

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". 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! שימוש בקוד של קבוצות אחרות מהקורס יעשה אך ורק באישור הקבוצה, ויש לציין זאת.
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 4: גרסה 0.1 3.12018

מטלה זאת מסכמת את מטלות הקורס (אחרונה ודי) ובה נשלים את המערכת, בפרט נרחיב את מטלה 2 בשני תחומים:

- הוספת תמיכה ב Gradle כדי לאפשר בניה של הפרויקט שלכם באופן קל ונוח, בפרט עליכם להוסיף tasks לצורך קומפילציה, תיעוד, יצירת jar, zip, ו"ניקוי קבצים זמניים" – כמובן ניתן להוסיף.
 - שימוש בבסיס נתונים לצורך קריאה של נתוני ההסריקות.
- יש לתכנן לפני שמתחילים לממש. חשוב להתחיל בתכנון כללי של מסכי התצוגה והרצת כתיבת של מסמך קצר שבו תצינו איזה מחלקות וקבצים יהיו במערכת שלכם, מה כל רכיב יעשה, ומה יהיו הקשרים ביניהם. וכמובן פירוט לגבי אופן מימוש האלגוריתמים שלכם.

חלק ראשון מסמכי מערכת מסכמים:

בחלק זה עליכם להשלים את מסמכי התכנון ותיאור המערכת (מסמכים ולא קוד) – כך שניתן יהיה להבין בקלות את מבנה המערכת וכן ניתן יהיה לקמפל ולהריץ אותה. המסמכים צריכים לכלול דיאגרמות UML, ואז מסמכים שמסבירים את מבנה המערכת, אנו ממליצים על שימוש בדיאגרמות הבאות, אם כי בהחלט אפשר להחליף את התרשימים בטקסט - לשיקולכם:

- How to compile & run
- Conceptual model
- Collaboration diagram (כולל התייחסות לתהליכים\חוסים שונים במערכת)
- Class diagram (כולל התייחסות לחלוקה לחבילות)

חלק שני: השלמות:

- הוספת תמיכה ב Gradle
- הוספת יכולת קריאה של נתונים מטבלה בסיס נתונים, בסיס הנתונים: יש לאפשר בחירה של IP, port, user, password וכן בסיס נתונים וטבלה, ואז לאפשר הוספה של סריקות ה wifi שנמצאות בטבלה לנתונים הקיימים.

הדרכה:

- a. יש להתייחס לבסיס הנתונים כמקור נוסף – משמע צריך לאפשר "הוספה" של טבלה, וכן לבדוק האם בוצע בה שינוי ואם כן לעדכן את האפליקציה (בדומה לקובץ או תיקייה).
- b. אין צורך לאפשר כתיבה לטבלה מרוחקת.
- c. דוגמא מפורטת של טבלה והשימוש בה מצורפת למטלה זאת – עשו בה שימוש.

בהצלחה!!