### תבנית מס' 1 – Adapter

* סיבת הבחירה / שימוש בתבנית:

בחרנו לממש Adapter ל- List על מנת לאפשר Lazy Creation ושמירה של נתונים לתוך סינגלטון מערכתי שיצרנו. השימוש ב-Adapter מאפשר יצירה אוטומטית שלו על ידי הסינגלטון רק בעת שקוראים לו.

* אופן המימוש:

ה-Adapter הוא המחלקה UserRankListAdapter<T> שמכיל בתוכו שדה List<UserRank<T>>. יש לו בנאי פרטי שמגדיר את השדה כ-null. יש לו מתודה סטטית פרטית initUserRankListAdapter שדרכה הסינגלטון קורא ליצירת האובייקט.  
בקוד שלנו אנחנו משתמשים באפשרות הזו כאשר מוצאים עבור המשתמש את רשימת האנשים שמתוייגים איתו הכי הרבה בתמונות, וכאשר מוצאים את רשימת האנשים שהיו איתו בהכי הרבה אירועים. ה-Adapter מאפשר לנו לשמור את המידע בסינגלטון ולהימנע מבדיקה חוזרת שתניב בדיוק את אותה תוצאה.

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם

### תבנית מס' 2 – Facade

* סיבת הבחירה / שימוש בתבנית:

רצינו אובייקט שישמור בתוכו כחבר אובייקט מסוג אחר ויספק סטטיסטיקה נוספת לגבי אותו אובייקט, ובנוסף יסתיר פעולות שאפשר לעשות על החבר בד"כ ויחשוף רק מתודות שרלוונטיות לשימוש שלנו לצורך פשטות.

* אופן המימוש:

ה-Facade אצלנו הוא מחלקה בשם UserRank<T> הנוצרת מאובייקט FacebookWrapper.ObjectModel.User ומכילה אותו כחבר פרטי ומסתירה את כל המתודות הקיימות שלו מלבד User.ImageSquare. בנוסף המחלקה מאפשר לשמור רשימת אובייקטים המשוייכים לאותו יוזר, לקבל את גודל הרשימה, ולהשוות בין אובייקטים מאותו סוג (קודם לפי גודל הרשימה ואז לפי שם המשתמש).  
בקוד שלנו אנחנו משתמשים באובייקט הזה כאשר מוצאים עבור המשתמש את רשימת האנשים שמתוייגים איתו הכי הרבה בתמונות, וכאשר מוצאים את רשימת האנשים שהיו איתו הכי הרבה אירועים. ה-Facade מאפשר לנו לשמור עבור כל משתמש שמצאנו את האירועים/תמונות שהוא נמצא בהן, לעשות השוואה בין המשתמשים כאשר נרצה למיין לפי הכמות שנמצאה, להוציא את התמונה של המשתמש, ולהציג שם + כמות ב-UI שלנו (תוך שימוש בשדה “NameCount”).

* Sequence Diagram

[sequence diagram שמציגה את התהליך שקשור לתבנית והאינטראקציה בין המחלקות]

* Class Diagram

[class diagram שמתארת את המחלקות שלכם שמעורבות בתבנית (תיאור מלא שכולל Properties ו- Methods) והיחסים ביניהם (תאור מלא עבור כל יחס כפי שלמדנו בכיתה). עבור כל מחלקה שלכם, כיתבו מי המקבילה שלה (אם יש כזו) בתבנית (Pattern) שבחרתם