

הקדמה

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

עיבוד המידע נעשה באמצעות התכנית שכתבנו:

מבנה התכנית-

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

מחלקת Wifi-

יצרנו טיפוס מסוג Wifi, אשר מכיל את המשתנים הבאים: Time, ID, LAT, LON, ALT, SSID, MAC, 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/ID/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.
- 6) מתודת kmlGenerator(Document doc) - מקבלת אובייקט Document ומוסיפה לו ייצוג נקודת הוואיפיי בפורמט kml

מחלקת q2-

המחלקה הזו בנויה כולה מפונקציות.

במחלקה זו –

- 1) קוראים מקבצי ה-csv (שיובאו מהאפליקציה) את המידע על נקודות ה-wifi.
- 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) findCsvFiles - הפונקציה מקבלת את מיקום התיקייה שבא אמורים להימצא קבצי ה-csv. מחזירה מערך של קבצים במידה וקיימים קבצים כאלה בתיקייה, ו- null אחרת.
- 2) פונקציה פרטית סטטית (File f) getFileFormat - פונקציית עזר לפונקציה הקודמת. הפונקציה מקבלת קובץ ומחזירה את סוגו.

- (3) פונקציה פרטית סטטית `-LinkedList<Wifi>fileToList(File[] listOfFiles)` הפונקציה מקבלת מערך של קבצי csv, ומחזירה רשימה מקושרת שכל איבר בה הוא מטיפוס Wifi ומכיל מידע על נקודת Wifi בודדת.
- (4) פונקציה פרטית סטטית `legit(String str)` – פונקציית עזר לפונקציה בסעיף הקודם. הפונקציה מקבלת מחרוזת של שורה מקובץ csv, ובודקת אם היא בעלת נתונים חלקיים/מיותרים. באמצעות פונקציה זו מסננים נקודות wifi. הפונקציה תחזיר אמת אם המידע תקין, ושקר אחרת.
- (5) פונקציה פרטית סטטית `-listToCSV(LinkedList<Wifi>wifis,String CSVpath)` הפונקציה מקבלת את הרשימה המקושרת עם המידע על נקודות Wifi, וכותבת את המידע לקובץ csv חדש בפורמט הנדרש. הקובץ נשמר תחת `CSVpath`.
- (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.
- (7) פונקציה ציבורית סטטית `-LinkedList <Wifi> list(String folderPath)` פונקציית עזר למחלקה 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-

- (1) מסננים את המידע שבקובץ ה csv. בוחרים לפי איזה נתון (זמן/מקום/מזהה) רוצים לסנן ומהו המידע שרוצים למצוא בנתון זה.
- (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 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 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.
- (4) פונקציה פרטית סטטית `"kml"` המקבלת רשימה מקושרת של נקודות WIFI וכתובת של קובץ KML. הפונקציה מחזירה אמת אם הצליחה לייצא קובץ KML, ושקר אחרת.
- (5) 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 הנדרש ממנו. הפונקציה מחזירה רשימה מקושרת של נקודות WIFI העונות על הנדרש. הפונקציה משמשת פונקציית עזר לקודמתה, וכן מאפשרת לבחון את התוצר של הפונקציה הפרטית `"csvtoList"`.

7) פונקציה ציבורית סטטית "listOrganized", שמקבלת רשימה מקושרת ומחזירה רשימה 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.

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

מחלקות בדיקות

לכל אחת מהמחלקות Wifi, q2, q3, 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.

"testFolder3" - תיקייה עם קובץ csv, שלפיו נבחן אם המחלקות מעבדות את המידע כנדרש.

"expectedCsv.csv" - קובץ csv הסופי שמחלקה 2 צריך לייצר, כאשר היא ניגשת ל- "testFolder3".

כלי תוכנה

לצורך הפעלת התכנית השתמשו במספר כלי תוכנה:

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

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

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

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

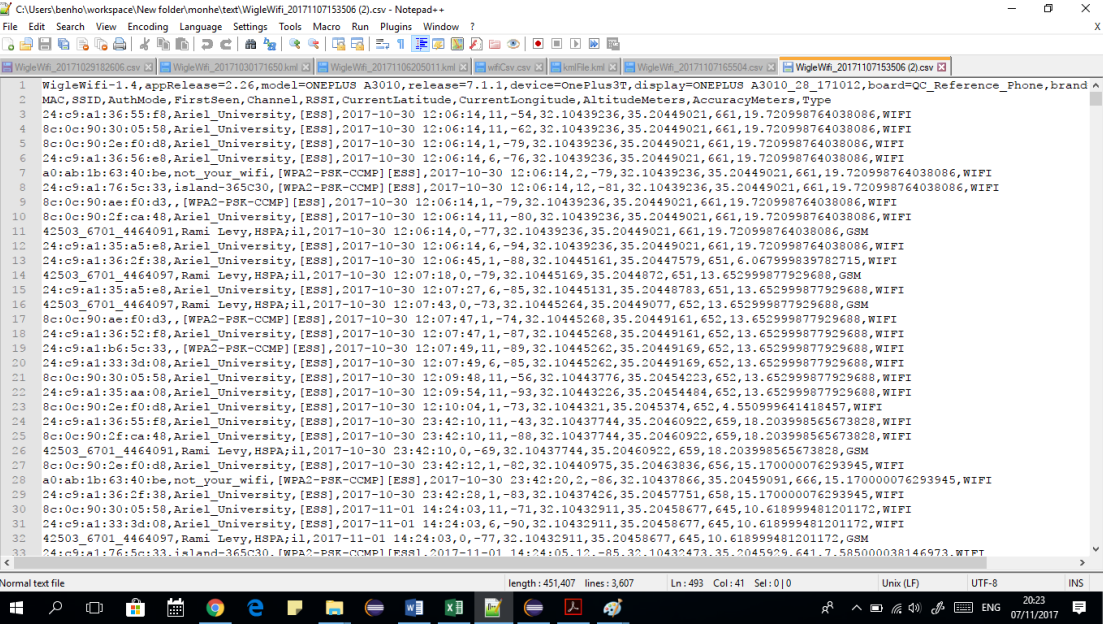
יבוא org.junit.Assert.* - על מנת ליצור מחלקת בדיקות.

יבוא ספריית JAK-java API kml המשמשת ליצירת קובץ kml לפי חוקי הפורמט. קובץ ה-kml מייצג נקודות על מפה, במקרה שלנו את נקודות ה-wifi.

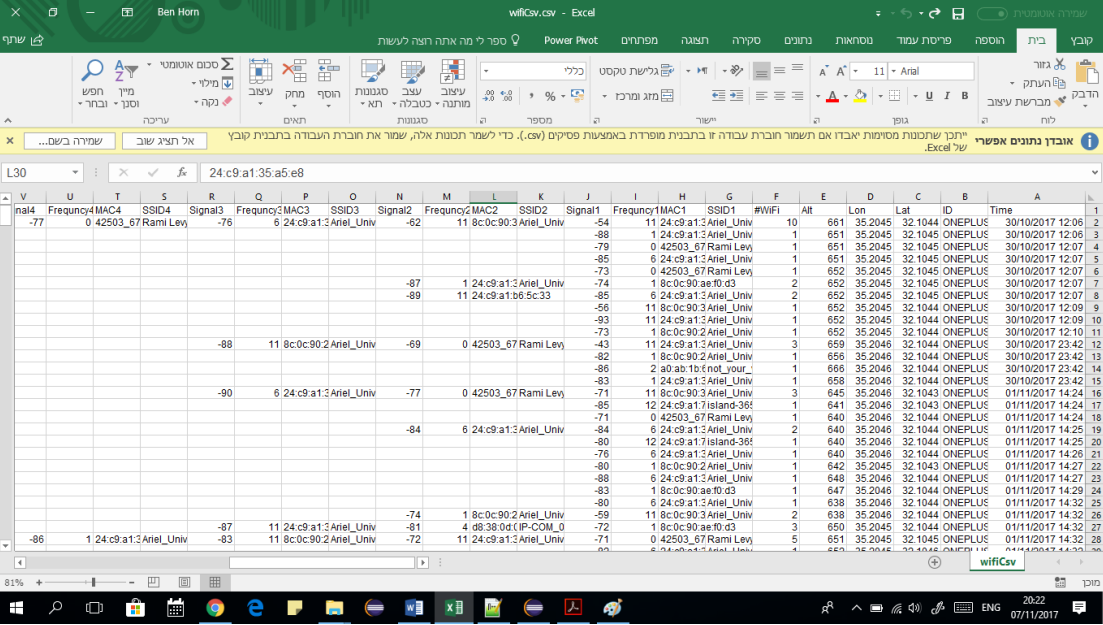
בניסוי הדלקנו את אפליקציית קליטת נתוני Wifi במכשיר הטלפון למשך יום לימודים שלם- ממעונות הסטודנטים עד לכיתות הלימוד וחזרה. לאחר מכן ייצאנו לקובץ CSV והפעלנו את התוכנה. כך נראה הייצוא לקובץ KML על גבי מפה (לאחר הכנסת פילטר החיפוש: ID, OPPO3T - ID של המכשיר בו השתמשנו).

נספחי הניסוי

הקובץ שיובא מאפליקציית wigGLE שבמכשיר הפלאפון.

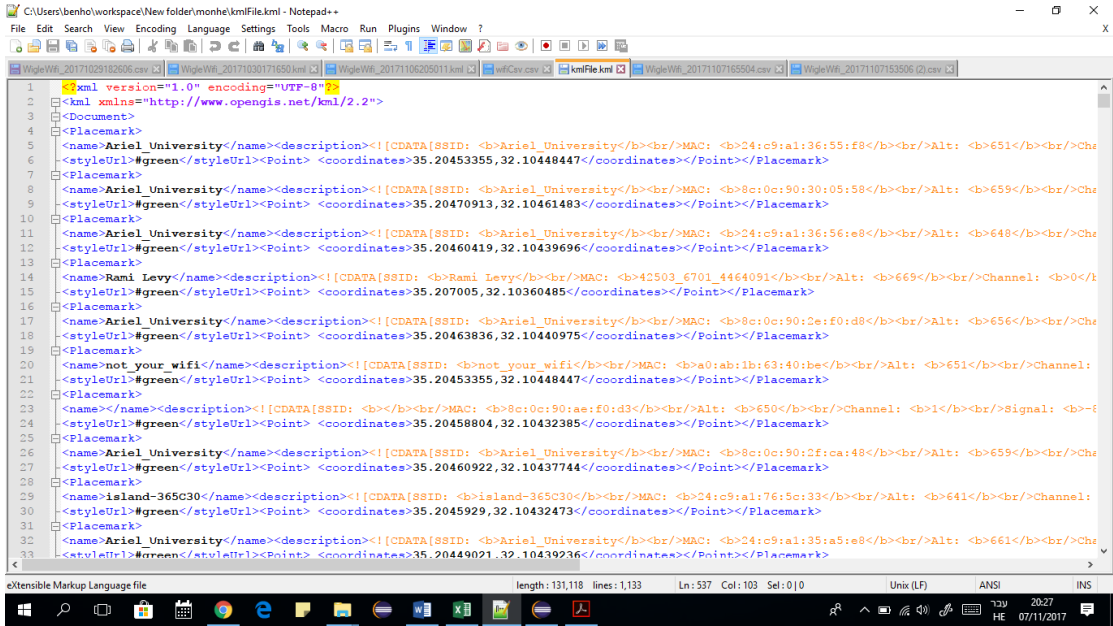


קובץ CSV שנוצר בעקבות הקריאה לפונקציה "q2.toCSV"



V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	h
na14	Frequency4	MAC4	SSID4	Signal4	Frequency3	MAC3	SSID3	Signal3	Frequency2	MAC2	SSID2	Signal2	Frequency1	MAC1	SSID1	#WiFi	Alt	Lon	Lat	ID	Time	1
-77	0	42503_67	Rami Levy	-76	6	24:c9:a1:3	Ariel_Univ	-62	11	8c:0c:90:3	Ariel_Univ	-54	11	24:c9:a1:3	Ariel_Univ	10	651	35.2045	32.1044	ONEPLUS	30/10/2017 12:06	2
												-84	12	24:c9:a1:3	Ariel_Univ	1	651	35.2045	32.1045	ONEPLUS	30/10/2017 12:06	3
												-79	0	42503_67	Rami Levy	1	651	35.2045	32.1045	ONEPLUS	30/10/2017 12:07	4
												-85	6	24:c9:a1:3	Ariel_Univ	1	651	35.2045	32.1045	ONEPLUS	30/10/2017 12:07	5
												-73	0	42503_67	Rami Levy	1	652	35.2045	32.1045	ONEPLUS	30/10/2017 12:07	6
												-74	1	8c:0c:90:ae:f0:d3		2	652	35.2045	32.1045	ONEPLUS	30/10/2017 12:07	7
												-85	6	24:c9:a1:3	Ariel_Univ	2	652	35.2045	32.1045	ONEPLUS	30/10/2017 12:07	8
												-56	11	8c:0c:90:3	Ariel_Univ	1	652	35.2045	32.1044	ONEPLUS	30/10/2017 12:09	9
												-93	11	24:c9:a1:3	Ariel_Univ	1	652	35.2045	32.1044	ONEPLUS	30/10/2017 12:09	10
												-73	1	8c:0c:90:2	Ariel_Univ	1	652	35.2045	32.1044	ONEPLUS	30/10/2017 12:10	11
												-82	11	24:c9:a1:3	Ariel_Univ	3	659	35.2046	32.1044	ONEPLUS	30/10/2017 23:42	12
												-82	1	8c:0c:90:2	Ariel_Univ	1	656	35.2046	32.1044	ONEPLUS	30/10/2017 23:42	13
												-86	2	ab:1b:63:40:be	not_your_wifi	1	666	35.2046	32.1044	ONEPLUS	30/10/2017 23:42	14
												-84	6	24:c9:a1:3	Ariel_Univ	2	640	35.2046	32.1044	ONEPLUS	01/11/2017 14:25	15
												-85	12	24:c9:a1:3	island-365c30	1	641	35.2046	32.1043	ONEPLUS	01/11/2017 14:24	16
												-71	0	42503_67	Rami Levy	1	640	35.2046	32.1044	ONEPLUS	01/11/2017 14:24	17
												-84	6	24:c9:a1:3	Ariel_Univ	2	640	35.2046	32.1044	ONEPLUS	01/11/2017 14:25	18
												-80	12	24:c9:a1:3	island-365c30	1	640	35.2046	32.1044	ONEPLUS	01/11/2017 14:25	19
												-76	6	24:c9:a1:3	Ariel_Univ	1	640	35.2046	32.1044	ONEPLUS	01/11/2017 14:26	20
												-80	1	8c:0c:90:2	Ariel_Univ	1	642	35.2045	32.1043	ONEPLUS	01/11/2017 14:27	21
												-83	6	24:c9:a1:3	Ariel_Univ	1	648	35.2046	32.1044	ONEPLUS	01/11/2017 14:27	22
												-83	1	8c:0c:90:ae:f0:d3		1	647	35.2046	32.1044	ONEPLUS	01/11/2017 14:29	23
												-59	6	24:c9:a1:3	Ariel_Univ	1	638	35.2046	32.1044	ONEPLUS	01/11/2017 14:32	24
												-59	11	8c:0c:90:3	Ariel_Univ	2	638	35.2046	32.1044	ONEPLUS	01/11/2017 14:32	25
												-72	1	8c:0c:90:ae:f0:d3		3	650	35.2045	32.1044	ONEPLUS	01/11/2017 14:32	26
												-71	0	42503_67	Rami Levy	5	651	35.2045	32.1045	ONEPLUS	01/11/2017 14:32	27
												-62	6	24:c9:a1:3	Ariel_Univ	4	652	35.2046	32.1046	ONEPLUS	01/11/2017 14:32	28

(ממכשירים אחרים)



פתיחת קובץ KML בgoogle earth.

ניתן לראות את נקודות ה-WiFi ברחבי האוניברסיטה ואת המידע עליהן.

