## דו"ח מטלה מספר 0

## מבוא לתכנות מונחה עצמים

**מגישות:**

* אינה גריגור ת.ז 320442643
* דנה ברבן ת.ז 308158468

**הסבר קצר מה המערכת עושה**

1. כתיבת הקובץ הCSV המסודר:

הפונקציה readFromFilePath מקבלת נתיב של תיקיה ממנה נרצה לקרוא את הנתונים, הפונקציה writeToFilePath נותנת נתיב בו צריך ליצור את הקובץ CSV המסודר עם כל הנתונים.

הפונקציה readFromFilePath לוקחת את כל הקבצים בתיקיה ומבצעת סינון של סוגי הקבצים: אם הקובץ הינו קובץ CSV היא משתמשת בו ואם לא היא מתעלמת.

לאחר מכן, אנחנו עוברים בלופ על כל קבצי ה-CSV ומפעילים את הפונקציה readFromFileAndGroup על כל קובץ בנפרד. פונקציה זו קוראת את כל נקודות ה-wifi ומוסיפה אותם ל-ArrayList.

לאחר מכן, השתמשנו במבנה נתונים מסוג map ששומר key ו-value והכנסנו לתוכו את הנתונים לאחר שביצענו groupingBy (בעזרת ייבוא java.util.stream.Collectors) לפי השדות: Time, Lat, Lon, Alt.

בעצם, ה-map מחזיק כרגע את כל הנתונים הממויינים, מכווצים לפי קבוצות ומסודרים.

לאחר מכן, בפונקציה writreToFile אנחנו קוראים את ה-map ורושמים את הנתונים בקובץ CSV שפתחנו בעזרת הפונקציה writeToFilePath.

אם הרשימה של כל קבוצה מכילה יותר מ-10 רשתות, הפונקציה writreToFile מבצעת גם סינון של עשרת הרשתות החזקות ביותר ומדפיסה רק אותן (collections.sort(wifiList)).

1. קריאת המידע מקובץ הCSV המתוקן וסינון:

לאחר שקיבלנו ויצרנו את הקובץ הCSV שרצינו לפי הפורמט הרלוונטי – כעט צריך לקרוא ממנו ולסנן לפי שדות רלוונטיים כדי ליצור את הקובץ הKML שאנו רוצים.

עשינו זאת על ידי הקלאס ReadToKml שהוא מקבל את הקובץ הCSV שיצרנו בסעיף הקודם וקורא אותו.

הוא קורא אותו בפונקציה filterByParameter.  
כלומר הוא מסנן בעזרת הפונקציה הזאת קודם לפי מה המשתמש מכניס (לפי מיקום, לפי זמן או ID), מקבל את המידע ועל ידי SWITCH שולח אותו לפונקצית private ששם היא מסננת לפי המידע שהמשתמש רוצה ומחזירה רשימה רלוונטית.  
השתמשנו גם במבנה נתונים MAP שוב כדי להשוות את הכתובות בMAC כדי לראות אם אין אותה נקודה יותר מפעם אחת – ואם יש אז שתחזיר את הנקודה עם העוצמה החזקה יותר (signal).

נעזרנו גם במחלקה WifiPoint ששם יש קונסטרקטור וcompareTo (על מנת לבדוק עוצמת הsignal ) וגם setSharedInfo שאלו השדות המשותפים.

1. כתיבת הKML:

על מנת לכתוב את הKML נעזרנו בסיפריות חיצוניות שמצורפות לפרוייקט ואיתן יצרנו את הקובץ KML אחרי הסינון מהקלאס ReadToKml – משם בעצם מקבלים את רשימה המסוננת כבר.

רצנו בלולאת for כדי להדפיס כל נקודה והקובץ KML נוצר בPATH שבו הגדרנו.

בחרנו לעבוד עם external Jar libraries for KML מכיוון שהסיפריות האלו יוצרות את הנקודות בפורמט שדרוש לKML באופן מובנה וברור יותר מאשר לעשות ידני.

**כלי תוכנה בהם השתמשנו:**

* Eclipse
* WIGLE-Wi-Fi
* external Jar libraries for KML
* Google earth

**הסבר על הניסוי שביצענו:**

תחילה ערכנו סריקה ביישוב ברקן ממכשיר LG. לאחר מספר ימים ניסינו לבצע סריקה באוניברסיטה ללא הצלחה - המכשיר לא קלט את אות ה-GPS, אז ביצענו סריקה ממכשיר onePlus של סטודנט מהכיתה.

מצורפים לפרוייקט KML לדוגמא וקבצי CSV לדוגמא שהפרוייקט שלנו יצר.