

## מבוא לתכנות מונחה עצמים – מטלה 4

במשך הסמסטר בנינו תוכנית Java הקוראת/מייצאת/שומרת נתונים על רשתות אלחוטיות. תחילה כתבנו קוד הקורא נתונים מקבצי CSV שקיבלנו מהאפליקצייה Wiggle-wifi אשר מופיעים בהם נתונים על רשתות האינטרנט בסביבה(נתוני מיקום, כתובת ראוטר, עוצמת חיבור וכו'), התוכנית שלנו קוראת את הנתונים ושומרת אותם בתוך אובייקטים שבנינו.

בעזרת התוכנית ניתן לייצא את הנתונים לקבצי CSV(בפורמט המבוקש במטלה), וקבצי KML(צפייה ברשתות על גבי google earth).

ניתן לסנן את המידע שקראנו על ידי מסננים שבנינו(סינון לפי: זמן, מיקום, שם משתמש). ובנוסף ניתן לחשב בעזרת התוכנה מיקום של משתמש, לפי הנתונים הנמצאים במאגר המידע שקראנו.

במטלה 3 בנינו ממשק גרפי לשימוש בקוד(בעזרת [Windows builder](#)):

במטלה 4 הוספנו אפשרות לקריאת נתונים מבסיס מידע(MySql).

The screenshot shows the 'wifi scanner system' application window. It features a sidebar with file management options: 'Choose folder', 'Add csv file', 'Clean data', 'Save to CSV file', and 'Save to KML file'. The main area is divided into several functional sections. On the left, there are input fields for 'Lines' and 'Networks'. Below these are buttons for 'Submit filter', 'Unfilter', 'Save filter', and 'load filter'. A section for filter management includes 'Current filter:', 'Imported filter:', and buttons for 'And filters', 'Or filters', and 'The current filter without the imported'. At the bottom left, there are input fields for 'IP:', 'USER:', 'PASSWORD:', 'PORT:', 'DATABASE:', and 'TABLE:', followed by a 'GET DATA FROM SQL' button. The right side of the window contains configuration for two algorithms. 'Algorithm 1' includes an 'Enter MAC' field and a 'find location' button. 'Algorithm 2' includes input fields for 'mac 1', 'mac 2', 'mac 3', 'sig 1', 'sig 2', and 'sig 3', an 'Enter string:' section, and another 'Find location' button.

## איך להריץ:

את התוכנה ניתן למצוא [כאן](#), ולאחר מכן לייבא אותה לעclipse.

## ייבוא/ייצוא קבצים:

בחירת תיקייה – בחירת תיקייה המכילה קבצי CSV מהאפליקציה (מצורפת התיקייה input files)  
הוספת קובץ CSV – בחירת קובץ בפורמט המטלה (46 עמודות) והוספת הנתונים למאגר המידע של התוכנית.  
ניקוי – מחיקת הנתונים מהמאגר.  
שמירת קבצים – שמירת קובץ CSV (46 עמודות) וקובץ KML (עבור גוגל ארث').

לאחר לחיצה במערכת על כפתור שעלול לשנות את מאגר הנתונים שלה, המערכת תחשב מחדש את מספר השורות והרשתות במאגר המידע.  
בנוסף לאחר בחירת תיקייה, המערכת מקצה "חוט" אשר "מאזין" לתיקייה, כאשר יבוצע שינוי המערכת תבדוק את הקבצים מחדש, לבדיקת השינויים.

## סינון מידע:

את הנתונים במאגר המידע ניתן לסנן לפי מספר פרמטרים (זמן, שם משתמש ומיקום), לאחר ביצוע הפילטר ניתן לשמור אותו כקובץ במחשב, ולאחר מכן לטעון אותו ולהשתמש בו יחד עם הפילטר שיוזן במערכת ("או"/"גם"/"ללא").

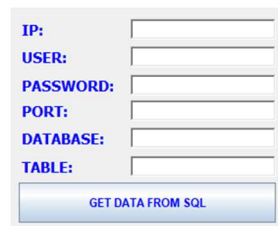
## אלגוריתמים:

במטלה 2 בנינו שני אלגוריתמים לחישוב מיקום של משתמש, על פי מאגר הנתונים שברשותנו.

אלגוריתם 1- הזנת כתובת MAC, וקבלת מיקום.  
אלגוריתם 2- הזנת זוגות של כתובות MAC ועוצמת קליטת, או הזנת שורה מקובץ 46 עמודות, וקבלת מיקום.

## בסיס מידע:

במטלה 4, הרחבנו את האפשרות לקריאת נתונים גם לקריאת מידע מטבלה בmysql.



בעזרת המערכת התחברנו לשרת שניתן במטלה, קראנו את השורות והכנסנו את הנתונים אל תוך האובייקטים של המערכת, ולאחר מכן צירפנו את הנתונים למאגר המידע שלנו.

## חבילות ומחלקות:

**MyObjects** - החבילה מכילה את האובייקטים הבאים:

1. Checks - אובייקט המייצג שורה מתוך הקובץ CSV של 46 השורות.
2. Wifi - אובייקט המייצג נתונים של רשת אלחוטית.
3. WifiSort – אובייקט שנבנה עבור Checks, המכיל רק 4 נתונים על הרשת (כתובת, עוצמה, ערוץ, שם).

**MainPackage** - מכילה את המחלקות הבאות:

1. MatrixSortByMAC - מיון מבני נתונים לפי כתובת MAC.
2. ReadCsvMatalaFormar - קריאת קובץ CSV של המטלה.
3. WifiScanner - קריאת קובץ CSV של האפליקציה.

**ExportFiles** - ייצוא קבצי CSV ו kml.

**LinkedListFilters** – מחלקות לסינון מידע ברשימה מקושרת (מבנה הנתונים של המערכת).

**Algorithms** - החבילה מכילה את שני האלגוריתמים, ובדיקות.

**GUI** - הממשק הגרפי של המערכת.

**Database** – חיבור ל mySql.