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

* Best Posts Finder – מאתרת עבור המשתמש את הפוסטים הכי פופולריים שלו על ידי פרמטרי חיפוש שהוגדרו על ידי המשתמש.
* Facebook Conditionally – מאפשרת למשתמש לבצע פעולות פייסבוק כגון לייק או תגובה באופן מתוזמן ובתנאי מימוש שהוגדרו על ידי המשתמש.

**בתרגיל הנוכחי** מימשנו **Two Way Data Binding** כך שניתן לראות את הטקסט של Pending Facebook Action בתוך הTextBox וניתן לשנות את הטקסט הנ"ל (כל עוד הפעולה עדיין נמצאת בPending Actions ולא בוצעה כמובן). ניתן לראות זאת בקוד במקומות הבאים:

* void UpdateTextOfTask(string text);
* public abstract string GetTextOfTask();
* private void listBoxPending\_SelectedIndexChanged(object sender, EventArgs e)
* private void textBoxPrepareTextToSubmit\_Leave(object sender, EventArgs e)

כמו כן, מתבצע שימוש ב**Threads** בשני הפיצ'רים:

* BestPostFinder – בעת לחיצה על הכפתור GenerateResults בUI נפתח Thread חדש אשר מוצא את הפוסטים הכי "טובים" ובמקביל המשתמש יכול להמשיך לבצע שינויים בפילטרים.
* Facebook Conditionally – בעת טעינת הForm פעולת הFetch של כל הפוסטים מתבצעת בThread חדש ובכך פותרת את הבעיה שעבור Thread יחיד המסך נתקע לכמה שניות בעת טעינת הפיצ'ר.

### תבנית מס' 1 – [Abstract Factory]

* סיבת הבחירה / שימוש בתבנית:

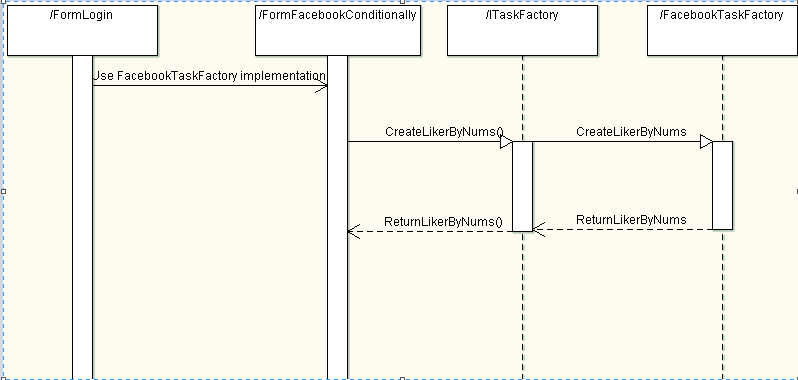
[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

ליצור מבנה אשר יתן לנו אפשרות בהמשך להחליף את הFactory של ייצור הCommenters ו-Likers לרשת חברתית אחרת שהממשק תואם לה.

* אופן המימוש:

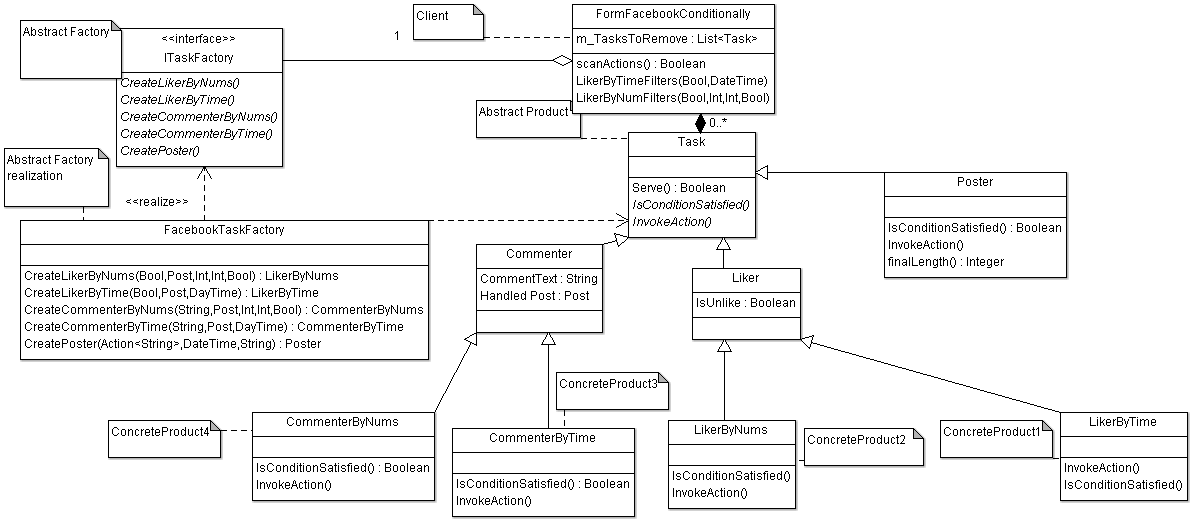
יצרנו Interface בשם ITaskFactory שמממש פעולות של יצירת משימות למיניהן ברשת החברתית. כמו כן, יצרנו מחלקה בשם FacebookTaskFactory אשר מממשת את הInterface באופן שמתאים לFacebook API כאשר בזמן יצירת הFormFacebookConditionally הוא מקבל ITaskFactory אשר מממש יצירת האובייקטים הנדרשים לנו לאינטרקציות בין-אישיות ברשת החברתית.

* 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 – [Class Proxy]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]

שכלול השירות שניתן לנו ע"י הListBox הקיים כך שיתאים לצרכינו. (הוספנו את היכולת הבאה: לצבוע את שלושת הListboxes ב-FormFacebookConditionally בוורוד כאשר ישנם ישנן פעולות Pending Actions ListBox, כאשרListbox זה מתרוקן מפעולות ברצוננו להחזיר את שלושת ה-Listboxes לצבעם המקורי, אך ברצוננו שהPending Actions ListBox יבצע זאת בצורה שונה מהמימוש של Listbox בכך שהוא מהבהב לפני שצבעו חוזר לצבע ברירת המחדל שלו.

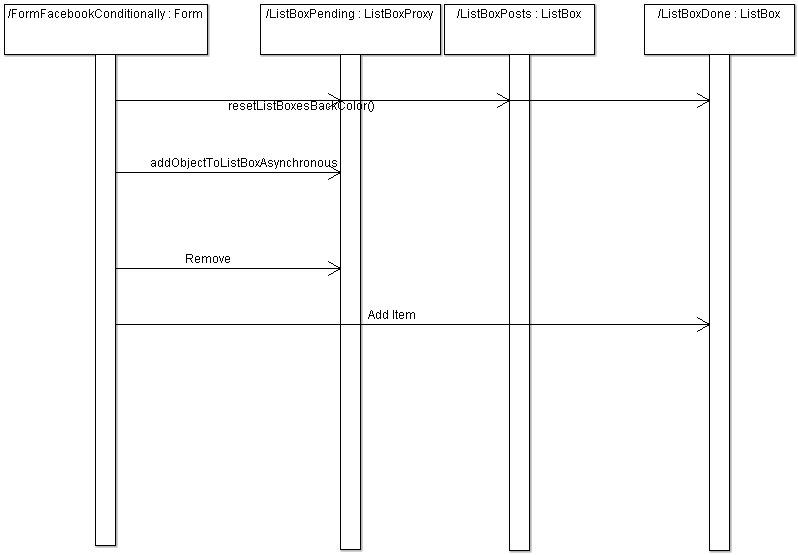
* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

יצרנו מחלקה חדשה ListBoxProxy שיורשת מListBox ובעצם תשמש חיקוי שלה מלבד שכלול השירות שניתן ע"י הפונקציה ResetBackColor כך שתבצע פעולת הבהוב לפני שהListBox חוזר לצבע ברירת המחדל שלו. כך אפשרנו פולימורפיזם בין ה-Listboxes (Pending actions ListBox הוא בעצם ListBoxProxy וכל השאר הם מסוג ListBox) ובנוסף Pending Actions Listbox שכלל את השירות ומאפשר לנו את מחוות ההבהוב תוך כדי תהליך חזרתו לצבעו המקורי (כאשר שאר הListboxes יחזרו לצבעם המקורי באופן רגיל לפי המימוש באבא Listbox)

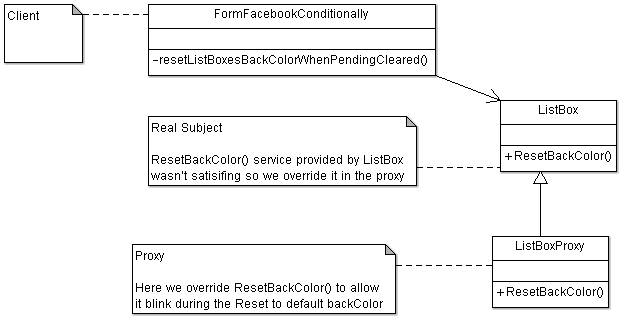
* Sequence Diagram

[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' 3 – [Method Factory]

* סיבת הבחירה / שימוש בתבנית:

[תיאור הסיבה / הצורך בשימוש בתבנית במערכת שלכם]  
מניעת יישום לוגיקת יצירה של אובייקט בClient

כלומר, FormBestPostFinder לא יצטרך להחליט ולבצע את היצירה של IPostScoreCalculator ספציפי

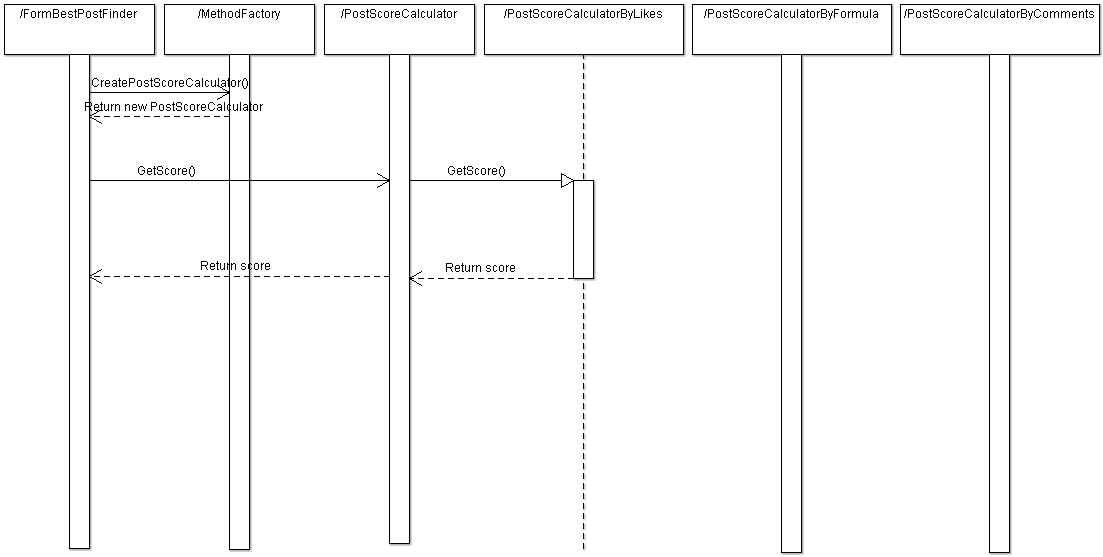
* אופן המימוש:

[תיאור המימוש והיכן ניתן למצוא אותו בקוד]

יצרנו Interface בשם IPostScoreCalculator שמממש ממשק של חישוב תוצאה של Post בעזרת מתודה בשם GetScore והוספנו 3 מחלקות ((..ByLikes,..ByComments,..ByFormulaאשר מממשות ממשק זה ומחשבות את התוצאה בצורה מסויימת. הקליינט (FormBestPostFinder) מחזיק כDate Member את הMethodFactory(PostScoreCalculatorFactory) ובזמן ריצה ,כאשר המשתמש מבקש להציג את הפוסטים הכי מוצלחים, על מנת לבצע את פעולת החישוב, הוא מייצר את הCalculator הרלוונטי ומחשב את התוצאה עבור כל Post

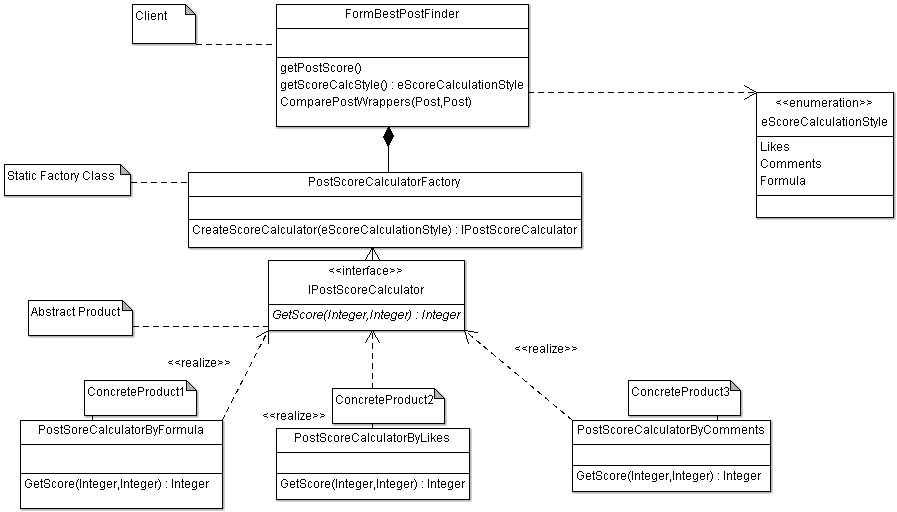
* Sequence Diagram

[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

****

