### תיאור קצר של הפיצ'רים

My Albums

פיצ'ר זה מאפשר למשתמש לדפדף ברשימת התמונות הקיימות באלבום ולבחור בקלות ע"י סימון התמונות הרצויות מהאלבום ולאחר בחירת נתיב ההורדה במחשב, התמונות המסומנות ירדו יחד.

שיפור נוסף למשתמש יש אפשרות סינון של התמונות באלבום על פי מאפיינים : כמות לייקים, כמות תגובות ולפי תאריך העלאה.

הוספנו אפשרות סינון לרשימת האלבומים לפי כמות התמונות באלבום (סדר עולה/יורד), תאריך יצירה של האלבום (סדר עולה/יורד).

Video Streamer

פיצ'ר זה מאפשר למשתמש לצפות בסרטונים הקיימים על קיר המשתמש באמצעות נגן אפליקטיבי ע"י בחירתם מהפלייליסט ומאפשר למשתמש חווית צפייה נוחה

שיפור נוסף הורדת הסרטונים מהפלייליסט למחשב בצורה נוחה.

### תבנית מס' 1 – [Strategy]

* סיבת הבחירה / שימוש בתבנית:

באפלקציה שלנו קיים פיצ'ר שבו ניתן לבחור אלבום ספציפי מתוך רשימת האלבומים של המשתמש. לנוחות השימוש של המשתמש אפשרנו מיון של האלבומים לפי מס' התמונות באלבום באופן דומה אפשרנו למשתשמש אפשרות מיון הסרטונים עפ"י כמות הלייקים של הסרטון.

* אופן המימוש:

יצרתי מחלקה ValueSorter שמחזיקה מצביע למחלקה האבסטרקטית OrderDefiner המחלקה ValueSorter מקבלת בבנאי שלה מיון קונקרטי מאחד המחלקות הבאות:

OrderAlbumsByCreatedTimeASC, OrderAlbumsByCreatedTimeDEC OrderAlbumsByPhotosCountDEC OrderAlbumsByPhotosCountASC,

מחלקות אלו יורשות מ OrderDefiner וממשות את המטודה האבסטרקטית IsNeedSwap בהתאם למיון המבוקש. את המיון הרצוי המתודה Sort מסדרת את הרשימה הספציפית בהתאמה כאשר ב injection point מוחזרת תשובה בוליאנית בהתאם לאסטרטגית המיון הנבחרת.

תפקידים:

ValueSorter הוא Context

OrderDefiner הוא Strategy

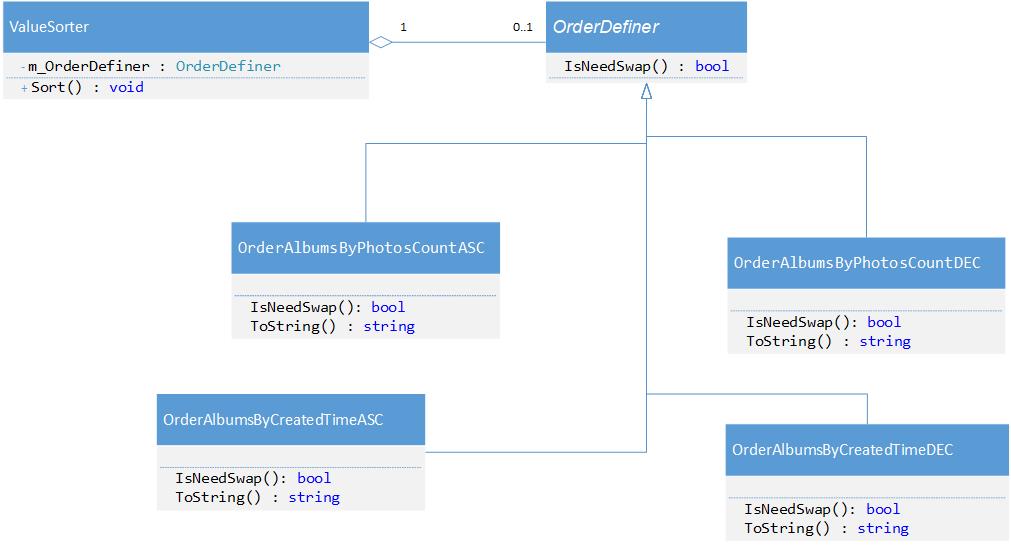
ConcreteStrategy:

OrderAlbumsByCreatedTimeASC

OrderAlbumsByCreatedTimeDEC

OrderAlbumsByPhotosCountASC

OrderAlbumsByPhotosCountDEC



### C:\Users\Tzach\AppData\Local\Microsoft\Windows\INetCache\Content.Word\StrategySequence.jpg

### תבנית מס' 2 – [Observer]

* סיבת הבחירה / שימוש בתבנית:

להגדיר תלות יחיד-רבים בין אובייקטים, כאשר אחד משנה state כל השאר מתעדכנים אוטומטית. השימוש אצלנו בא לידי ביטוי כאשר הלקוח מעונין לקבל אינפורמציה מלאה ע"י לחיצה על כפתור fetchOrClearAllInformationButton ומעדכן את שאר כפתורי קבלת המידע הספציפיים על מצבו ובהתאמה שינוי מצבם ללא לחיצים ובאופן דומה לחיצה נוספת על fetchOrClearAllInformationButton מאפסת את המידע ואת מצב הכפתורים למצבם המקורי (לחיצים).

* אופן המימוש:

מימשתי את תבנית העיצוב Observer בכך שתחילה יצרנו מודיע בשם AllInfoButtonNotifier שמחזיק רשימה של מאזינים

כמו כן יצרנו מחלקה של ButtonObserver שבה הבנאי מקבל כפרמטר את הכפתור הספציפי שרוצה להירשם לאירוע.

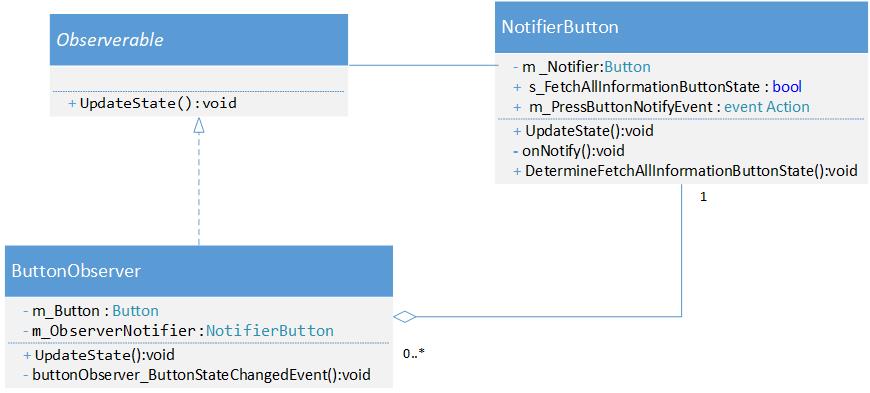
הוספנו את המאזינים למערך של AllInfoButtonNotifier כדי שהמודיע יידע על שינוי מצבו לרשימת המאזינים ובכך כל אוביקט שמאזין יפעל בהתאם.

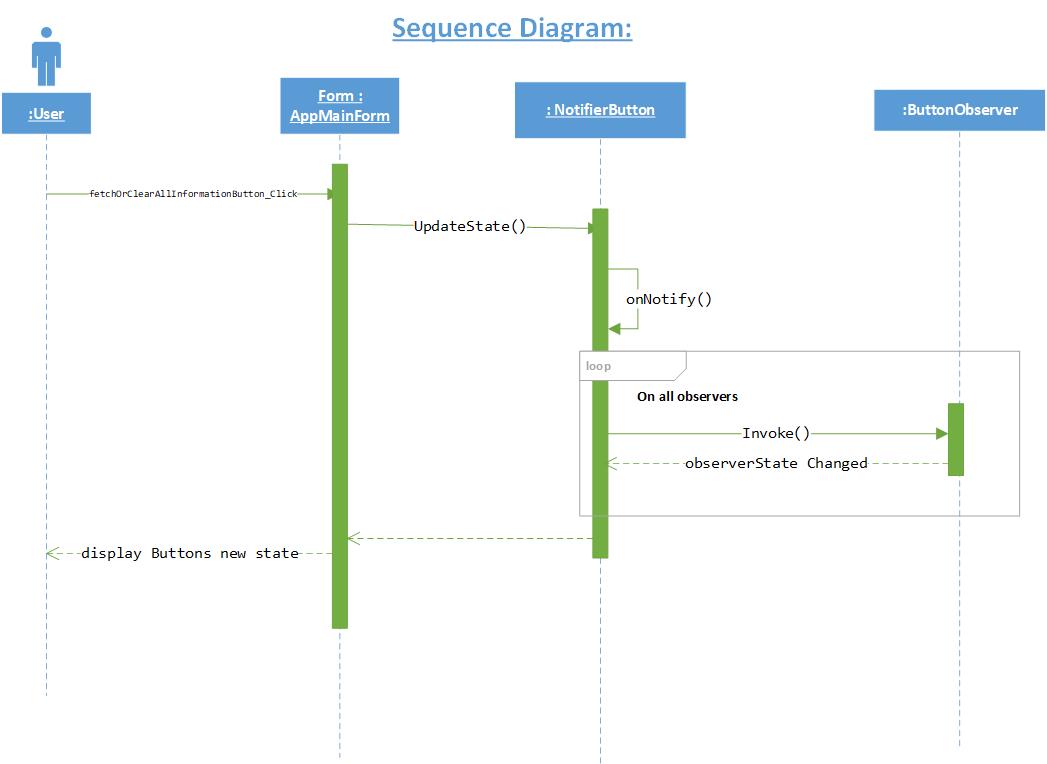
תפקידים:

NotifierButtonהוא Subject

Observerable הוא Observer

ButtonObserver הוא Concrete Observer





### תבנית מס' 3 – [Iterator]

* סיבת הבחירה / שימוש בתבנית:

מעבר על קולקציות שונות בצורה כללית מבלי חשיפת היצוג הפנימי של הרכיב.

במקרה שלנו מעבר על אוסף תמונות מתוך אלבום מסוים בעת הורדת תמונות מהאלבום.

אופן המימוש:

עטפתי את השורות של ה DataGrid כאשר בכל שורה קיימת תמונה והמאפינים שלה (כמות לייקים, תאריך העלאה , כמות תגובות ) וייצרנו איטרטור

קונקרטי rowsDataGridIterator שממש את Iteratorב pattern זה השתמשנו ביכולות השפה C# 3 באמצעות yield return.

תפקידים:

AppMainFormהוא Client

IAggregate הוא Aggregate

DataGridRowsAggregateהוא Concrete Aggregate

IIterator הוא Iterator

RowsDataGridIterator הוא Concrete Iterator

