### מבוא לתכנות מונחה עצמים-מטלות 1+0.

מגישים: בן הורן 208569848 נועה חדד ת.ז. 208569848

#### הקדמה

בחרנו באפליקציית WigGle על מנת לאסוף מידע על נקודות wifi ברחבי האוניברסיטה. אפליקציה זו טיפקה לנו את כל הנתונים הנדרשים למטלה, וכמו כן אפשרה לנו לייבא את המידע לקובצי csv.

עיבוד המידע נעשה באמצעות התכנית שכתבנו:

#### מבנה התכנית-

- . q3 ,q2 ,Wifi ,main מחלקות (1
  - . filter ממשק (2
- 3) מחלקות בדיקות WifiTest,q2Test,q3Test.

### מחלקת Wifi-

יצרנו טיפוס מסוג Wifi, אשר מכיל את המשתנים הבאים: Wifi, LAT ,LON ,ALT ,SSID ,MAC, המשתנים הבאים: Frequency ,signal.

למחלקה יש מספר מתודות, כולן ציבוריות:

- 1) בנאי אשר מקבל את הנתונים על נקודת ה-wifi ומכניס אותם לתוך המשתנים.
  - 2) מתודות (.../get (Time/IT/LAT...) המחזירות את הנתון המבוקש.
- 3) מתודת (equals(Wifi other) המשווה בין שתי נקודות wifi, ומחזירה אמת אם הן זהות, אחרת מחזירה שקר.
- 4) מתודה סטטיק (-..Correct(Time,ID,LAT, המקבלת נתונים על נקודת wifi ובודקת את תקינותם. מחזירה אמת אם הנתונים תקינים, ושקר אחרת.
  - 5) מתודת (getCoordinates- המחזירה מחרוזת של נקודות LAT,LON.
- ומוסיפה לו ייצוג -kmlGenerator(Document doc) מתודת (6 -kml אובייקט bocument) מקבלת אובייקט אובייקט הואיפיי בפורמט

### <u>-q2 מחלקת</u>

המחלקה הזו בנויה כולה מפונקציות.

## במחלקה זו –

- .wifi שיובאו מהאפליקציה) את המידע על נקודות (שיובאו מהאפליקציה) את המידע על נקודות (1
  - 2) בודקים את תקינות המידע.
- 3) שומרים את המידע במבנה נתונים, וזאת לשם כתיבתו לקובץ חדש בפורמט הנדרש.

#### הפונקציות:

- 1) פונקציה פרטית סטטית (String folderPath) פונקציה פרטית סטטית (מחזירה מערך של קבצים במידה -findCsvFiles) את מיקום התיקייה שבא אמורים להימצא קבצי הcsv. מחזירה מערך של קבצים במידה וקיימים קבצים כאלה בתיקייה, ו- null
- getFileFormat (File f) פונקציה פרטית סטטית (2 -getFileFormat) פונקציה מקבלת קובץ ומחזירה את סוגו.

- -LinkedList<Wifi>fileToList(File[] listOfFiles) פונקציה פרטית סטטית (3 csv) מחזירה רשימה מקושרת שכל איבר בה הוא הפונקציה מקבלת מערך של קבצי Wifi בודדת.
- 4) פונקציה פרטית סטטית (String str) פונקציית עזר לפונקציה בסעיף הקודם. הפונקציה מקבלת מחרוזת של שורה מקובץ הcsv, ובודקת אם היא בעלת נתונים חלקיים/מיותרים. באמצעות פונקציה זו מסננים נקודות wifi . הפונקציה תחזיר אמת אם המידע תקין, ושקר אחרת.
  - -listToCSV(LinkedList<Wifi>wifis,String CSVpath) פונקציה פרטית סטטית (5 את הרשימה המקושרת עם המידע על נקודות הWifi, את הרשימה המקושרת עם המידע על נקודות הcsv אונדע לקובץ csv חדש בפורמט הנדרש. הקובץ נשמר תחת
- 6) פונקציה ציבורית סטטית (toCSV(String folderPath, String CSVpath) הפונקציה מקבלת כתובת של תיקייה. באמצעות הפונקציות האחרות במחלקה: היא בודקת אם יש בה קבצי csv, בודקת את תקינות המידע בקבצים, ומארגנת אותו לתוך קובץ csv חדש שנשמר בכתובת השנייה שקיבלה. הפונקציה מדפיסה הודעת אישור על הצלחת יצירת הקובץ החדש, או לחלופין הודעת שגיאה אם נכשל .
  - לוnkedList <Wifi> list (String folderPath) פונקציה ציבורית סטטית (לואמת. הפונקציה מקבלת כתובת של תיקייה, ומחזירה רשימה פונקציית עזר למחלקה הקודמת. הפונקציה זו נבנתה על מנת שנוכל לבדוק את נכונות מקושרת של נקודות שנתקבלו מהפונקציה הפרטית "fileToList".

# <u>מחלקת q3</u>-

המחלקה הזו גם בנויה כולה מפונקציות.

### במחלקה זו-

- 1) מסננים את המידע שבקובץ ה csv. בוחרים לפי איזה נתון (זמן/מקום/מזהה) רוצים לסנן ומהו המידע שרוצים למצוא בנתון זה.
- זהים-משאירים רק את הנקודה (2 מארגנים את המידע-כאשר נמצאות נקודות בעל נתוני MAC זהים-משאירים רק את הנקודה עם הסיגנל החזק ביותר.

#### הפונקציות:

- 1) פונקציה פרטית סטטית "csvtoList"- הפונקציה מקבלת את כתובת קובץ הcsv, הפרמטר לסינון והמידע הנדרש ממנו. הפונקציה מחזירה רשימה מקושרת עם נקודות Wifi שעונות לדרישות הסנן.
- (2) פונקציה בוליאנית (Fit (Wifi wifi, String data, String req). הפונקציה היא למעשה פונקציית עזר מיישמת את חתימת הפונקציה שבממשק filter. הפונקציה היא למעשה פונקציית עזר למחלקה הקודמת. היא מקבלת מידע על נקודת wifi, הפרמטר לסינון והמידע הנדרש ממנו. מחזירה אמת אם הנקודה עונה לבקשת הסנן, ושקר אחרת.
- (3) פונקציה פרטית סטטית "organize"-הפונקציה מקבלת רשימה מקושרת של נקודות wifi. בודקת אם קיימות בה נקודות בעלות אותו MAC. במידה וכן, בודקת למי מבניהן יש את הסיגנל הכי חזק, ובונה רשימה חדשה של הנקודות עם הסיגנל הכי חזק.
- וכתובת של קובץ WIFI פונקציה פרטית סטטית "kml" המקבלת רשימה מקושרת של נקודות WIFI וכתובת של קובץ . ושקר אחרת. KML הפונקציה מחזירה אמת אם הצליחה לייצא קובץ.
  - לקובץ CSVtokmL" פונקציה ציבורית סטטית "CSVtokmL" המקבלת כתובת של קובץ, CSV, כתובת של קובץ, KML, פרמטר לסינון והמידע הנדרש ממנו. הפונקציה מחזירה אמת אם יוצא קובץ KML, ושקר אחרת.
  - 6) פונקציה ציבורית סטטית "list" המקבלת כתובת של קובץ CSV, פרמטר לסינון והמידע הנקציה ציבורית סטטית "list" המקבלת כתובת של נקודות WIFI העונות על הנדרש. הפונקציה מחזירה רשימה מקושרת של נקודות התוצר של הפונקציה הפונקציה משמשת פונקציית עזר לקודמתה, וכן מאפשרת לבחון את התוצר של הפונקציה הפרטית "csytolist".

7) פונקציה ציבורית סטטית "listOrganized", שמקבלת רשימה מקושרת ומחזירה רשימה רק חדשה מאורגנת, כלומר במידה ויש בה נקודות בעלות אותו MAC , היא תכניס לרשימה רק את הנקודה בעלת הסיגנל הגבוהה ביותר מביניהן. הפונקציה משמשת פונקציית עזר ל(5), וכן מאפשרת לבחון את התוצר של הפונקציה הפרטית "organize".

מחלקת ()Main- קוראת לפונקציות האחרות.

<u>ממשק Filter</u>- מכיל חתימה של פונקציה בוליאנית בשם "fit". הפונקציה מקבלת טיפוס מסוג Wifi, מחרוזת שמייצגת את הנתון שלפיו רוצים לסנן ומחרוזת המייצגת את המידע המבוקש.

### <u>מחלקות בדיקות</u>

לכל אחת מהמחלקות q2,q3,Wifi, נבנתה גם מחלקת בדיקות. העיקרון המנחה היה לבדוק עבור מקרים מסוימים/מקרי קצה, אם מה שצפינו שהפונקציה תעשה –זה אכן מה שהיא ביצעה בפועל.

לצורך הבדיקות, נבנו 3 תיקיות בדיקות וקובץ csv:

"testFolder1"- תיקייה ריקה.

"testFolder2"-תיקייה עם קובץ טקסט.

### כלי תוכנה

לצורך הפעלת התכנית השתמשו במספר כלי תוכנה:

יבוא java.io.File - על מנת לגשת לקבצים.

יבוא את הקבצים. - java.io.FileReader ,java.io.BufferedReader

יבוא java.io.lOException ,java.io.FileNotFoundException- על מנת "לתפוס" חריגות בעת הרצת התוכנית.

יבוא java.util.LinkedList על מנת להשתמש במבנה נתונים בדמות רשימה מקושרת.

יבוא \*.org.junit.Assert - על מנת ליצור מחלקת בדיקות.

יבוא סיפריות JAK-java API kml המשמשת ליצירת קובץ kml לפי חוקי הפורמט. קובץ הkml מייצג נקודות על מפה, במקרה שלנו את נקודות הifin. על מנת לייצא קובץ kml מרובה נקודות, בחבילת נקודות על מפה, במקרה שלנו את נקודות הifin. על מנת לייצא קובץ placemark ולייצג גם יותר מנקודה jakn נכללת גם סיפריית Document, בה ניתן להוסיף מספר placemark נוסף תיאור (Descriprion), בתיאור הנקודה צורף שם נקודת הוואיפיי, עוצמת הנקודה, הmac ושאר נתוני הוואיפיי. לכל placemark נוסף מיקום הנקודה בקורדינטות בעזרת פונקציית withCoordinates וחתימת זמן (תאריך ושעה) בעזרת SetTimeStamp.

<sup>&</sup>quot;testFolder3"-תיקייה עם קובץ csv, שלפיו נבחן אם המחלקות מעבדות את המידע כנדרש.

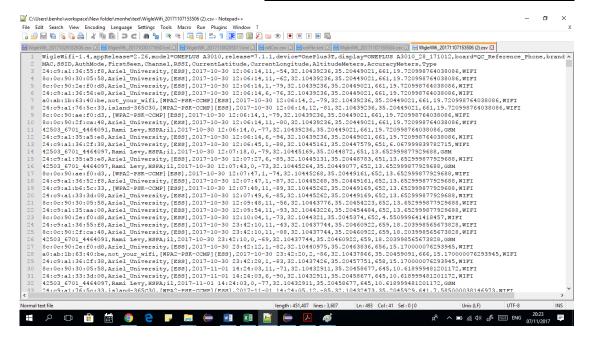
<sup>&</sup>quot;expectedCsv.csv"-קובץ csv קובץ csv". "expectedCsv.csv".

#### <u>הניסוי</u>

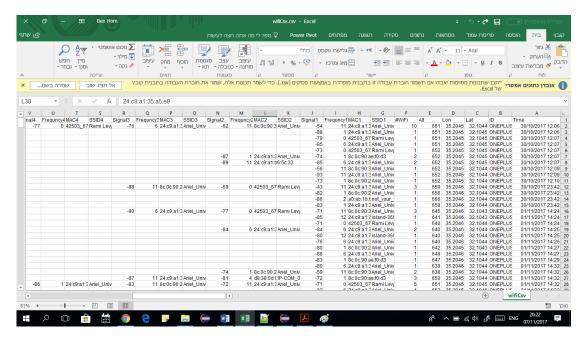
בניסוי הדלקנו את אפליקציית קליטת נתוני הWifi במכשיר הטלפון למשך יום לימודים שלם- ממעונות הסטודנטים עד לכיתות הלימוד וחזרה. לאחר מכן ייצאנו לקובץ CSV והפעלנו את התוכנה. כך נראה הייצוא לקובץ ID, OPPO3T על גבי מפה (לאחר הכנסת פילטר החיפוש: ID, OPPO3T - הID של המכשיר בו השתמשנו).

#### נספחי הניסוי

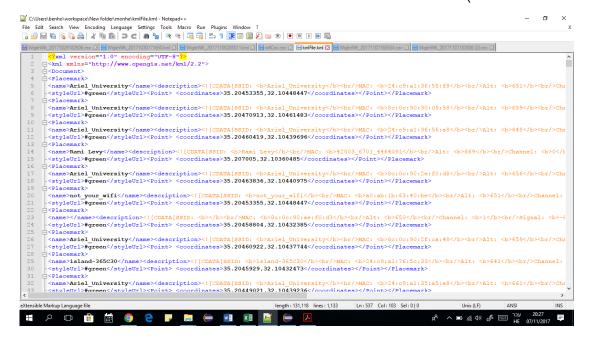
### הקובץ שיובא מאפליקציית wigGLE שבמכשיר הפלאפון.



"g2.toCSV" שנוצר בעקבות הקריאה לפונקציה CSV קובץ



שנוצר בעקבות הקריאה לפונקציה "q3.CSVtoKML" שנוצר בעקבות הקריאה לפונקציה ''q3.CSVtoKML שנוצר בעקבות הקריאה לפונקציה ''ממכשירים אחרים' .



# פתיחת קובץ KML פתיחת קובץ

ניתן לראות את נקודות ה WIFI ברחבי האוניברסיטה ואת המידע עליהן.

