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-

- 1) מחלקות MAC ,Weight ,q3 ,q2 ,Wifi ,main).
 - . 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, DL, LAT ,LON ,ALT ,SSID ,MAC.

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.
- שומרים את המידע במבנה נתונים, וזאת לשם כתיבתו לקובץ חדש בפורמט הנדרש.

הפונקציות:

- 1) פונקציה פרטית סטטית (String folderPath) פונקציה פרטית סטטית (מונקציה לדייה פרטית סטטית (מחזירה מערך של קבצים במידה את מיקום התיקייה שבא אמורים להימצא קבצי הימצא (מונקצים במידה מערך של קבצים במידה ו-mull 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 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 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 (7 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".

<u>מחלקת q3</u>

המחלקה הזו גם בנויה כולה מפונקציות.

במחלקה זו-

- 1) סינון המידע שבקובץ ה csv- בוחרים לפי איזה נתון (זמן/מקום/מזהה) רוצים לסנן ומהו המידע שרוצים למצוא בנתון זה.
- 2) ארגון המידע-מוצאים את כל הנקודות בעלות אותו MAC, משאירים נקודה אחת עם הסיגנל הכי חזק, ולה מצמידים מרכז כובד משוקלל.
 - 3) כתיבת המידע לקובץ kml.
 - 4) אלגוריתם המעריך את מיקום המשתמש.

הפונקציות:

- 1) פונקציה פרטית סטטית "csvtoList"- הפונקציה מקבלת את כתובת קובץ הvsv, הפרמטר לסינון והמידע הנדרש ממנו. הפונקציה מחזירה רשימה מקושרת עם נקודות Wifi שעונות לדרישות הסנן. (בנוסף אליה, יש פונקציה ציבורית סטטית "list", שקוראת לה ובכך מאפשרת לנו לערוך בדיקות על התוצר שלה)
 - 2) פונקציה בוליאנית "fit". הפונקציה מיישמת את חתימת הפונקציה שבממשק filter. מפונקציה בוליאנית "fit מידע על נקודת wifi, הפונקציה היא למעשה פונקציית עזר למחלקה הקודמת. היא מקבלת מידע על נקודת הפרתטר לסינון והמידע הנדרש ממנו. מחזירה אמת אם הנקודה עונה לבקשת הסנן, ושקר אחרת.

- wifi פונקציה פרטית סטטית "organize"-הפונקציה מקבלת רשימה מקושרת של נקודות ולקדות מוקציה מחזירה רשימה חדשה שבה ובודקת אם קיימות בה נקודות בעלות אותו MAC. הפונקציה מחזירה רשימה חדשה שבה לכל נקודה MAC יחידאי ולכל נקודה מוצמד מרכז כובד משוקלל. הפונקציה גם יוצרת קובץ כsv שבו נמצאים נתוני הרשימה. (בנוסף אליה, יש פונקציה ציבורית סטטית "csv "organize", שקוראת לפונקציה "organize", ומאפשרת לבדוק את התוצר שלה)
- 4) פונקציה פרטית סטטית "findPlace1"-פונקציית עזר למחלקה הקודמת, היא מקבלת כתובת של קובץ הרטית סטטית "mac ומחזירה טיפוס ממחלקה Csv המאוחד וכתובת השל קובץ הרטים ממחלקה (בנוסף אליה, יש פונקציה ציבורית סטטית "findPlaceAlgorithm1" שקוראת לה, ומאפשרת לנו לבדוק את הפלט שלה).
- קובץ קובץ GPS מאוחד עם נתוני GPS, קובץ קובץ GPS, כתובת אליה תכתוב קובץ חדש שבו יוצמד לכל שורת מידע מיקום משוערך. GPS, כתובת אליה תכתוב קובץ חדש שבו יוצמד לכל שורת מידע מיקום משוערך. כא נתוני GPS הפונקציה קוראת את שני הקבצים, מחשבת מיקום משוערך, וכותבת את המידע לקובץ csv חדש. (בנוסף אליה, יש פונקציה ציבורית סטטית "findPlaceAlgorithm2" שקוראת לה, ומאפשרת לנו לבדוק את הפלט שלה).
- וכתובת של קובץ WIFI פונקציה פרטית סטטית "kml" המקבלת רשימה מקושרת של נקודות WIFI וכתובת של קובץ 6 KML. הפונקציה מחזירה אמת אם הצליחה לייצא קובץ
- 7) פונקציה ציבורית סטטית "CSVtoKML" המקבלת כתובת של קובץ CSV, כתובת של קובץ 7KML, פרמטר לסינון ,המידע הנדרש ממנו וכתובת של קובץ מרכז כובד משוקלל. הפונקציה מחזירה אמת אם יוצא קובץ KML, ושקר אחרת.

<u>מחלקות Weight, WaC,</u> המחלקה MAC מכילה מידע על mac ומרכז כובד משוקלל. המחלקה Weight מכילה מידע על משקל (דמיון בנקודה מסוימת) ומיקום. שתי המחלקות נבנו כדי לעזור באחסון נתונים לשם מימוש אלגוריתמים 1 ו-2.

<u>מחלקת ()Main-</u> 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.

מחלקות בדיקות

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, יש להשתמש בתיקיות הבאות:

[&]quot;testFolder1"- תיקייה ריקה.

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

[&]quot;testFolder3"-תיקייה של קבצי csv עם נתוני wifi (קבצי הcsv") משל מנדוק של מנדוק שקובץ"). המרצים כדי שנבדוק שקובץ המאוחד תואם לשלנו).

[&]quot;BM1_wifiscans","BM1_wifiscans","BM1_wifiscans". תיקיות של קבצי csv עם נתוני "BM1_wifiscans". שם נתוני wifi נוספים שהועלו ע"י המרצים לצורך בדיקות.

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.

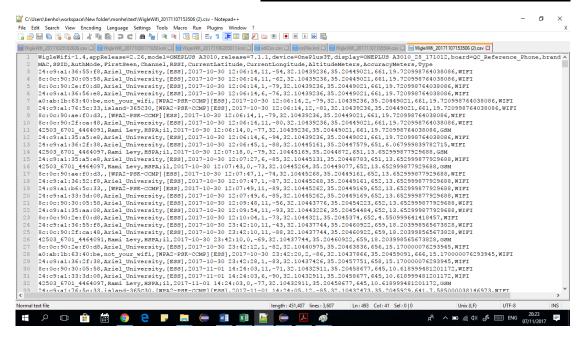
יבוא סיפריות JAK-java API kml המשמשת ליצירת קובץ kml לפי חוקי הפורמט. קובץ הlam מייצג נקודות על מפה, במקרה שלנו את נקודות הwifi. על מנת לייצא קובץ kml מרובה נקודות, בחבילת נקודות על מפה, במקרה שלנו את נקודות הitml, על מנת לייצא קובץ placemark , ולייצג גם יותר מנקודה jaka נכללת גם סיפריית placemark, בה 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.

הניסוי

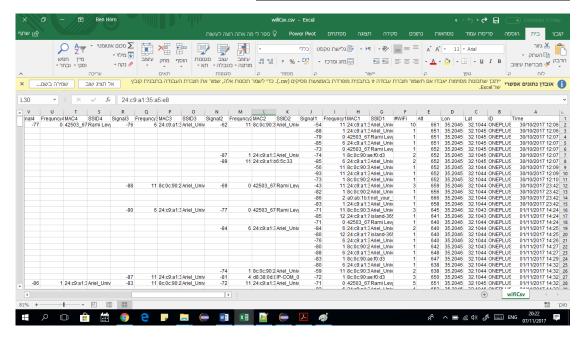
בניסוי הדלקנו את אפליקציית קליטת נתוני הWifi במכשיר הטלפון למשך יום לימודים שלם- ממעונות הסטודנטים עד לכיתות הלימוד וחזרה. לאחר מכן ייצאנו לקובץ CSV והפעלנו את התוכנה. כך נראה הייצוא לקובץ ID, OPPO3T על גבי מפה (לאחר הכנסת פילטר החיפוש: ID, OPPO3T - הID של המכשיר בו השתמשנו).

נספחי הניסוי

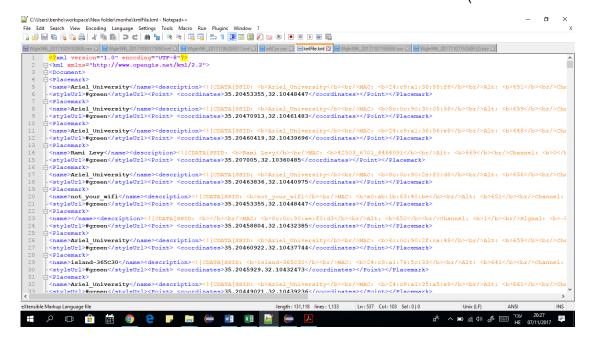
הקובץ שיובא מאפליקציית wigGLE שבמכשיר הפלאפון.



"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.

