### תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הקודם:

* רשימה שמסננת את העמודים שהמשתמש אהב ומראה רק עמודים שלא היו פעילים מספר מסויים של ימים (בין 1 ל-60, לבחירת המשתמש) במטרה להיות מועמדים למחיקה על ידי המשתמש (ניקוי של עמודים לא פעילים מהפרופיל).
* חלון עריכת תמונה אשר מאפשר שימוש בפילטרים גרפיים מובנים ויצירת פילטרים גרפיים חדשים, שמירתם ושימוש חוזר שלהם על תמונות נוספות. אפשרות העלאה של התמונה שנוצרה לפרופיל בפייסבוק (בפרטיות Only Me).
* יכולת ליצור פילטרים חדשים של תמונות, להפעיל את הפילטרים על התמונות פרופיל, ולהעלות אותן לפייסבוק. יכולת ליצור קבוצות של פילטרים במבנה של עץ, ויכולת לבצע שינויים בקבוצה של פילטרים שישפיעו אוטומטית על כל הפילטרים שבתוכה.

### תבנית מס' 1 – Iterator

* סיבת הבחירה / שימוש בתבנית:

בתרגיל הקודם, מלבד האופציה להגדיר פילטרים חדשים שהמשתמש יוצר בעצמו, הוספנו את האופציה ליצור קבוצות של פילטרים ולשנות את אופיין ביתר קלות.

ככל שהמשתמש יוצר מגוון רחב יותר של פילטרים חדשים/קבוצות של פילטרים התמצאות בתפריט הפילטרים עלול להיות מעיק ולא פרודוקטיבי, ולכן טבעי היה להוסיף אופציית חיפוש פילטר מקיפה לפי שם.

פשוט הזן את שם הפילטר בשדה החיפוש, ותקבל אוטומטית את הפילטרים המתאימים.

* אופן המימוש:

מכיוון שמבנה הנתונים המאכלס את כל הפילטרים/קבוצות פילטרים, הוא למעשה הcomposite שמימשנו בתרגיל הקודם [במקרה שלנו הוא FilterGroup] נוצר לנו מבנה מסוג עץ, כך שבתוך FilterGroup בודד עשויים להיות תתי קבוצות, תתי תתי קבוצות וכך הלאה.

לשם חיפוש שמי במבנה מורכב זה, יצרנו עבורFilterGroup מחלקה מקוננת בשם FilterGroupIterator, שהיא למעשה מממשת את הממשק IEnumerator<IFilter>

הtraverse שמומש באיטרטור הוא למעשה מימוש של DFS באמצעות מחסנית של איטרטורים. ככל שיורדים בעומק המבנה, נדחפים אובייקטים למחסנית, וכשמסיימים רמה נשלף האובייקט העליון מהמחסנית.

כדי להשתמש ביכולות של linq שהכרנו בכיתה, FilterGroup מימשה את הממשק IEnumerable, ואז ביצענו query פשוט שמסנן את כל האובייקטים לפי השם שניתן בחיפוש, והצגנו את התוצאות.

* Sequence Diagram



* Class Diagram

Aggregate : IEnumerable

Iterator : IEnumerator

ConcreteAggregate: FilterGroup

ConcreteIterator: FilterGroupIterator

Client: FormImageEditing



### תבנית מס' 2 – Strategy

* סיבת הבחירה / שימוש בתבנית:

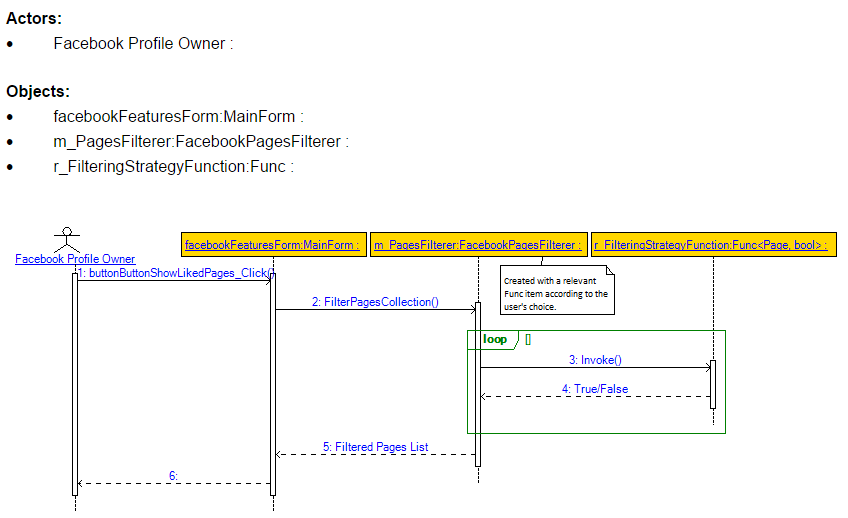
משתמש פייסבוק לעתים ירצה לראות אילו עמודים שהוא "אהב" כדאי לו למחוק מהפייסבוק. לשם כך הוא ירצה לראות רק את חלקם בהתאם לפילטר מסויים. לדוגמה: פילטר שמראה רק את העמודים הלא פעילים, או פילטר המראה רק את העמודים בשפה מסויימת, או פילטר שמראה רק את העמודים שיש להם פחות ממס' לייקים מסויים וכו'. זאת במטרה לדעת אילו עמודים כדאי לו (לפי בחירתו) למחוק.

* אופן המימוש:

המימוש נעשה בעזרת מחלקה בשם FacebookPagesFilterer שה-MainForm מחזיק ממנה אינסטנס. במחלקה יש פונקציה שיודעת לעבור על מקבץ של דפי פייסבוק, ולהחזיר מקבץ חדש רק של הדפים שעברו את הפילטר בהצלחה. את המחלקה יוצרים עם הפונקציה שתהווה את האסטרטגיה של הפילטור (בתור פרמטר - Delegate). נכון לכרגע בקוד יש 4 אסטרטגיות פילטור שונות שאיתן המחלקה יכולה להיווצר.

הקונטקסט הוא ה MainForm-שמשתמש ב-Strategy שהוא FacebookPagesFilterer, הנוצר עם פונקציה ש-MainForm שולח לו כאסטרטגיה שבה הוא ישתמש. האינסטנס של FacebookPagesFilterer שיש בו פונקציה מסויימת, הוא ConcreteStrategy.

* Sequence Diagram



* Class Diagram

Context: MainForm

Strategy: IPagesFilterer

ConcreteStrategy1: InactivePageFilter

ConcreteStrategy2: LikesFilter

ConcreteStrategy3: HebrewFilter

ConcreteStrategy4: EnglishFilter



### תבנית מס' 3 – Visitor

* סיבת הבחירה / שימוש בתבנית:

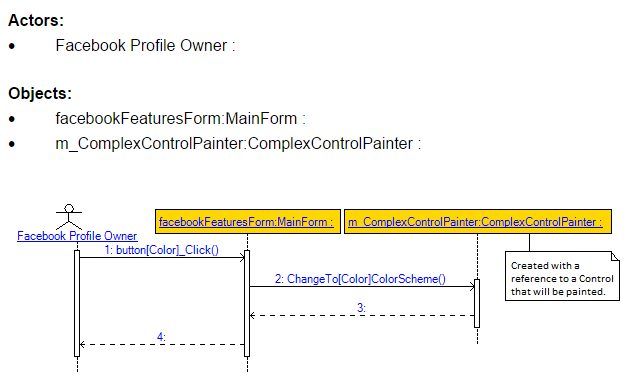
אנו רוצים להנעים את השימוש באפליקציה עבור המשתמש ולכן מאפשרים צביעה של כל אחד מהחלונות באפליקציה בנפרד בסכמת צבעים אחת מתוך 4 אפשרויות.

* אופן המימוש:

ה-Visitor הוא המחלקה ComplexControlPainter. הוא מקבל אובייקט קונטרול ויודע לצבוע את חלקיו השונים לפי סכמת צבעים קבועה. יש שימוש באובייקט של מחלקה זאת בכל אחד מהחלונות שקיימים באפליקציה, ולכן הלוגיקה של צביעת החלון יצאה מחוץ לטופס ספציפי אחד.

ה-Visitor כאן הוא המחלקה ComplexControlPainter אשר יודעת לקבל רפרנס לאובייקט מסוג Control (שהוא הורה של Form) ולצבוע אותו באחת מ-4 ערכות צבע שונות. אנו מחזיקים אובייקט Visitor כזה בכל אחד מהטפסים האופציונליים באפליקציה ומשתמשים באותו קוד שלו בכדי לצבוע כל אחד מהטפסים (המשתמש בוחר כפתור צביעה בטופס והטופס נצבע על ידי ה-Visitor).

* Sequence Diagram



* Class Diagram

Visitor: ComplexControlPainter

Element: Control

ConcreteElement1 : AdditionalPageInfoForm

ConcreteElement2 : FormImageEditing

ConcreteElement3 : MainForm

