send_msg
```

```
    raise MemoryError
MemoryError
```