- תרגיל בית מס' 14 -

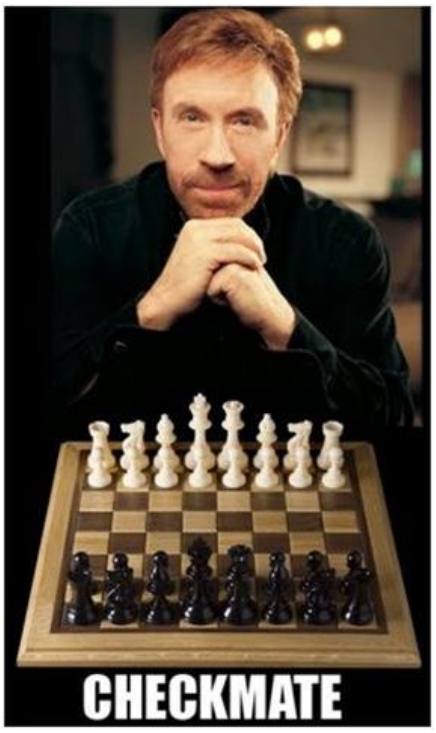
עיצוב תוכנה וקוד נקי

כללי

אז שיעור מס' 4 היה מאוד עמוס...
בשיעור ראינו שפיתוח פרויקט או מוצר - זה תהליך שמורכב מכמה חלקים ודורש המון מחשבה, תכנון והשקעה.

כחלק מהתהליך, למדנו עקרונות בתכנון ועיצוב של פרויקט תוכנה.

בתרגיל הבית הקרוב לא נכתוב פרויקט חדש, אלא ניישם את מה שלמדנו בשיעור על פרויקט שכבר עשינו – פרויקט השחמט (הזכור לטוב).



הוראות הגשה

את העבודה על התרגיל נבצע על גבי ה-Git Repository של פרויקט השחמט (שאותו ביצענו בשבועות 7-8). נגיש למדריך קישור ל-Repo.

שימו ❤

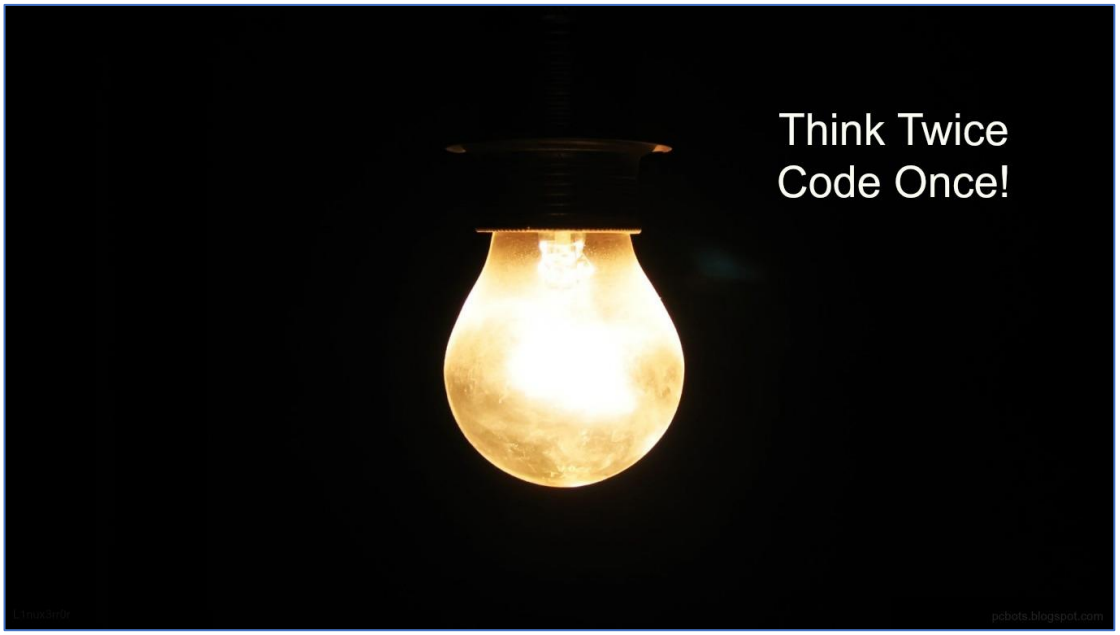
במקור פרויקט השחמט נעשה בזוגות, והתרגיל הזה נועד לעבודה עצמית.
לכן בשביל להעלות את המסמכים והתיקונים שלכם/ן, אנא פתחו branch חדש ב-repository של פרויקט השחמט עם השם שלכם/ן:
לדוגמא Lesson4B_HW_Israel_Israeli

העלו לענף החדש את כל המסמכים והתיקונים הרלבנטיים לתרגיל.

מה עושים בתרגיל?

כמו שלמדנו - לפני שקופצים למקלדת, חשוב לתכנן ולבנות את המתווה שעל פיו נממש את התוכנית. בשיעור ראינו מודלים שונים שדרכם אפשר לתאר את התוכנית שלנו – High Level Design, UML, דיאגרמות וכו'...

המשימה שלנו היא להוסיף ולְשַׁפְּצֶר[1] את המסמכים האלו על פרויקט השחמט שלנו.



מוכנים? בואו נתחיל!

[1] מונח צהל"י שמשמעותו שיפוץ ושיפור

משימה 1 - High Level Design

כפי שלמדנו, בתחילת תהליך התכנון של פרויקט – נרצה ליצור מסמך פשוט שמסתכל על הפרויקט מלמעלה.

המסמך יכלול סקירה כללית של המערכת ותיאור בסיסי של הרכיבים העיקריים. במסמך נשתמש בשפה כללית, ולא שפה מקצועית. לא נפרט טכנולוגיות ושפות תכנות שבהן נשתמש.

משימה - ניצור מסמך **High Level Design** המתאר את פרויקט השחמט ב"מבט על", ונוסיף אותו ל-repo של הפרויקט.

שימו ♥ - התרשים לא צריך להיות מפורט, אלא רק להכיל את הרכיבים המרכזיים הנמצאים בפרויקט.

זכרו לכלול את הרכיבים המרכזיים בצד שמבצע את ה"לוגיקה" והחישובים (**Backend**), וגם את הצד הפונה למשתמש (**Frontend**) ומאפשר לו לתקשר עם התכנית שלנו (אין צורך להיכנס למימוש שלו). זכרו לכלול גם את האופן שבו ה-Backend וה-Frontend מתקשרים.

המסמך אמור להיות קובץ **Word**, כמה דפים, לא ארוך מדיי.

משימה 2 – תיקון UML

אולי כבר הספקנו לשכוח, אבל לפני שהתחלנו לממש את הפרויקט – כל זוג שלח למדריך/ה שלו את ה-Design של הפרויקט בצורה של UML.

עכשיו, אחרי שלמדנו על עקרונות לכתיבת קוד נקי (Clean Code) כגון KISS, DRY ו-SOLID - נוכל לתקן את ה-UML ההוא שכתבתנו, ונוסיף גרסה מתוקנת ו-'נקייה' שלו.

משימה - ניצור UML מעודכן (ומתוקן) ונוסיף ל-repo של הפרויקט.

שימו ♥ - ציינו את ההפרות שהיו ב-UML המקורי, ואת התיקונים שאנחנו מציעים ב-UML המחודש.

משימה 3 – יצירת תרשימי רצף

כפי שלמדנו, כלי מאוד שימושי בעיצוב של תוכנה הוא יצירת תרשימי רצף (Sequence Diagram) כדי לתאר תהליכים שמתרחשים במערכת.

בפרויקט השחמט יש מספר תהליכים, ובתרגיל נרצה לתאר שניים מהם.

בתרגיל הזה נוכל להשתמש בשירות חינמי באינטרנט שמאפשר לנו ליצור תרשימי-רצף, חפשו אחד כזה בגוגל והשתמשו בו. בסוף התהליך אתם אמורים להגיש תמונה של התרשים.

כדי להיזכר איך תרשים-רצף נראה – נוכל להסתכל במצגת על הדוגמה של "צפייה ב-memes בפייסבוק".

משימה - ניצור שני תרשימי-רצף, ונוסיף אותם ל-repo:

1 - תרשים רצף בין ה-Frontend וה-Backend

התרשים הזה יתאר את מהלך הריצה של הפרויקט, וגם את פרוטוקול התקשורת בין ה-Backend ל-Frontend.

התרשים צריך לענות על השאלות הבאות:

- מתי ה-Frontend פונה ל-Backend?
- מתי ה-Backend פונה ל-Frontend?
- מה המידע שמועבר בין שני הרכיבים?
- מתי מסתיימת התקשורת ביניהם?

2 - ביצוע מהלך על הלוח

התרשים הזה יתאר את הזרימה של המידע בין רכיבי הפרויקט השונים (מחלקות) החל מהרגע שבו בוצע מהלך ב-GUI ועד לרגע שהוא המהלך מבוצע על הלוח ב-Backend.

התרשים צריך לענות על השאלות הבאות:

- מי מבצע את המהלך ?
- איזה מתודות משתתפות בתהליך?
- איך מוודאים שמהלך חוקי?
- איזה קוד חוזר ל-Frontend במצבים שונים?
(תזוזה לא חוקית, שח, הזזת כלי לא נכון וכו')

משימה 4 – סדנת Code Review

אחד הדברים שראינו בשיעור הוא שחשוב מאוד לקבל פידבק וביקורת על קוד שאנחנו כותבים לאורך העבודה.

Code Review הוא חלק בלתי נפרד מתהליך העבודה, ומאוד חשוב לתת לזוג עיניים נוספות לעבור על קוד שנדחף לפרויקט.

משימה - נסיים את משימת ה-Code Review שהתחלנו בשיעור (על פי ההוראות במסמך התרגיל). אל תשכחו להקפיד על העקרונות שלמדנו במצגת!

משימה - נבחר מחלקה או פונקציה **אחת** שבה יש הפרה/שגיאה לוגית/או שניתן לשפר אותה מתוך פרויקט השחמט, ונבצע עליה CR על פי העקרונות שלמדנו.

משימה 5 – GUI של גיט

אחרי שלמדנו לעבוד בצורה טובה עם גיט, לא חייבים להמשיך לעבוד עם ה-CMD. לפני שאנחנו מתחילים פרויקט גדול – חשבנו שזה יהיה יותר מגניב שנלמד לעבוד עם GUI של גיט. אז עבדנו ממש קשה והכנו לכם סדרת סרטונים (יש גם מסמך טקסט אם אתם מעדיפים) שמלמדים אותנו איך משתמשים בתוכנה **Git Extensions** (תוכנה מאוד קלה ואינטואיטיבית שמספקת GUI לעבודה עם גיט).

אתם לא חייבים לבצע את המשימה הזאת, אבל ממש כדאי לכם לצפות בסרטונים ולתרגל קצת עבודה עם Git Extensions כי הרבה יותר כיף לעבוד עם GUI. כמובן שלא צריך להגיש שום דבר. אם לא תספיקו לעבור על הסרטונים השבוע – מוזמנים לחזור אליהם מאוחר יותר.

הסרטונים נמצאים בתיקיית השיעור. מקווים שתאהבו! ❤️

בהצלחה!