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

* 10 תמונות עם הכי הרבה לייקים.
* מידע מויקפדיה על הדפים שאהבתי.
* (הוספתי) משחק זיכרון – תמונה אקראית מאחד האלבומים מופיע וצריך לנחש מאיזה שנה התמונה.
* הקפצת התראה על אירוע שמתרחש באותו היום.

### תבנית מס' 1 – [Observer]

* סיבת הבחירה / שימוש בתבנית:

סיבת הבחירה:

היכולת להתריע למשתמש על אירוע שמתקיים מבלי שההתראה תכיר את האירוע. היכולת בעתיד להוסיף עוד אובייקטים שיאזינו לאירוע.

שימוש בתבנית:

ה- Subject הוא הProxyEvent-

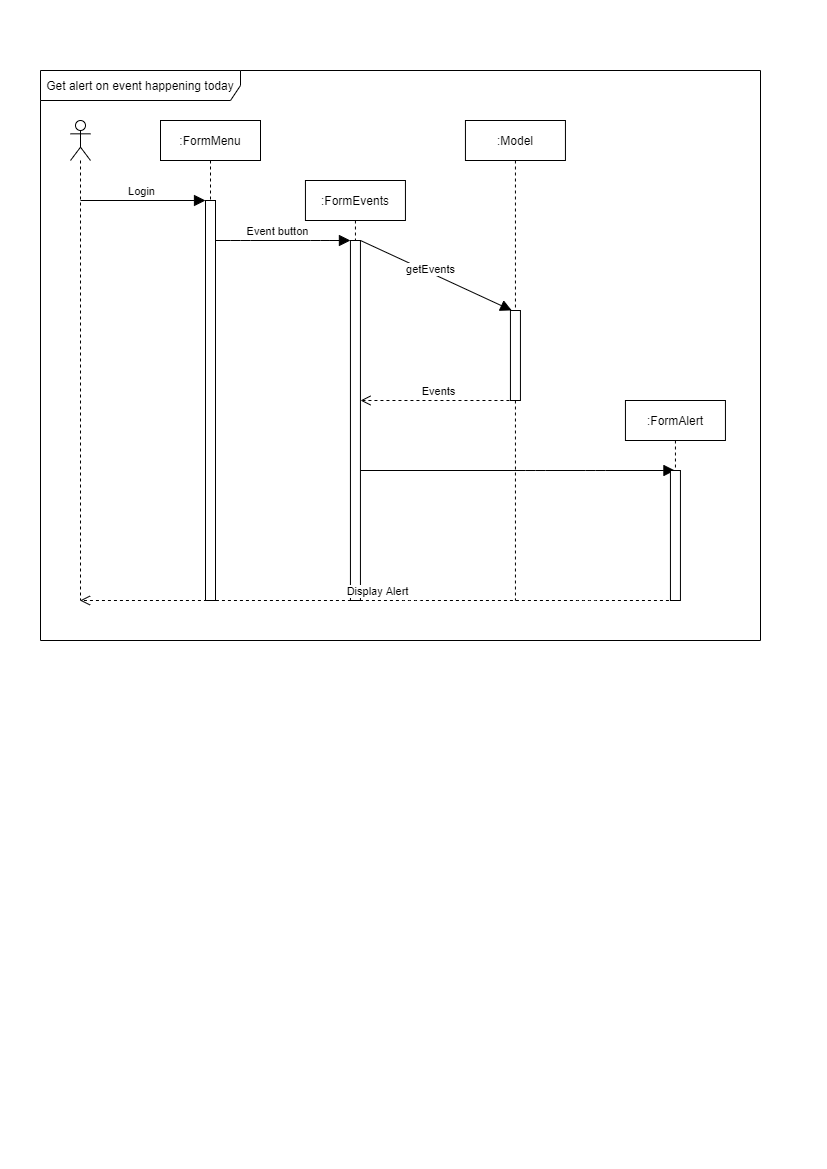
ה- Observer הוא ה- Alert (ובעתיד עוד אובייקטים שרוצים לדעת שיש היום אירוע)

* אופן המימוש:

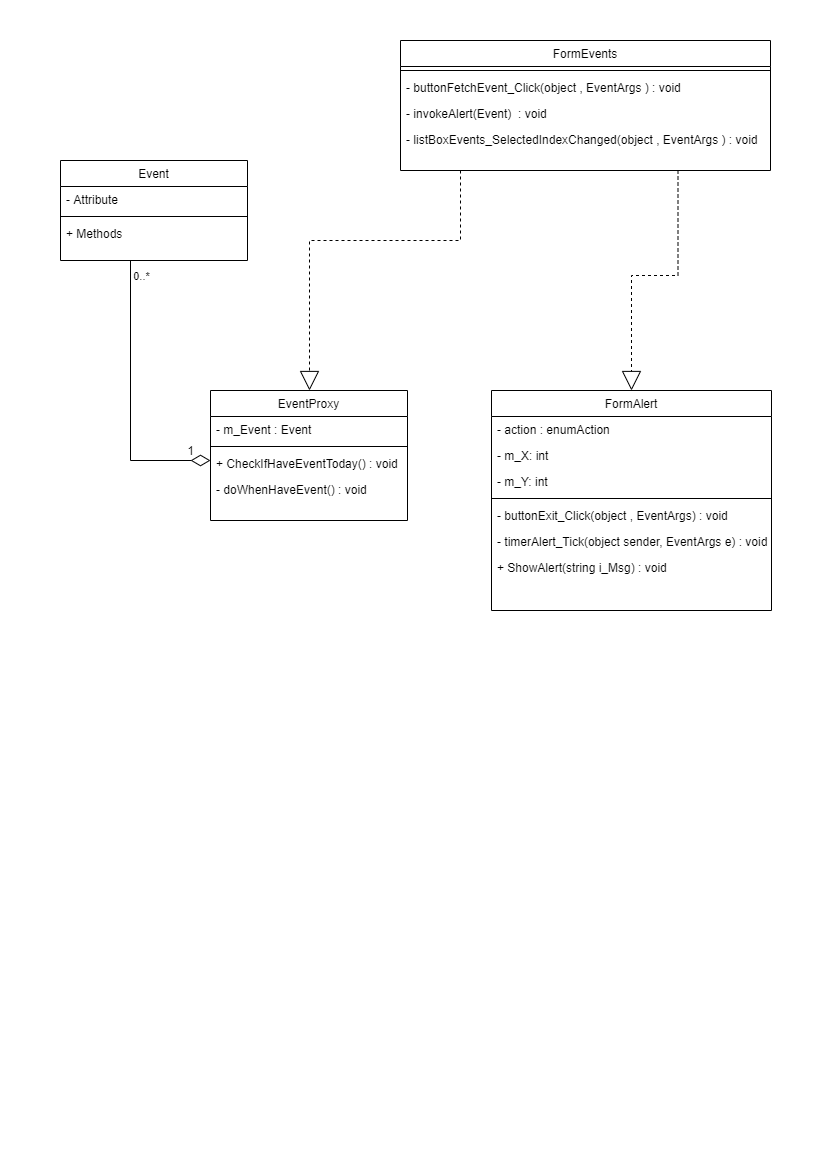
השתמשנו ב- Delegate

יצרנו FormAlert שמאזין לproxyEvent וכאשר הevent מתרחש באותו היום מופעל פוינטר לפונקציה שנמצא ב FormEvent

Sequence Diagram:



Class Diagram:



### תבנית מס' 2 – [Strategy]

* סיבת הבחירה / שימוש בתבנית:

סיבת הבחירה:

כדי לאפשר לשנות את אלגוריתם חישוב נקודה במשחק בצורה פשוטה ללא צורך לשנות ללא תלות במשתמשים במחלקה.

שימוש בתבנית:

Contex – FasadeGame

Strategy - ScoreStrategy

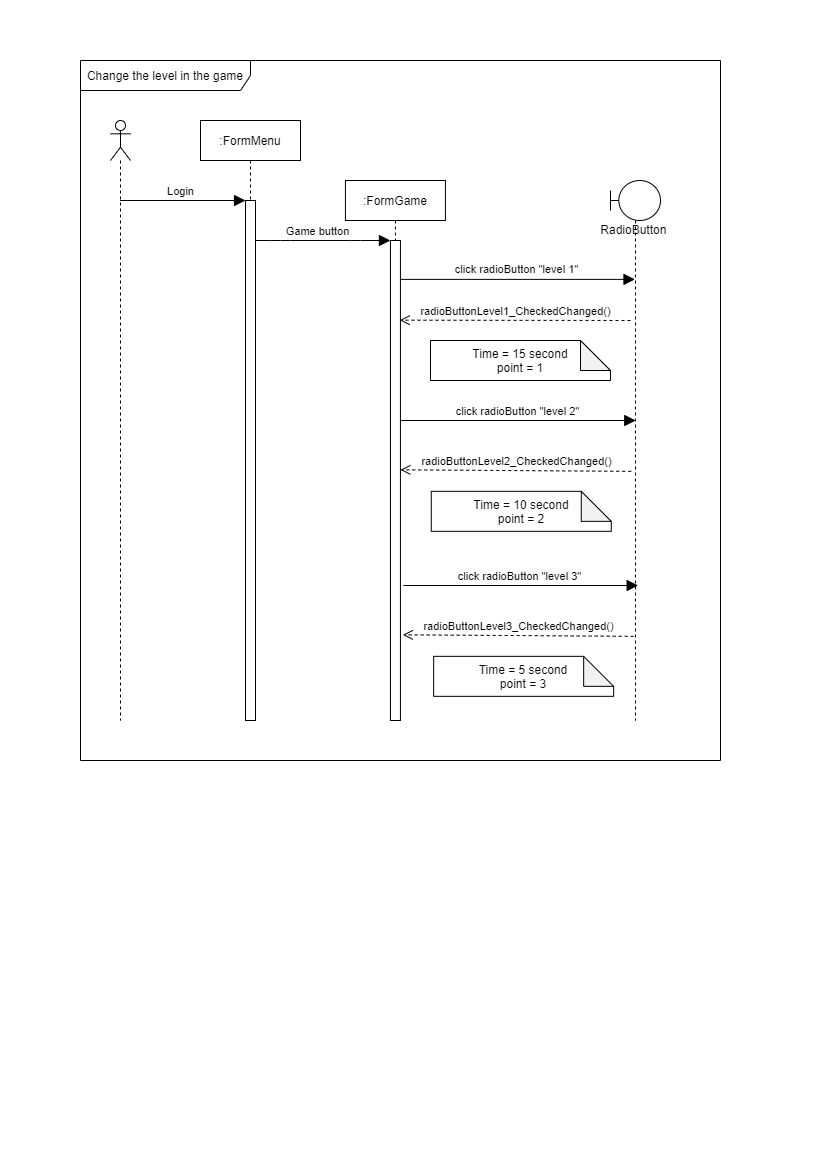
אופן המימוש:

ConcretStrtegy1 – r\_FacadeGame.ScoreStrategy = (win) => win \* 1;

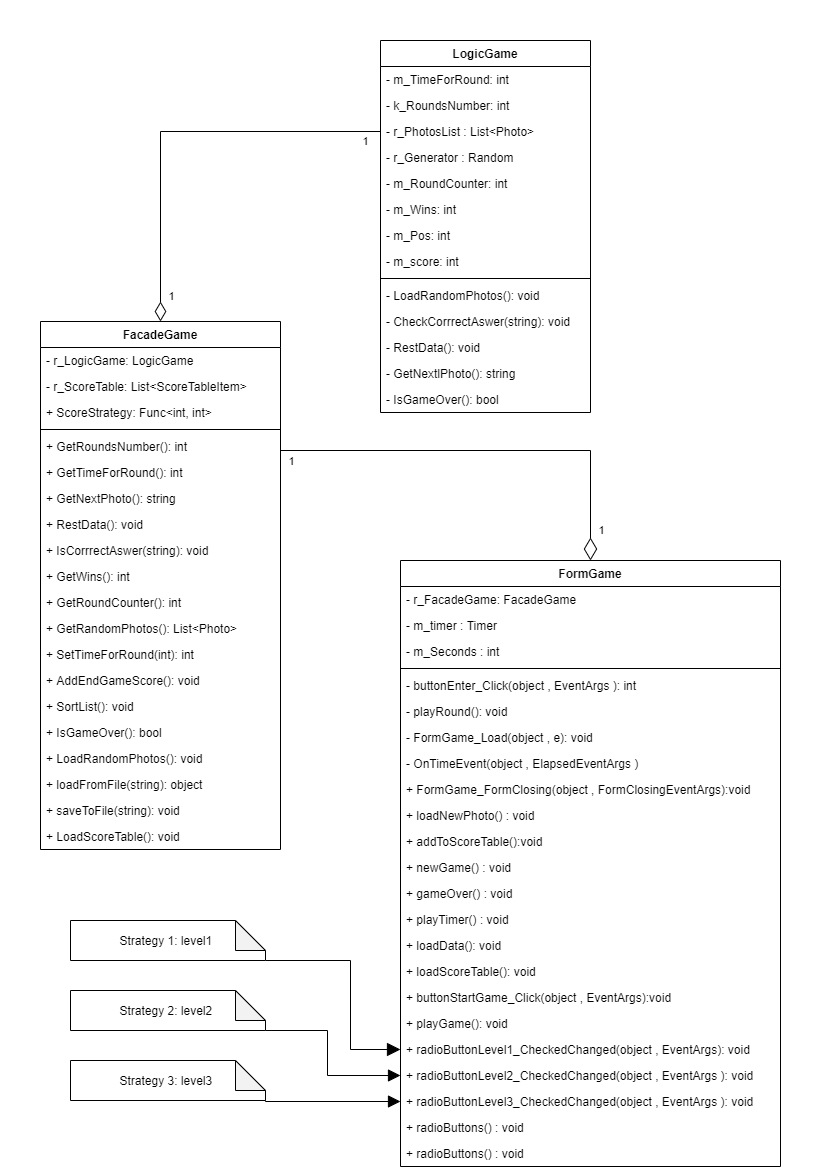
ConcretStrtegy2 - r\_FacadeGame.ScoreStrategy = (win) => win \* 2;

ConcretStrtegy3 - r\_FacadeGame.ScoreStrategy = (win) => win \* 3;

Sequence Diagram



Class Diagram



### תבנית מס' 3 – [Iterator]

* סיבת הבחירה / שימוש בתבנית:

סיבת הבחירה:

לגשת לאלמנטים באובייקט מבלי לחשוף את המבנה הפנימי שלו.

ובנוסף אם בעתיד נרצה לשנות את מבנה הנתונים לא נצטרך לשנות את הקוד.

שימוש בתבנית:

Aggregate – List<ScoreTableItem>

Iterator - ScoreIterator

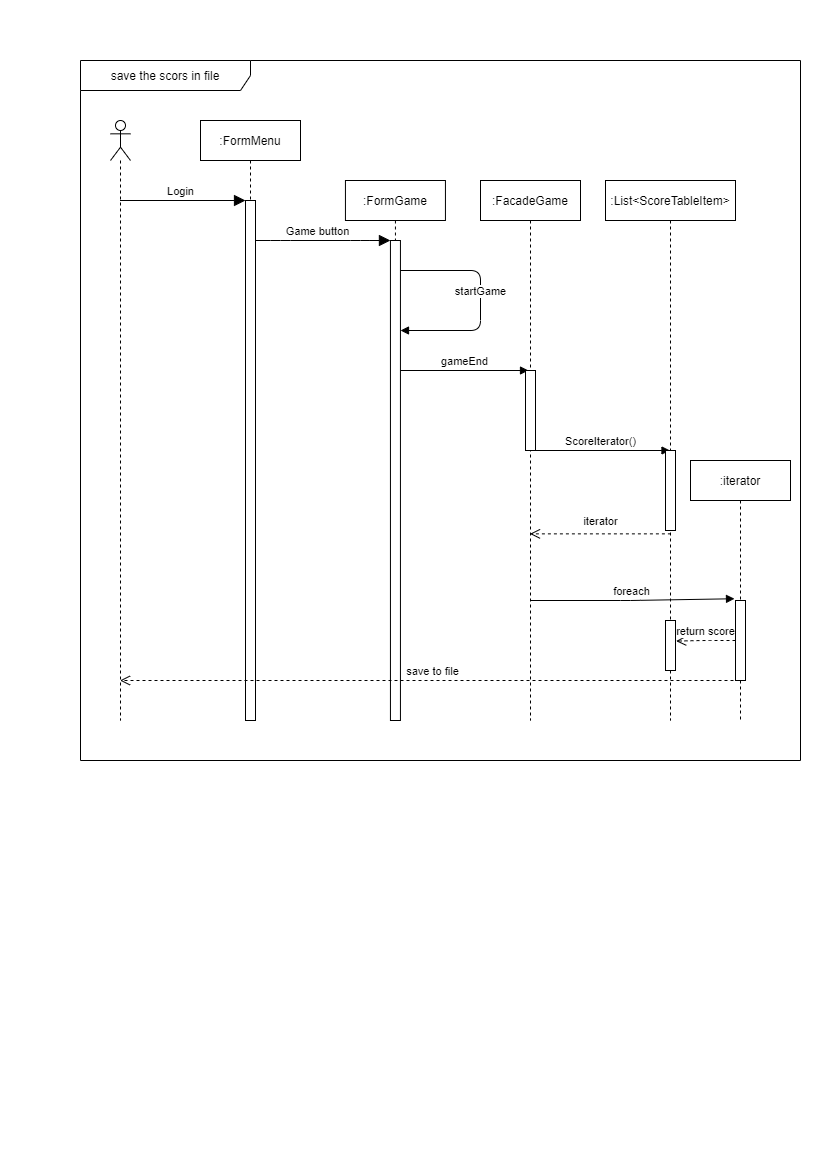
אופן המימוש:

השתמשתי בספריה Linq במחלקה FasadeGame

כדי לממש iterator שמביא את 10 התוצאות הטובות ביותר ללא תוצאות של 0

זה נעשה באמצעות המילים שמורות from,where,orderby,select

Sequence Diagram:



Class Diagram:

