תיאור כללי לשלב 4

להלן דוגמא אפשרית להגדרת הישויות והפונקציונאליות של השכבה הלוגית.

הדוגמא לא מחייבת, אך כל בחירה אחרת למימוש חייבת לכלול את הפונקציונאליות הנדרשת עבור הפעולה הסופית של הפרויקט, שהיא: לאפשר למנהל לנהל את מאגר החנות: מוצרים, מבצעים ולקוחות (כולל הוספה, מחיקה, עדכון ושליפה) וכן ביצוע קניה בחנות תוך שימוש במבצעים וחישוב הסכום הסופי לתשלום.

לצורך הפשטת הפרויקט, לא נשמור את ההזמנות בזיכרון, אלא הן יעברו כאובייקטים בין השכבות BL, UI ובסיום קניה, היא נמחקת מהזיכרון. (כמובן שיש לעדכן את המלאי בהתאם).

1. דוגמא אפשרית להגדרת הישויות הלוגיות:

**ישויות לוגיות ראשיות:**

BO.Client – מכילה את אותן תכונות כמו DO.Client

BO.Product – מכילה את אותן תכונות כמו DO.Product   
ובנוסף תכונה של רשימת מבצעים למוצר – מסוג SaleInProduct

BO.Sale – מכילה את אותן תכונות כמו DO.Sale

BO.Order – ישות חדשה לתיאור הזמנה. מכילה את התכונות הבאות:

* האם זו הזמנה של לקוח "מועדף" או לקוח מזדמן (bool)
* רשימת המוצרים בהזמנה (רשימה של BO.ProductInOrder)
* המחיר הסופי לתשלום (double)

**ישויות עזר לוגיות:**

BO.ProductInOrder – ישות עזר לתיאור מוצר בהזמנה. מכילה את התכונות הבאות:

* מזהה מוצר (int)
* שם מוצר (string)
* מחיר בסיס למוצר (double)
* כמות בהזמנה (int)
* רשימת מבצעים למוצר זה (רשימה של BO. SaleInProduct)
* מחיר סופי למוצר (double)

BO.SaleInProduct – ישות עזר לתיאור מבצע של מוצר. מכילה את התכונות הבאות:

* מזהה מבצע (int)
* כמות למבצע (int)
* מחיר (double)
* האם המבצע מיועד לכל הלקוחות (bool)

1. דוגמא אפשרית להגדרת הממשקים:

BlApi.IClient – אותן מתודות כמו DalApi.IClient, בתוספת מתודה לבדיקה האם לקוח קיים – מחזירה bool.

BlApi.IProduct – אותן מתודות כמו DalApi.IProduct בתוספת מתודה לקבלת כל המבצעים שבתוקף של המוצר (מקבלת קוד מוצר ומשתנה בוליאני לציון האם לקוח "מועדף" או לא)

BlApi.ISale – אותן מתודות כמו DalApi.ISale

BlApi.IOrder – כאן יהיו המתודות לניהול ההזמנה תוך כדי ביצועה וכן ביצוע ההזמנה הסופי:

* AddProductToOrder - מתודה להוספת מוצר להזמנה – מקבלת כפרמטרים הזמנה (BO.Order), מזהה מוצר (int) וכמות להזמנה (int). מחזירה רשימת מבצעים שהתקבלו למוצר שנוסף (רשימה מסוג BO.Sale).
* CalcTotalPriceForProduct - מתודה לחישוב המחיר הסופי למוצר – מקבלת כפרמטר מוצר מסוג BO.ProductInSale ולא מחזירה ערך.
* CalcTotalPrice - מתודה לחישוב המחיר הסופי להזמנה – מקבלת כפרמטר הזמנה (BO.Order) ולא מחזירה ערך
* DoOrder - מתודה לביצוע הזמנה – מקבלת כפרמטר הזמנה (BO.Order) ולא מחזירה ערך.
* SearchSaleForProduct - עדכון המבצעים המתאימים למוצר בהזמנה

1. דוגמא אפשרית למימושים:

המימושים של שלושת הממשקים המקבילים לממשקים בשכבת DAL (IClient,ISale,IProduct) יחסית ברורים ולא מסובכים ולכן לא נפרטם כאן.

בגדול, הרעיון הוא לזמן את המתודה המתאימה בשכבת DAL כאשר יש צורך לבצע המרות מאובייקטים של BO לאובייקטים של DO וכן להיפך.

נפרט את המימוש של המתודות בממשק IOrder:

1. **SearchSaleForProduct** – עדכון המבצעים המתאימים למוצר בהזמנה  
   מקבלת כפרמטרים מוצר בהזמנה ופרמטר בוליאני – האם ההזמנה היא ללקוח קיים  
   לא מחזירה ערך.  
   מימוש הפונקציה:  
   מציבים לתכונה Sales של המוצר בהזמנה את תוצאת השאילתה הבאה:  
   כל המבצעים שמיועדים למוצר המבוקש, שהם בתוקף, ושהכמות בהזמנה הגיעה לכמות במבצע ואם ההזמנה היא לא עבור לקוח קיים, יש לבדוק גם שהמבצע מיועד לכל הלקוחות (ניתן לפצל לשתי שאילתות נפרדות)  
   הרשימה תמוין לפי מחיר ליחידה במבצע (כלומר, לפי כדאיות המבצע).  
   כמובן שיש להמיר את האובייקטים מסוג DO.Sale לסוג BO.SaleInProduct
2. **CalcTotalPriceForProduct** – חישוב המחיר לתשלום עבור מוצר בהזמנה, כולל מימוש מבצעים.  
   לא מחזירה ערך.  
   מימוש הפונקציה:  
   (אם לא קיימים מבצעים מתאימים למוצר זה (רשימת המבצעים של המוצר ריקה)  
   שולפים את המחיר של המוצר מהזיכרון (פניה ל DAL)  
   מציבים בשדה TotalPrice את מכפלת המחיר בכמות בהזמנה.  
   אם קיימים מבצעים למוצר –) – אפשר לדלג שומרים במשתנה (count) את הכמות בהזמנה.  
   עוברים על רשימת המבצעים של המוצר  
   אם count קטן מהכמות הנדרשת למבצע זה, ממשיכים למבצע הבא (continue)  
   אחרת, בודקים כמה פעמים ניתן לנצל את המבצע (חלוקה של count בכמות למבצע) ומכפילים במחיר של המבצע. את הסכום הזה מוסיפים למחיר הסופי.  
   מעדכנים את count – מורידים את הכמות שחישבנו במבצע זה (נשארת שארית החלוקה)  
   שומרים את המבצע שמומש ברשימת מבצעים מקומית בפונקציה.  
   אם count==0 – יוצאים מהלולאה (break)  
   אחרת, ממשיכים למבצע הבא.  
   בסיום הלולאה, מוסיפים למחיר הסופי את מכפלת המחיר הבסיסי של מוצר זה ב count (הכמות שנשארה ולא מומש עבורה מבצע)  
   מעדכנים את המבצעים של המוצר להיות המבצעים שהיו בשימוש  
   את שדה TotalPrice של המוצר מעדכנים להיות הסכום לתשלום שהצטבר.
3. **CalcTotalPrice** – חישוב הסכום הסופי לתשלום להזמנה.  
   מקבלת כפרמטר הזמנה ולא מחזירה ערך.  
   מימוש הפונקציה: מציבים בשדה TotalPrice של ההזמנה את הסכום של מחירי כל המוצרים.
4. **AddProductToOrder** – הוספת מוצר להזמנה.  
   מקבלת כפרמטרים: הזמנה, מזהה מוצר להוספה וכמות להוספה (יכולה להיות שלילית)  
   מחזירה רשימת מבצעים שמומשו עבור מוצר זה בהזמנה זו.  
   מימוש הפונקציה:  
   שולפים את המוצר הנבחר מהזיכרון (פניה ל DAL)  
   מנסים למצוא את המוצר הנבחר ברשימת המוצרים בהזמנה  
   אם קיים – בודקים שלאחר הוספת הכמות המבוקשת, יש מספיק במלאי מהמוצר ומעדכנים את הכמות בהזמנה. (אם אין מספיק במלאי – זורקים שגיאה מתאימה)  
   אם המוצר עדיין לא קיים בהזמנה, בודקים שיש במלאי את הכמות המבוקשת ומוסיפים מוצר חדש להזמנה (אם אין מספיק במלאי – זורקים שגיאה מתאימה)  
   בודקים מה המבצעים הקיימים למוצר זה (SearchSalesForProduct)  
   מחשבים את המחיר למוצר זה כולל מימוש המבצעים (CalcTotalPriceForProduct)  
   מחשבים את המחיר הסופי להזמנה (CalcTotalPrice)  
   מחזירים את רשימת המבצעים שמומשו.
5. **DoOrder** – מקבלת כפרמטר הזמנה ולא מחזירה ערך.  
   עבור כל מוצר בהזמנה, מייצרים בקשת עדכון ל DAL כדי להוריד ממלאי המוצר את הכמות שבהזמנה.