

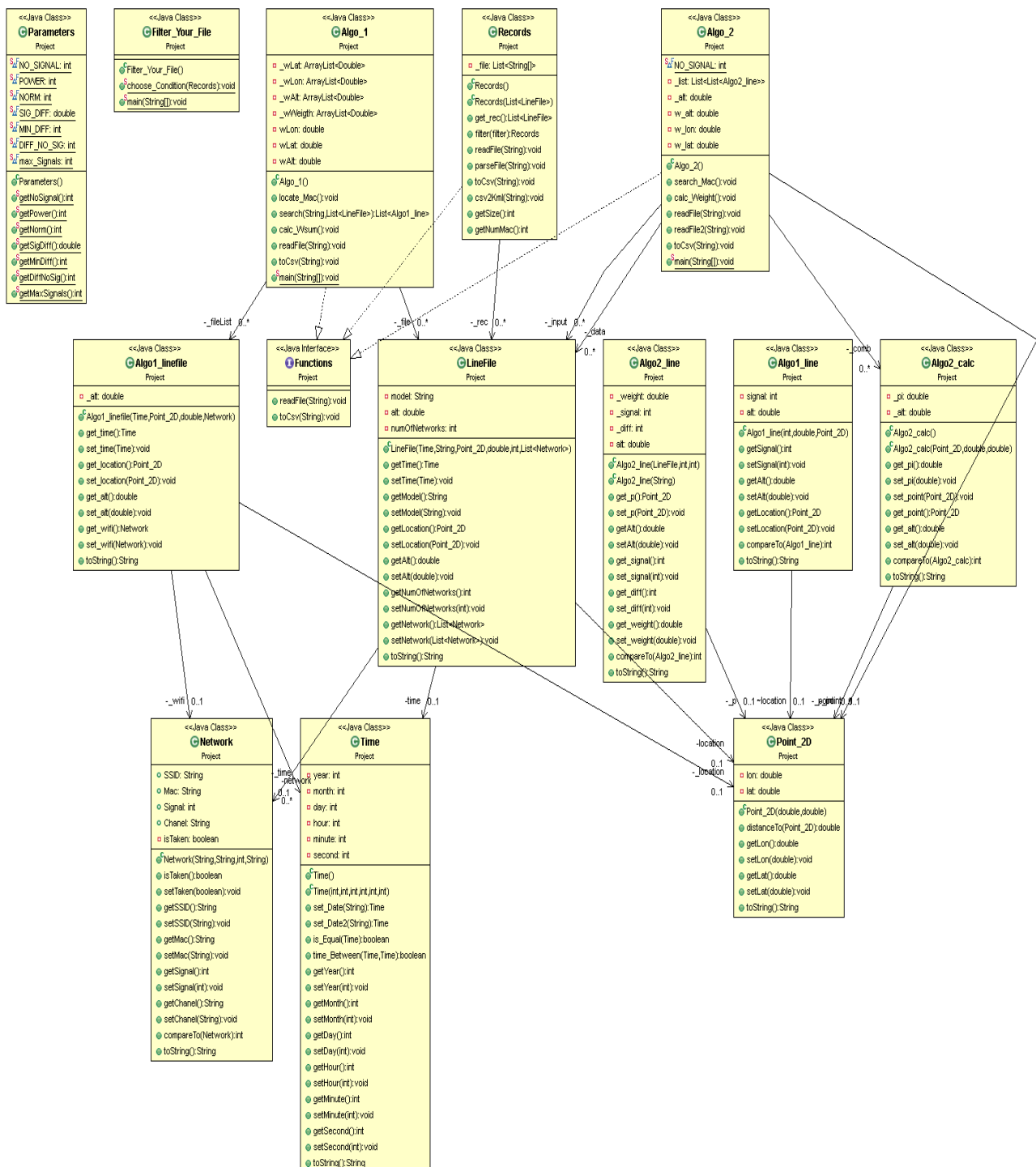
תכנות מונחה עצמים – סיכום הפרויקט

במטלה זאת פיתחנו מערכת מורכבת שמאפשרת איסוף מידע גיאוגרפי הפקה של תובנות ממידע זה והצגת המידע בכלים גרפיים.

המטלה מכילה כמה פרויקטים אשר מבצעים פעולות שונות המתחברות ביחד לתוצר הסופי.

Project package

החבילה הזו מכילה כמה קלאסים אשר קולטות קבצי csv אשר נקלטו מאפליקצית wileWifi, מאחדות אותם לקובץ המאחד ראטרים אשר נקלטו באותה נקודת זמן ומאפשר לייצא את מבנה הנתונים המאוחד לקובץ csv וkml.



הקלאס המרכזית בחבילה הזו היא Records. קלאס זה יוצרת רשימה מסוג LINEFILE אשר תכיל את התוצר הסופי – הקובץ המאוחד. בכדי ליצור את הרקורד נפעיל את פונקציית parseFile- פונקציה זו מקבלת נתיב לתיקיה אשר מכילה את קבצי ה csv אשר נקלטו באפליקציה, היא קוראת כל קובץ ומכניסה אותו לרשימה בצורה מסודרת כפי שהתבקשו במטלה. לכל נקודת זמן אשר נקלטה בסריקה נייצר שורה מסוג LineFile- שורה זו מכילה 5 מרכיבים זהים אשר יש לכל הראוטרם באותה נק' זמן – שעה, ID, LAT, ALT, LON ועמודה עם מספר הראוטרם אשר נקלטו בדגימת הזמן הנתונה. לאחר מכן יש עוד 10 עמודות המכילות נתונים על הראוטרם אשר נקלטו הממוינים לפי עוצמת הקליטה שלהם. את הרשימה שהתקבלה נוכל לפלטר ולייצא לקובץ csv kml.

על מנת לפלטר את הרשימה נלך לקלאס Filter_Your_File שהיא למעשה הקלאס המרכזית של הפרויקט ודרכה מריצים אותו (לפני הGUI).

```
public static void main(String[] args) {
    Records r = new Records();
    r.parseFile("C:\\Users\\noyvtvili\\git\\00_Project");
    r.toCsv("Merge_File.csv");
    r.csv2Kml("Merge_File_kml.kml");
    Scanner s = new Scanner(System.in);
    System.out.println("Would you like to filter your merge_file? press 1");
    int c = s.nextInt();
    if(c==1){
        choose_Condition(r);
    }
    else{
        System.out.println("Good-Bye");
    }
    s.close();
}
```

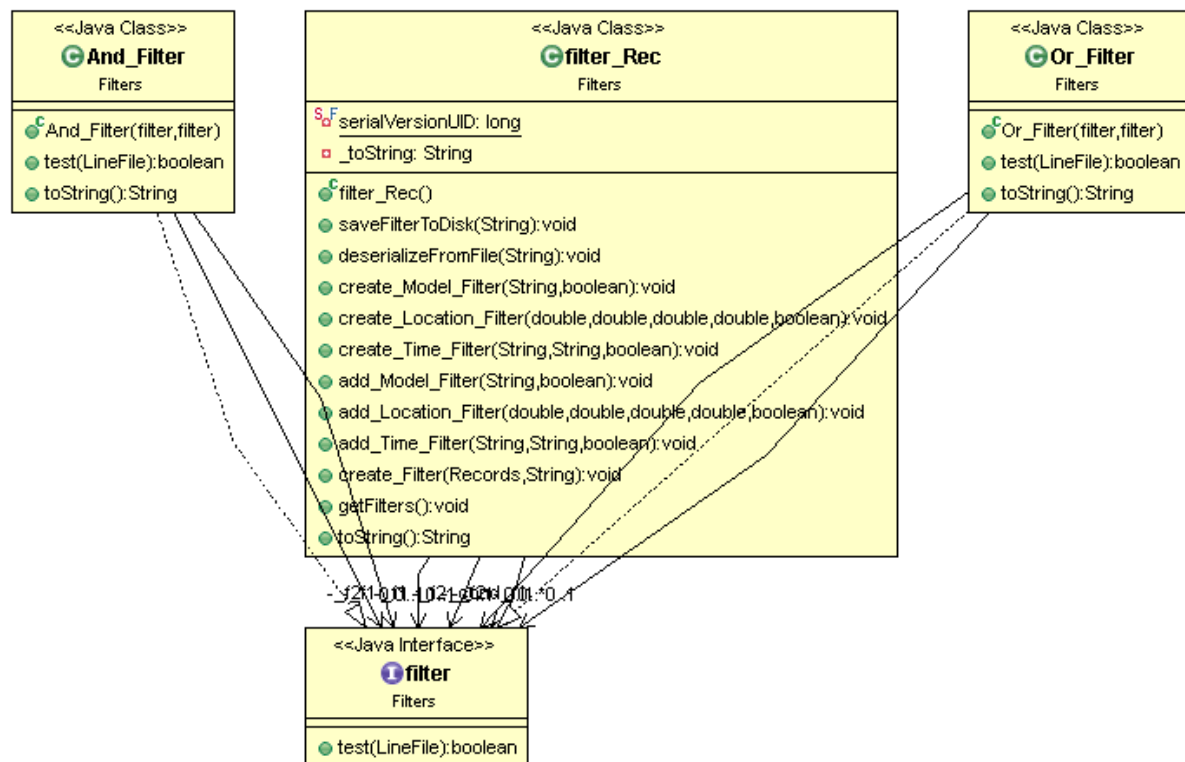
נכניס את הנתיב שבו נמצאים קבצי האפליקציה שסרקנו, נבחר את שמות הקבצים החדשים שנרצה לייצר ונריץ. לאחר שנוצרו הקבצים, נפתחת אופציה לפלטר את הקובץ המאוחד. אם לוחצים על 1 נוכל לבחור 3 דרכים לפלטר את הרשימה: time, model, location. כל בחירה תגיד למשתמש איזה נתונים להכניס ובסוף תייצר קובץ מפולטר לפי הדרישה.

על מנת לייצר אופציה של פילטור הקובץ, ייצרנו את **Filters package**: חבילה זו מכילה ממשק filter המחזיר משתנה בוליאני לתנאי של פילטור הקובץ.

כאשר נקרא למשתנה מסוג filter, הוא יעבור על הרשימה ויבדוק אם התנאי שהכנסנו מתקיים, אם כן יכניס אותו לרשימה המפולטרת.

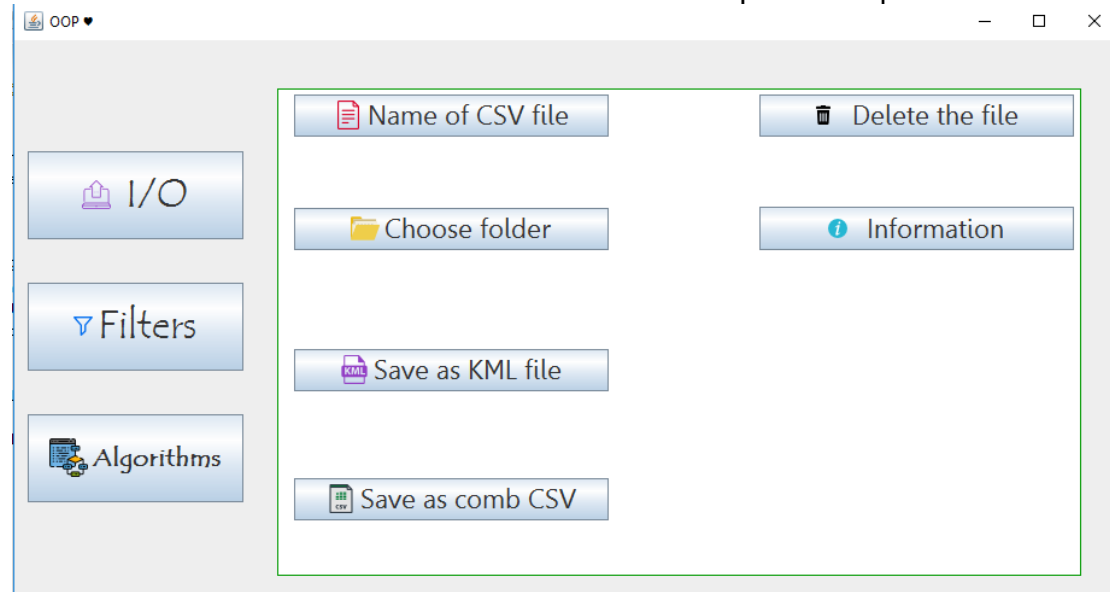
- עבור ה gui במטלה 3 נתבקשו להוסיף גם אופציה לפלטר רשימה ביותר מפילטר אחד ולכן הוספנו 2 קלאסים המאפשרות לפלטר רשימה לפי && או ||. בנוסף הוספנו קלאס שנקראת filter_Rec. קלאס זה מכילה רשימה של תנאים אשר המשתמש ביקש לפלטר לפיהם ומפלטרת את הרשימה בהתאם לשער הלוגי ולתנאים שהוכנסו.

בר זמסקי- 312534753
דוריה שפילמן- 313466625
נוי טוילי- 308426790



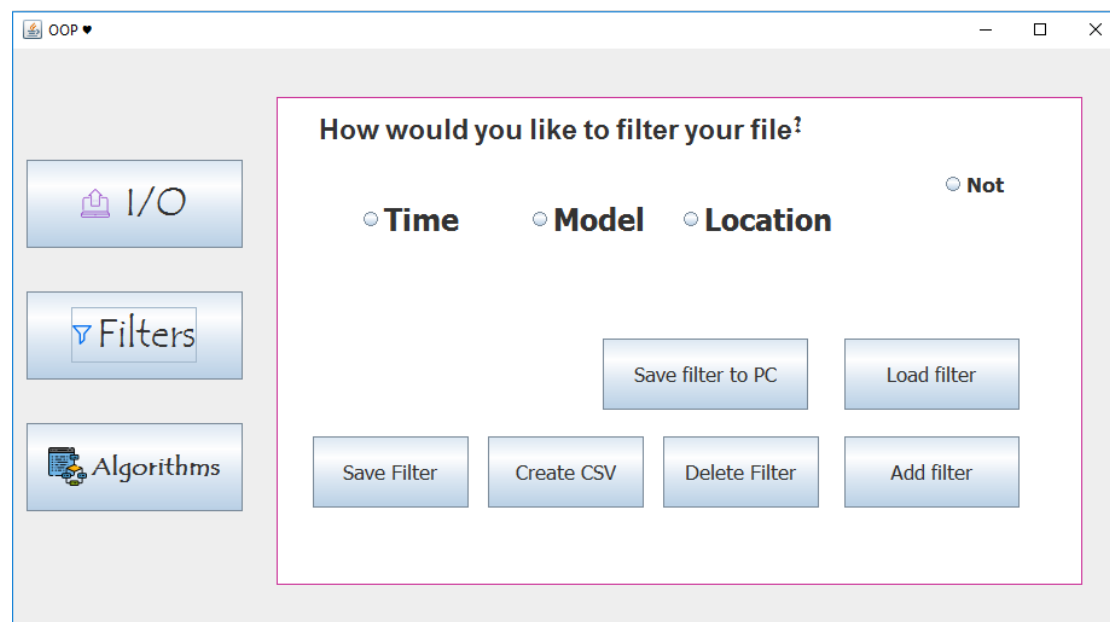
הרצת הפרויקט דרך GUI package:

נפעיל את התוכנה ונקבל את המסך הראשי :



יש להכניס את שם הקובץ אשר נרצה לייצר ולבחור את התיקייה אשר מכילה את הקבצים שנרצה לקרוא. לאחר מכן נוכל לשמור את הקובץ הראשוני (לפני פילטור) ל csv ולkml .

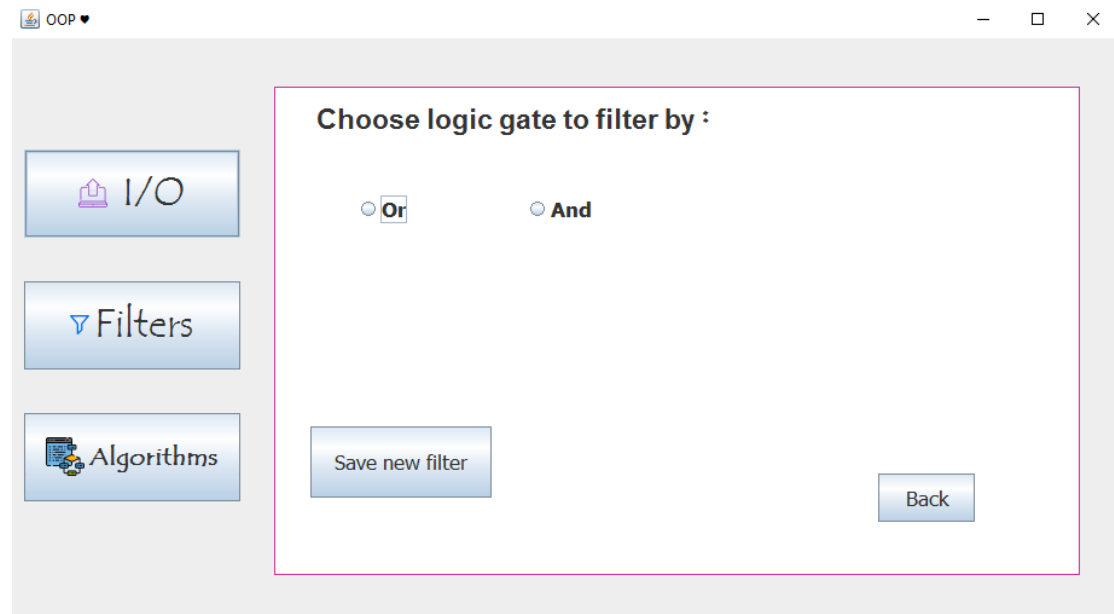
לאחר מכן נעבור ללשונית Filters -



נבחר לפי מה נרצה לפלטר את הרשימה המאוחדת שיצרנו. אם נרצה שלילה נלחץ על not. לאחר הכנסת המידע לפי שהתבקשתם יש ללחוץ על כפתור save filter. אם נרצה ליצר קובץ עם פילטר אחד תוכלו ללחוץ ישר על create CSV ולראות את הקובץ בתיקייה.

בר זמסקי- 312534753
דוריה שפילמן- 313466625
נוי טוילי- 308426790

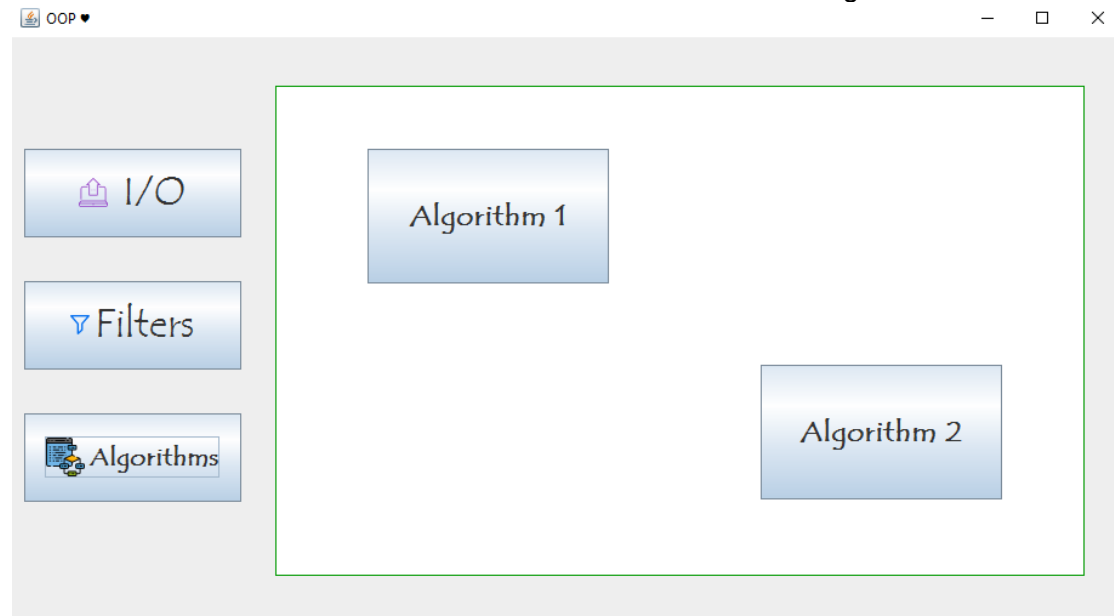
אם תרצו להוסיף פילטר נוסף, יש ללחוץ על לשונית Add filter-



יש לבחור את השער הלוגי שלפיו תרצו לפלטר בין שני התנאים. לאחר בחירת השער יופיעו שוב שלושת האופציות שניתן לפלטר איתן את הרשימה. לאחר הבחירה יש ללחוץ על save new filter והקובץ המפולטר + kml שלו יופיעו בתיקייה.

בר זמסקי- 312534753
דוריה שפילמן- 313466625
נוי טוילי- 308426790

במטלה 2 ייצרנו 2 אלגוריתמים למציאת מרכז כובד משוקלל .
ניכנס ללשונית Algorithms :



בלחיצה על אלגוריתם 1 תתבקשו להכניס את שם הקובץ המאוחד (שנוצר מהתוכנית שלנו) ותקבלו את הקובץ המתאים לאלגוריתם ה-1.

בלחיצה על אלגוריתם 2 יש להכניס שני שמות של קבצים : אחד המכיל שורה אחת של הקובץ המאוחד ללא מיקום והשני המכיל את הקובץ המאוחד המתאים לקובץ הראשון. בלחיצה על אוקי יתווסף לתיקיה קובץ סופי עבור האלגוריתם השני.

בר זמסקי- 312534753
דוריה שפילמן- 313466625
נוי טוילי- 308426790

UML DIAGRAM FOR THE WHOLE PROJECT

