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 בחרנו אפליקציית של מנת לאסוף מידע על נקודות csv. סיפקה לנו את כל הנתונים הנדרשים למטלה, וכמו כן אפשרה לנו לייבא את המידע לקובצי

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...) מתודות (2
- 3) מתודת (equals(Wifi other -המשווה בין שתי נקודות wifi, ומחזירה אמת אם הן זהות, אחרת מחזירה שקר.
 - 4) מתודת (String toString)- המדפיסה את הנתונים על נקודת ה-wifi.
- 5) מתודה סטטיק (-..Correct(Time,ID,LAT, המקבלת נתונים על נקודת wifi ובודקת את תקינותם. מחזירה אמת אם הנתונים תקינים, ושקר אחרת.
 - 6) מתודת (getCoordinates) המחזירה מחרוזת של נקודות
- 7) מתודת (kmlGenerator(Document מקבלת אובייקט -kmlGenerator) מתודת (f kml נקודת הוואיפיי בפורמט

מחלקת q2-

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:

- לותdCsvFiles (String folderPath) פונקציה פרטית סטטית (1 לותקציה מקבלת -findCsvFiles) את מיקום התיקייה שבא אמורים להימצא קבצי היסטטית מחזירה מערך של קבצים במידה את מיקום התיקייה, ו- null אחרת.
- 2) פונקציה פרטית סטטית (getFileFormat(File f)- פונקציית עזר לפונקציה הקודמת. הפונקציה מקבלת קובץ ומחזירה את סוגו.

- -LinkedList<Wifi>fileToList(File[] listOfFiles) פונקציה פרטית סטטית (csv פונקציה מקבלת מערך של קבצי wifi בודדת.

 מטיפוס Wifi ומכיל מידע על נקודת 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 הפונקציה מקבלת את הרשימה המקושרת עם המידע על נקודות הWifi וכותבת את csv המידע 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 (לוביה ביבורית סטטית (אולקה הקודמת. הפונקציה מקבלת כתובת של תיקייה, ומחזירה רשימה מקושרת של נקודות wifi תקינות. פונקציה זו נבנתה על מנת שנוכל לבדוק את נכונות רשימת הנקודות שנתקבלו מהפונקציה הפרטית "fileToList".

מחלקת q3-

המחלקה הזו גם בנויה כולה מפונקציות.

במחלקה זו-

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

הפונקציות:

- 1) פונקציה פרטית סטטית "csvtoList"- הפונקציה מקבלת את כתובת קובץ ה,csv 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 KML הפונקציה מחזירה אמת אם הצליחה לייצא קובץ KML , ושקר אחרת.
 - לובץ אל קובץ CSVtokmL" פונקציה ציבורית סטטית "CSVtokmL" פונקציה ציבורית סטטית (5 אמת שמנו. הפונקציה מחזירה אמת אם יוצא קובץ KML, פרמטר לסינון והמידע הנדרש ממנו. הפונקציה מחזירה אמת אם יוצא קובץ toll, ושקר אחרת.
 - והמידע "list" פונקציה ציבורית סטטית "list" פונקציה ציבורית סטטית המקבלת כתובת של קובץ CSV, פרמטר לסינון והמידע הנדרש. הנדרש ממנו. הפונקציה מחזירה רשימה מקושרת של נקודות WIFI העונות על הנדרש.

- הפונקציה משמשת פונקציית עזר לקודמתה, וכן מאפשרת לבחון את התוצר של הפונקציה "csvtoList".
- 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:

<u>כלי תוכנה</u>

לצורך הפעלת התכנית השתמשו במספר כלי תוכנה:

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

יבוא את הקבצים. - java.io.FileReader ,java.io.BufferedReader - על מנת לקרוא

יבוא java.io.IOException, 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 נוקודת על מפה, במקר שלנו את נקודות הwifi.

[&]quot;testFolder1"- תיקייה ריקה.

[&]quot;testFolder2"-תיקייה עם קובץ טקסט.

[&]quot;testFolder3"-תיקייה עם קובץ csv, שלפיו נבחן אם המחלקות מעבדות את המידע כנדרש.

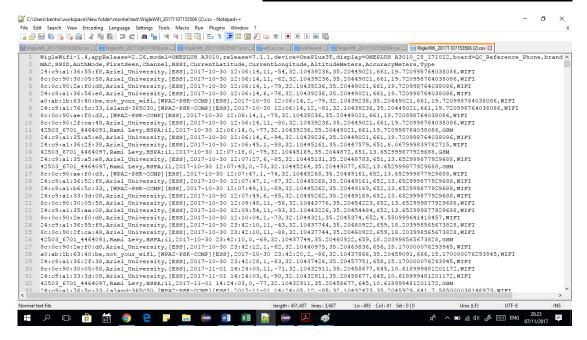
[&]quot;expectedCsv.csv"-קובץ csv קובץ "expectedCsv.csv".

הניסוי

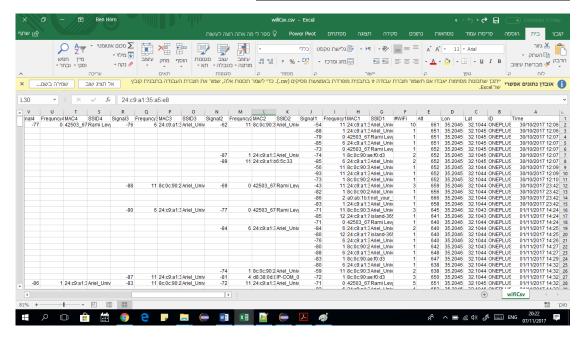
בניסוי הדלקנו את אפליקציית קליטת נתוני הWifi במכשיר הטלפון למשך יום לימודים שלם- ממעונות הסטודנטים עד לכיתות הלימוד וחזרה. לאחר מכן ייצאנו לקובץ CSV והפעלנו את התוכנה. כך נראה הייצוא לקובץ ID, OPPO3T על גבי מפה (לאחר הכנסת פילטר החיפוש: ID, OPPO3T - הID של המכשיר בו השתמשנו).

נספחי הניסוי

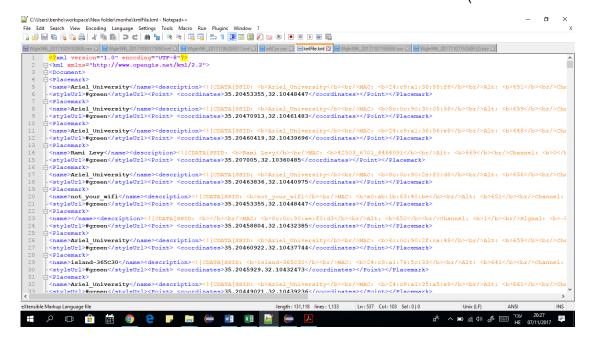
הקובץ שיובא מאפליקציית wigGLE שבמכשיר הפלאפון.



"q2.toCSV" שנוצר בעקבות הקריאה לפונקציה CSV קובץ



WIFI שנוצר בעקבות הקריאה לפונקציה "q3.CSVtoKML" שנוצר בעקבות הקריאה לפונקציה סיננה (הפונקציה טיננה נקודות אחרים) .



.google earth בKML פתיחת קובץ

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

