**תיכון מונחה עצמים- פרויקט סיום קורס**

**מגישי הפרויקט:**

תום מנדל 205633688

ליטל סייג 313295412

1. **תבנית עיצוב Bulider**
   * **קוד ישן:**

Model- מספר שורות 29-30 ו37.

Lecturer- מספר שורות 19-25.

Permanent Lecturer- מספר שורות 16-20.

External Lecturer- מספר שורות 10-15.

* + **קוד חדש:**

Model- מספר שורות 41-45 ו59-62.

Lecturer- מספר שורות 31-66.

Permanent Lecturer- מספר שורות 17-25.

External Lecturer- מספר שורות 11-15.

* + **בעיה קיימת:**

יצירת אובייקט התלוי בכמה תתי אובייקטים שונים.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Bulider המשמשת להפרדת יצירת האובייקט מייצוגו, המאפשרת שימוש באותו התהליך של יצירת האובייקט לכמה אובייקטים שונים הבלתי תלויים אחד בשני.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו מאפשרים בניית אובייקט בצורה חכמה אשר איננה תלויה בתתי מרכיבים, מה שמאפשר תפעול נכון יותר של האובייקט ושינויו ע"י פיצול הconstructor לתתי רכיבים.

1. **תבנית עיצוב Bridge**
   * **קוד ישן:**

Model- מספר שורות 28-39.

* + **קוד חדש:**

הוספת interface בשם salaryCalc.

Model- מספר שורות 43, 61.

Lecturer- מספר שורות 68-78.

* + **בעיה קיימת:**

הזנת המשכורת ע"י המשתמש ללא בקרת מעסיק.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Bridge המשמשת להסרת תלות בין הממשק למימושו. נבחר להשתמש בתבנית זו לטובת חישוב משכורת המרצה. אנו רוצים שמשכורת המרצה תבנה ע"י הזנת נתונים מהמשתמש בזמן דינמי.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו מחשבים בצורה נכונה את המשכורת עבור כל סוג מרצה (מרצה זמני, מרצה קבוע). תהליך זה מתבסס על ירושה ממחלקת האב שהינה מחלקת המרצים. חישוב השכר יבוצע ע"י ניהול של שעות העבודה והתייחסות לוותק ולא כפי שנוהל בתוכנית הקודמת ע"י הזנת נתונים מהמשתמש ללא בקרה.

1. **תבנית עיצוב Command**
   * **קוד ישן:**

Controller- מספר שורות 118-145.

View- מספר שורות 316-329.

* + **קוד חדש:**

Controller- מספר שורות 112-162.

View- מספר שורות , 488-496372-390.

Model- מספר שורה 12-9.

הוספת class בשם Organization command.

הוספת interface בשם AddandRemCommand.

* + **בעיה קיימת:**

הכנסת מרצים משני הסוגים לתוך איגוד המרצים.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Command המשמשת לייצוג פקודות. נבחר להשתמש בתבנית זאת לטובת הוספת המרצים אל האיגודים השונים (איגוד מרצים זמנים, איגוד מרצים קבועים).

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע הכנסה והסרה של מרצים (זמניים וקבועים) אל תוך האיגוד ומחוצה לו. הבחירה להשתמש בתבנית נבעה מהשימוש הידוע ביותר שלה והוא שימוש בפקודות Apply וundo. יישום פקודות אלו בקוד מייצגות את אישור הוספת המרצה לאיגוד ובמקרה ההפוך הסרתו.

1. **תבנית עיצוב Null object**
   * **קוד ישן:**

Model- מספר שורות 143-148.

* + **קוד חדש:**

Model- מספר שורות 190-202.

הוספת class בשם checkLec.

הוספת class בשם console Log.

הוספת class בשם null Log.

הוספת interface בשם Log.

* + **בעיה קיימת:**

שימוש בפונקציית השוואת איגודי המרצים ללא בדיקה מקדימה של מערכי המרצים.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב null object המשמשת כברירת מחדל לאובייקט ובכך מונעת את הצורך לבדוק האם האובייקט בו אנו משתמשים הוא null.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע בדיקה של מערכי המרצים השונים לטובת מימוש הפונקציה של השוואת האיגודים. השימוש העיקרי של תבנית זו הינו לבדוק האם האובייקט הנוכחי הוא null. במקרה שלנו הבדיקה הינה הכרחית לטובת מימוש פונקציית ההשוואה כיון שאין טעם לממש את הפונקציה כאשר המערך הוא ריק. היתרון הגדול ביותר הוא חיסכון בזמן ריצה מיותר והתראה מקדימה למשתמש.

1. **תבנית עיצוב Proxy**
   * **קוד ישן:**

College- מספר שורות 9-10.

Model- מספר שורות 28-40.

* + **קוד חדש:**

הוספת interface בשם iCollege.

הוספת class בשם college proxy.

College- מספר שורות 6,14-15,23-30,41-45.

Model- מספר שורות 47-57,65-75.

* + **בעיה קיימת:**

קליטת מרצים חדשים ללא בדיקת עמידות בתנאי העבודה.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Proxy המאפשרת גישה לשליטה באובייקט אחר. נבחר להשתמש בתבנית זו עקב השליטה הקיימת לאובייקט המקורי מה שמאפשר לנו לנהל את ההרשאות השונות עבורו.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע בדיקה עבור המרצים הקבועים לטובת עמידה בתנאי העבודה לקבלתם למכללה. נבחר להשתמש בתבנית זו כמטרה לשליטה בהרשאות הקיימות לאובייקט, קבלה רק לאחר חמש שנות ניסיון. תבנית זו מאפשרת לנו הכנסה מבוקרת של המרצים הקבועים ואישור עמידתם בתנאים ללא צורך בביצוע סינון.

1. **תבנית עיצוב Observer**
   * **קוד ישן:**

Model- מספר שורות 28-40, 129-140.

Lecturer- מספר שורות 12-17.

* + **קוד חדש:**

Model- מספר שורות 153-15642-53, , 61-71, 123-126, 134-137,177-195

, 164-167.

Lecturer- מספר שורות 23-37, 6.

Organization- מספר שורות 7,41-45.

הוספת class בשם Observable.

הוספת class בשם PropertyChangedEventArgs.

הוספת interface בשם observer.

* + **בעיה קיימת:**

בחירת היו"ר הינה ללא בקרה ארגונית ונתונה לגמרי לבחירת המשתמש.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Observer המאפשרת הגדרת תלות של אובייקט אחד לרבים. לעיצוב זה קיים מנגנון הפשטה של שמירת האובייקט ושימוש חוזר לאחר שינוי. שינוי של אובייקט אחד גורר שינוי באובייקט אחר.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע בקרה על בחירת יו"ר האיגוד ובכך נמנע את השארת הבחירה כולה למשתמש. האופציה הראשונית הינה בחירת המרצה הראשון שהוכנס למערך והגדרתו כיו"ר. לאחר שתבוצע כלל הכנסת המרצים למערך יתבצע הסינון ויבחר היו"ר החדש ע"פ התנאים שהוגדרו מה שיגרור שינוי עבור האובייקט הקיים שהוגדר בתחילה.

1. **תבנית עיצוב Mediator**
   * **קוד ישן:**

Model- מספר שורות 46-56.

* + **קוד חדש:**

Model- מספר שורות 15, 82-98.

הוספת class בשם chatroom.

* + **בעיה קיימת:**

שליחת מתנה לחג ע"פ סוג המרצה והמכללה בה המרצה מועסק.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Mediator המשמשת לטובת תיווך בין אובייקטים מקושרים. השימוש העיקרי של תבנית זו כאשר יש צורך בשימוש חוזר בפונקציה.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע שליחת הודעות למאגר המרצים עבור המכללה שאותה נבחר ע"י יצירת צ'אט. התבנית מאפשרת לנו הצגה של ההודעות שנשלחו עבור כל מרצה שנמצא במערך מבלי לבצע חזרתיות על הפונקציה ובכל פעם מבצעת עדכון של השינוי בתיבת השיח בהתאם למצב הנוכחי.

1. **תבנית עיצוב Memento**
   * **קוד ישן:**

View- מספר שורות 170-171.

Controller- מספר שורה 72.

* + **קוד חדש:**

View- מספר שורות 79-81 ,108-117,160-194, , 593-601, 475-478660-664.

Controller- מספר שורה 70-73, 190-193.

Model- מספר שורות 213-254.

הוספת class בשם memento.

הוספת class בשם salarySim.

* + **בעיה קיימת:**

ביצוע סימולציית שכר שנתית עבור מרצה קבוע.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Memento המשמשת לטובת שמירה של חלק מסוים מתוך האובייקט שבכל רגע נתון נוכל לחזור אליו. השימוש העיקרי של תבנית זו הינו כאשר אנו לא רוצים לפגוע בממשק הישיר של האובייקט ונוכל לחזור למצב אותו הגדרנו כנקודת הrestore.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע חישוב של סימולציית השכר. החישוב יתבצע לפי שנות הוותק של המרצה, שכר הבסיס המוצע כתלות במקדם הוותק. יישום התבנית בפונקציה זו הינו כאשר אנו יכולים לבחור לאיזו נקודת זמן בהתאם לנקודות השחזור שיצרנו נרצה להגיע ע"מ לבצע את החישוב הרלוונטי לשנה הנוכחית.

1. **תבנית עיצוב Prototype**
   * **קוד ישן:**

College- מספר שורות 7-11.

Model- מספר שורות 18-19.

* + **קוד חדש:**

College- מספר שורות 6, 9-12, 17-22, 47-51.

Model- מספר שורות 26-36.

* + **בעיה קיימת:**

יצירת כפילות של מכללה בכל הזנה של המשתמש כאשר מוזנת מכללה שכבר קיימת.

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Prototype כאשר השימוש העיקרי של התבנית הינו לייצר שכפול של אובייקט קיים. נשתמש בתבנית זו כאשר נרצה לשכפל מספר מופעים מאותו אובייקט במחלקה במקום ליצור מופע חדש בכל פעם.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע שכפול של מכללה עם כלל נתוניה כאשר היא תופיע יותר מפעם אחת ע"י קבלת הנתונים מהזנת המשתמש. תבנית זו עוזרת לנו ביעילות זמן ריצה קצר יותר שנובע משכפול האובייקט במקום יצירתו מחדש וגם מונעת מכפל באחסון הנתונים.

1. **תבנית עיצוב Singleton**
   * **קוד ישן:**

Controller- מספר שורות 171-190, 31-53.

* + **קוד חדש:**

Controller-מספר שורות 29-35, 195-205.

הוספת Class בשם File Singleton.

**בעיה קיימת:**

יצירת קבצי התוכנית לאחר שמירתה

* + **תבנית העיצוב שנבחרה:**

הינה תבנית עיצוב Singleton המשמשת לשם הקצאת מופע יחידי של האובייקט. המופע היחידי הינו של המחלקה הראשית ושאר המחלקות מתממשקות אליו.

* + **מדוע:**

בשימוש בתבנית עיצוב זו אנו נבצע יצירה חד פעמית של קבצי האובייקט ונקצה עבורו מקום יחידי בזיכרון. המשתמש יבצע שימוש בקבצים אלו מבלי לערוך אותם.