

מבוא לתכנות מונחה עצמים – מטלות קורס

מסמך זה מפרט את מכלול המטלות של קורס מבוא לתכנות מונחה עצמים, הרעיון המרכזי במטלות הקורס שהן למעשה מטלה אחת "מתגלגלת" כך שאתם למעשה תתבקשו כל הזמן לשפר ולהרחיב את המטלות הקודמות שלכם כדי לאפשר למידה מעמיקה ומתמשכת.

הנחיות כלליות:

1. את המטלה עושים בזוגות, יש להגיש את כל המטלות בזמן! לפי הנחיות, על כל איחור לא מוצדק תהיה הורדת ניקוד.
2. המטלות תיבדקנה באמת במהלך התרגולים, על כל אחד מבני הזוג להבין באופן מלא ושותף אל כל רכיבי המטלה בפרט כיצד להריץ לבדוק ולהכיר כל שורה בקוד.
3. המטלות תיבדקנה באופן אוטומטי באספקטים של "העתקות קוד" אין לבצע שום העתקה של קודים בין קבוצות שונות, מותר לעשות שימוש בקוד פתוח, אבל חובה לציין זאת בפירוש ולהביא את המקור המדויק. למען הסר ספק: שימוש בקוד פתוח (או כל קוד זמין ברשת) שלא יצוין מקור הקוד יחשב כהעתקה!
4. כלל הפיתוח יעשה בכלי בקרת התצורה של github, הכירו היטב את הכלי ועשו בו שימוש משמעותי ומעמיק, הן לקוד והן לתיעוד מסודר של הפרויקט שלכם.

המטלה עצמה:

במטלה זאת נפתח (בהדרגה) מערכת מורכבת שמאפשרת איסוף מידע גיאוגרפי הפקה של תובנות ממידע זה והצגת המידע בכלים גרפיים.

נסתכל על אפליקציה כגון waze, היא מפיקה מידע לגבי עומסי התנועה בזמן אמת ע"י צבירת המידע מנהגים רבים, וטיוב שלו.

בדומה הסתכלו על האפליקציות הבאות:

[openSignal](#), [G-Mon](#), [OpenStreenMap](#) כולן מאפשרות איסוף מידע גיאוגרפי מגוון בשיטות בגישות שונות, המידע כולל: עוצמת קליטה של הטלפון, נתוני גלישה, מיקום, מהירות, מיפוי ומידע נוסף שהמשתמש מעלה. שימו לב שיש המון סוגים של "אפליקציות גיאוגרפיות" חלקן אוספות מידע על טיולים – נניח "עמוד ענן", אחרות על "איסוף מדדים גופניים, דופק, מהירות כו", ואפילו אפליקציות לאיסוף מידע של מחשב הרכב כגון [TORQUE](#)

באופן כללי נוכל לחלק את האפליקציות הללו לשלושה חלקים:

- אפליקציית לקוח (לרוב אפליקציית אנדרואיד) שאוספת את המידע ומעלה אותו "לשרת".
- "שרת" שמאפשר שמירה של נתוני המשתמשים, טיוב ובעיבוד שלהם.
- מערכת "תצוגה" וניהול שכוללת ממשק גרפי – לרוב בממשק של אפליקציית רשת.

מטלה 1:

במטלה זאת נארגן נשפר ונטייב את מטלה 0,

1. בשלב הראשון נבצע דוח ביקורת על שתי עבודות אחרות שנעשו: את הדוח יש לבצע באופן הבא: כל זוג יקבל שתי מטלות אחרות לבדיקה, ויעלה את חוות הדעת שלו לפי הסעיפים הבאים:

- איכות העבודה מבחינת כתיבת הדוח.
- איכות הקוד מבחינת אופן הכתיבה, התייעוד והחלוקה למחלקות
- איכות המערכת – יש להריץ ולבדוק את איכות התוצאות: כדי לבצע סעיף זה תצטרכו להתמודד עם בעיות טכניות קלות, כגון: ייבוא פרויקט לאקליפס, מציאת התוכנית הראשית, עדכון התוכנית הראשית שתעבוד עם קלטים שלכם, ועוד – עשו מאמץ להריץ את הקוד – אבל אם בסופו של דבר הקוד "לא רץ" ציינו זאת.
- דירוג השוואתי של העבודה שאתם בודקים למול העבודה שלכם + הערות טקסט כלליות.

על כל קבוצה לזהות את המטלה שלה בתיקיית רשת זו:

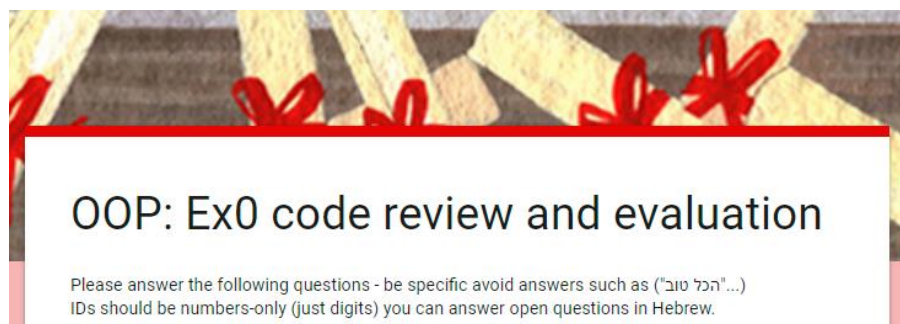
https://drive.google.com/open?id=1L_3mw0dhpGMHH-VWU8CLDqQJSail5xml

הורידו את שתי המטלות שנמצאות מעליכם ומתחתכם (הקפידו לעשות זאת כאשר רשימת המטלות ממוינת לפי שם – בסדר עולה, ראו תמונה)

📁	assignsubmission_file_matala0.7z_2957034_אבסקר שירן	👤
📁	assignsubmission_file_203037007_204326409_205746720.rar_2965239_אדרי טל	👤
📁	assignsubmission_file_321005928AND308366749.zip_2965301_אורן איזבלה גניה	👤
📁	assignsubmission_file_311233688.7z_2965324_אחוויל עדי	👤
📁	assignsubmission_file_308275601.zip_2965201_איבגי יאיר	👤
📁	assignsubmission_file_205749211.rar_2965255_אלון הדר סוליקה	👤
📁	assignsubmission_file_205484000.zip_2965278_טמיר עטור	👤

איור 1: החלק העליון של רשימת המטלות – ממוינות לפי שמות הקבצים בסדר עולה, בדוגמא זאת על "אדרי טל" לבדוק את המטלה של "אבסקר שירן" (מעל) ו"אורן איזבלה גניה" (מתחת). כיוון ש"אבסקר שירן" היא ראשונה היא תבדוק את המטלה השנייה "אדרי טל" ואת האחרונה ברשימה.

את הדוח יש למלא בטופס [הבא](#), יש לסיים להגיש את הביקורת על שתי המטלות עד יום חמישי 16.11 – כדי לאפשר לאחרים לתקן את מטלה 0 בהתאם לביקורת שלכם.



איור 2: צילום מסך של הטופס של ביצעו בדיקה של מטלה 0.

2. העבירו את הפרויקט שלכם ל github, סעיף זה דורש מכם (כמובן) להיות בעלי חשבון ב github, עדכנו את ה readme, ואת כל הקבצים הרלוונטיים (קוד, קובצי קלט, הסברים כו').
3. עדכנו את המטלה שלכם (ב github) לפי הביקורת שקיבלתם, אינכם חייבים לקבל את הביקורת אבל אם היא מוצדקת ונכונה – עשו מאמץ ושפרו את הקוד שלכם בהתאם.
4. הוסיפו לפרויקט שלכם ב github תיקייה עם קובצי קלט לדוגמא.
5. חפשו ומצאו חבילת תוכנה המאפשרת להמיר מידע מ-java לפורמט kml. הסבירו והצדיקו את הבחירה שלכם במסמך שמצורף למטלה.
6. הוסיפו לקוד שלכם רשימת בדיקות Unit Testing, הבדיקות חייבות לכלול גם בדיקות עומק של כל המכלולים.
7. שפרו את התצוגה כך שתכלול timeline ראו למשל:
<https://support.google.com/earth/answer/148094?hl=en>
בסעיף זה עליכם לשנות את פורמט הפלט של התוכנה שלכם (ה kml שהתוכנה יוצרת) כך שהקובץ יכלול את הזמן של כל מדידה, ויאפשר לכם להציג את המדידות על קו זמן.