

מיני פרויקט במערכות חלונות - תרגיל מספר 2

מטרת התרגיל

יצירת שכבת לוגיקה בסיסית
עבודה עם רשימות
ממשקים (IComparable)
חריגות

הנחיות לביצוע התרגיל והגשתו

✓ העבודה תתבצע בזוגות

✓ חובה להשתמש בכלי לניהול גרסאות GIT

✓ יש להגיש במודל קישור על פי ההנחיות בקובץ "הגשת מטלות בקורס מיני פרויקט במערכות

חלונות 153007!"

✓ נא להקפיד על פורמט זה על מנת למנוע מצב של אי קבלת ציון על תרגיל מסוים

המטלה

הפרויקט השנה עוסק בניהול מערכת מידע של שירות משלוחים המתבצעים באמצעות רחפנים, המשלוחים מתבצעים בין הלקוחות השונים של השירות. למשל משלוחים מחנויות ועסקים שונים לצרכנים השונים.

החברה מחזיקה תחנות-בסיס עבור תחזוקת וטעינת הרחפנים וכן מידע על לקוחות וחבילות שנשלחו באמצעות השירות.

החברה המפעילה את שירות המשלוחים רוצה לעקוב אחר ביצוע המשלוחים וכן לנהל את המעקב על תחזוקת הרחפנים. בהמשך הפרויקט יהיה מעקב רחפנים וחבילות.

בתרגילים המקדימים נתחיל לבנות את הישויות בהן נשתמש במהלך הפרויקט.

בתרגיל 1 התחלנו לבנות את שכבת הנתונים של הפרויקט וכתבנו תכנית ראשית לבדיקת השכבה.

בתרגיל זה:

- נעדכן את שכבת הנתונים עם הכלים הנוספים שנלמדו בינתיים
 - הפיכת מערכים ל-List <> והתאמת הקוד שנכתב בתרגיל 1 בהתאם במידת הצורך.
 - הוספת בדיקות בסיסיות בשכבת הנתונים הבודקות את חוקיות הבקשות ברמת שלמות נתונים מקומית (בעיקר בדיקת ID) וזריקת חריגות במקרים של טעויות.
- נתחיל להגדיר את השכבה הלוגית
 - התחלת הגדרת ישויות לוגיות
 - התחלת בניית הלוגיקה של הפרויקט
 - טיפול בחריגות המתקבלות משכבת הנתונים וזריקת חריגות חלופיות
- נבנה תכנית ראשית (לקונסול) נוספת לבדיקת הפונקציונליות של השכבה הלוגית שנבנית בתרגיל הזה
 - בתוכנית הראשית יהיה תפריט דומה לזה שבתרגיל 1 בתוספת הפעולות המפורטות בהמשך

b. התכנית תטפל בחריגות המתקבלות מהשכבה הלוגית ותדפיס הודעות שגיאה בהתאם

הנחיות לביצוע התרגיל:

1. עדכוני שכבת הנתונים (שכתבתם בתרגיל 1)

- **בישוי**
 - רחפן - יש לבטל (למחוק) את התכונות של מצב סוללה ומצב רחפן (ראו בהמשך בשכבה הלוגית)
 - יש למחוק את אנומרציית (enum של) מצב הרחפן (שתועבר לשכבה לוגית)
- **ב-DataSource**
 - יש לשנות את כל המערכים לרשימות גנריות <List>
 - יש לעדכן את פעולת Initialize בהתאם
 - יש להוריד מ-Config את כל מצייני המקום הפנוי הראשון שהוגדרו עבור עבודה עם מערכים
 - יש למחוק את אתחולי מצב סוללה ומצב רחפן ברחפנים
- **ב-DalObject**
 - יש לעדכן את חתימות המתודות בהתאם לשינוי ב-DataSource
 - יש לשנות את המימושים כך שישמשו ברשימות בצורה יעילה ומתאימה, לפי סוג מתודה (הוספה, בקשה, עדכון, מחיקה - CRUD [Create, Request, Update, Delete])
 - עבור מתודה לעדכון סטטוס או שדה\שדות אחרים של אובייקט נתונים, ראו דוגמה בקישור <https://gitlab.com/-/snippets/2194904>
 - במתודות המחזירות רשימת אובייקטים, סוג ערך מוחזר בחתימות המתודות יהיה IEnumerable<T>. יחד עם זאת המתודות תחזירנה בפועל רשימות חדשות מסוג List<T>.
 - **במקרה של אי הצלחה של מתודת נתונים (ראו למטה התייחסות לנושא הלוגיקה) יש לזרוק חריגה שתוגדר לפי הר"מ (למשל אם בהוספת אובייקט מתגלה שכבר קיים אובייקט עם אותו מספר מזהה, או בבקשת עדכון לא נמצא האובייקט שנדרש לשנותו, וכדומה)**
- **ממשק IDal**
 - יש ליצור ממשק בשם IDal במודול נפרד בתוך הפרויקט DAL
 - הממשק יוגדר תחת מרחב שמות IDAL
 - הממשק יכלול את כל המתודות שיש ב-DalObject
 - יש להוסיף מתודת בקשת צריכת חשמל ע"י רחפן שתחזיר מערך של מספרים מטיפוס double
 - במימוש ב-DalObject יש להחזיר מערך של חמישה ערכים ע"פ הנתונים ממחלקת Config, לפי הסדר הבא: פנוי, קל, בינוני, כבד (ראו בהמשך) וקצב טעינה
 - מחלקה DalObject תממש את הממשק
- **חריגות**
 - יש להוסיף הגדרת חריגות בהתאם לסוגי התקלות בשכבת הנתונים כנ"ל - במודול אחד נפרד במרחב שמות IDAL.DO
- **מחלקת Config**
 - יש למחוק את כל המציינים של המקום הפנוי הראשון
 - יש להוסיף ארבע תכונות סטטיות עבור צריכת חשמל לק"מ ע"י רחפן פנוי
 - נושא משקל קל
 - נושא משקל בינוני

- נושא משקל כבד
- יש להוסיף תכונה עבור קצב טעינת רחפן (%) (בשעה)
- ב-ConsoleUI
 - יש לבצע שינויים בהתאם לדרישות הנ"ל
 - יש להשתמש בלולאת איטרטור (foreach) על מנת להדפיס רשימות
 - יש לשמור אובייקט של DalObject במשתנה או בשדה מסוג IDal
 - יצירת האובייקט הנ"ל כמובן ישאר בעזרת המחלקה DalObject
- במידה ועבור המימוש בשכבה הלוגית ייראה שיש צורך במתודה נוספת ב-DalObject או בשדה נוסף בישות נתונים - יש לבצע את השינויים ואת ההוספות הרלוונטיות תוך שמירה על העקרונות הבאים בעניין הקוד של מחלקת DalObject:
 - כל מתודה תהיה קשורה לנתוני ישות אחת ספציפית ולא תבצע פעולות על מספר ישויות
 - אסורה כל לוגיקה למעט הרשום כדלקמן:
 - בדיקת תקינות בקשה מבחינת ייחודיות או קיום מספר מזהה של ישות (למשל בהוספת אובייקט - בדיקה שלא קיים אובייקט עם אותו מספר מזהה)
 - חיפוש אובייקט שעליו התבקש לבצע פעולה
 - בניית רשימת אובייקטים לפי תנאי מסוים
 - אסורים קלט ופלט בהחלט
 - מומלץ מאוד לפצל מודול עבור DalObject לכמה מודולים על ידי שימוש במחלקה חלקית - כל מודול יכול מתודות הקשורות לישות מסוימת, ויש לשקף את שם הישות גם בשם המודול (למשל DalObjectParcel עבור מתודות הקשורות למשל לחבילות)
 - **נ.ב.** אם לא שמרתם על העקרונות אלו בתרגיל 1 - עליכם לעדכן את הקוד בהתאם

2. שכבה לוגית

- עבור השכבה הלוגית יש להוסיף פרויקט חדש בשם BL
- הפרויקט החדש יהיה מסוג ספריית מחלקות (Class Library)
- כל מה שקשור לשכבה הלוגית יהיה בפרויקט הזה
- ישויות לוגיות
 - כל ישות תוגדר במודול נפרד ושמו כשמה של הישות
 - כל הישויות תוגדרנה תחת מרחב שמות "IBL.BO".
 - כל ישויות לוגיות תוגדרנה כמחלקות
 - במודול אחד נפרד יש להגדיר את כל האנומרציות (enum) בצורה זהה לאנומרציות בשכבת הנתונים ובתוספת אנומרציות שתידרשנה למימוש שלכם של השכבה הלוגית (לכאן תועבר אנומרציית מצב הרחפן)
 - המתודה היחידה שמותר להגדיר - היא דריסת ToString בלבד
- חריגות
 - יש להוסיף הגדרת חריגות בהתאם לסוגי התקלות בשכבה לוגית - במודול אחד נפרד במרחב שמות IBL.BO
- ממשק IBL
 - יש ליצור ממשק בשם IBL, במודול נפרד תחת מרחב שמות IBL
 - הממשק יכלול את כל המתודות הדרושות למילוי בקשות התוכנית הראשית החדשה הר"מ
- מחלקה BL
 - המחלקה תוגדר במודלים נפרדים לפי קבוצות הפעולות כמחלקה חלקית - החלוקה ע"פ הגיון בריא ועקרון SRP
 - המחלקה תממש את הממשק IBL

- אובייקט BL יתחזק רשימת רחפנים (ע"פ הישות הלוגית "רחפן לרשימה")
- אתחול בבנאי מופע:
 - יש ליצור אובייקט של DalObject שיישמר בשדה מסוג IDal וישמש כנקודת גישה לזימון מתודות שכבת הנתונים
 - יש לבקש משכבת הנתונים ולשמור בשדות נפרדים את צריכת החשמל ע"י הרחפנים ואת קצב טעינתם - בהתאם למה שרשום לעיל
 - יש לבקש את רשימת הרחפנים משכבת הנתונים
 - אם יש חבילה שתשוער לא סופקו אך הרחפן כבר שויך
 - מצב הרחפן יהיה כמבצע משלוח
 - מיקום הרחפן יהיה כדלקמן:
 - אם החבילה שויכה אך לא נאספה - מיקום יהיה בתחנה הקרובה לשולח
 - אם החבילה נאספה אך עוד לא סופקה - מיקום הרחפן יהיה במיקום השולח
 - מצב סוללה יוגרל בין טעינה מינימלית שתאפשר לרחפן לבצע את המשלוח ולהגיע לטעינה לתחנה הקרובה ליעד המשלוח לבין טעינה מלאה
 - אם הרחפן לא מבצע משלוח
 - מצבו יוגרל בין תחזוקה לפנוי
 - אם הרחפן בתחזוקה
 - מיקומו יוגרל בין תחנות התחנות הקיימות
 - מצב סוללה יוגרל בין 0% ל-20%
 - אם הרחפן פנוי
 - מיקומו יוגרל בין לקוחות שיש חבילות שסופקו להם
 - מצב סוללה יוגרל בין טעינה מינימלית שתאפשר לו להגיע לתחנה הקרובה לטעינה לבין טעינה מלאה
 - מיקום יוגרל בין התחנות הקיימות
- במתודות המחלקה אסור לבצע קלט או פלט
 - לכל אפשרות בתפריט של תכנית ראשית תוגדר מתודה נפרדת
 - מתודות המחלקה יקבלו פרמטרים עם הנתונים הדרושים הקשורים לישויות לוגיות, יבצעו את הלוגיקה הנדרשת תוך זימון מתודות שכבת הנתונים המתאימות
 - אחריות מתודות המחלקה בין היתר על יצירת אובייקטים של ישויות הנתונים על בסיס נתונים בישויות לוגיות (שיתקבלו מהתכנית הראשית החדשה) עבור פעולות הוספה או עדכון מלא
 - מתודות המחלקה אחראיות בין היתר על יצירת אובייקטים של ישויות לוגיות על בסיס נתונים בישויות הנתונים (שיתקבלו מזימון מתודות בשכבת הנתונים) - עבור פעולות בקשת נתונים
 - מתודות המחלקה יבצעו את כל הבדיקות הנדרשות על מנת לוודא שהפעולות המבוצעות תואמות למצב הנתונים (כגון שלא "יתסופק" משלוח חבילה שעוד לא שויכה ולא נאספה ע"י רחפן) - זאת אומרת: זוהי האחריות של השכבה הלוגית שהנתונים יהיו הגיוניים וסדר הפעולות המבוצעות יהיה תקין
 - מתודות המחלקה יתפסו את החריגות הנזרקות מתוך שכבת הנתונים ויטפלו בהן בהתאם לצורך והגיון בריא, ובמקרה הצורך תיזרק חריגה חדשה שתוגדר בישויות הלוגיות כנ"ל
 - מחלקות מתודות עזר
 - ניתן להוסיף מחלקות ומתודות עזר (שהרשאת הגישה שלהן איננה public)

○ מתודות עזר יכולות להיות במחלקה BL או במחלקות עזר שתגדירו במודלים נפרדים תוך שמירת עקרון SRP

3. תכנית ראשית

- עבור התכנית הראשית של תרגיל 2 יש ליצור פרויקט חדש בשם ConsoleUI_BL
- לפרויקט החדש יש להוסיף הפניה (Reference) לפרויקט BL (בלבד!)
- יש להפוך את הפרויקט החדש לפרויקט ההרצה
- בתכנית הראשית ניתן ומומלץ לפצל את התפריט למתודות נפרדות ע"פ תתי-תפריט וע"פ הגיון בריא
- ניתן ליצור מתודות עזר ואף מחלקות עזר
- יש ליצור אובייקט מסוג BL ולשומרו במשתנה או בשדה מסוג IBL בפרויקט הראשי, עבור כל אפשרות יתבצעו:
 - קלט (אם נדרשים נתונים עבור הבקשה)
 - זימון מתודה משכבה לוגית מתאימה
 - פלט התוצאות (בעיקר עבור בקשות נתונים) או פלט המייצג הצלחת ביצוע בקשה
- בתכנית הראשית אסור לבצע בדיקות לוגיות או בדיקות תקינות קלט למעט בדיקות תקינות פורמט קלט (כגון כשצריך מספר אבל הקלידו תווים שאין ספרות)
- התוכנית הראשית תתפוס את החריגות שנזרקו מהשכבה הלוגית עם פלט שמציג את הסיבה לתקלה ואפשר גם המלצות לתיקון

התוכנית הראשית תציג תפריט למשתמש עם האפשרויות הבאות:

נ.ב. התפקיד של כל אפשרות בתפריט הוא לבדוק מתודה אחת של IBL

• אפשרויות הוספה:

○ הוספת תחנת-בסיס

■ קלט:

• מספר תחנה (יכנס למספר מזהה)

• שם תחנה

• מיקום

• מספר עמדות טעינה (פנויות) - כל העמדות פנויות בהוספה

■ רשימת הרחפנים בטעינה תאותחל לרשימה ריקה

○ הוספת רחפן

■ קלט:

• מספר סידורי של היצרן (יכנס למספר מזהה)

• מודל הרחפן

• משקל מירבי (בחירת קטגוריה מתת-תפריט?)

• מספר תחנה לשים בה את הרחפן לטעינה ראשונית

■ BL:

• מצב סוללה יוגרל בין 20% ל-40%

• יוסף כנמצא בתחזוקה

• מיקום הרחפן יהיה כמיקום התחנה

○ קליטת לקוח חדש

■ קלט:

• מספר ת"ז של הלקוח (יכנס למספר מזהה)

• שם הלקוח

• מספר טלפון

• מיקום

○ קליטת חבילה למשלוח

■ קלט:

• לקוח שולח (ת"ז)

• לקוח מקבל (ת"ז)

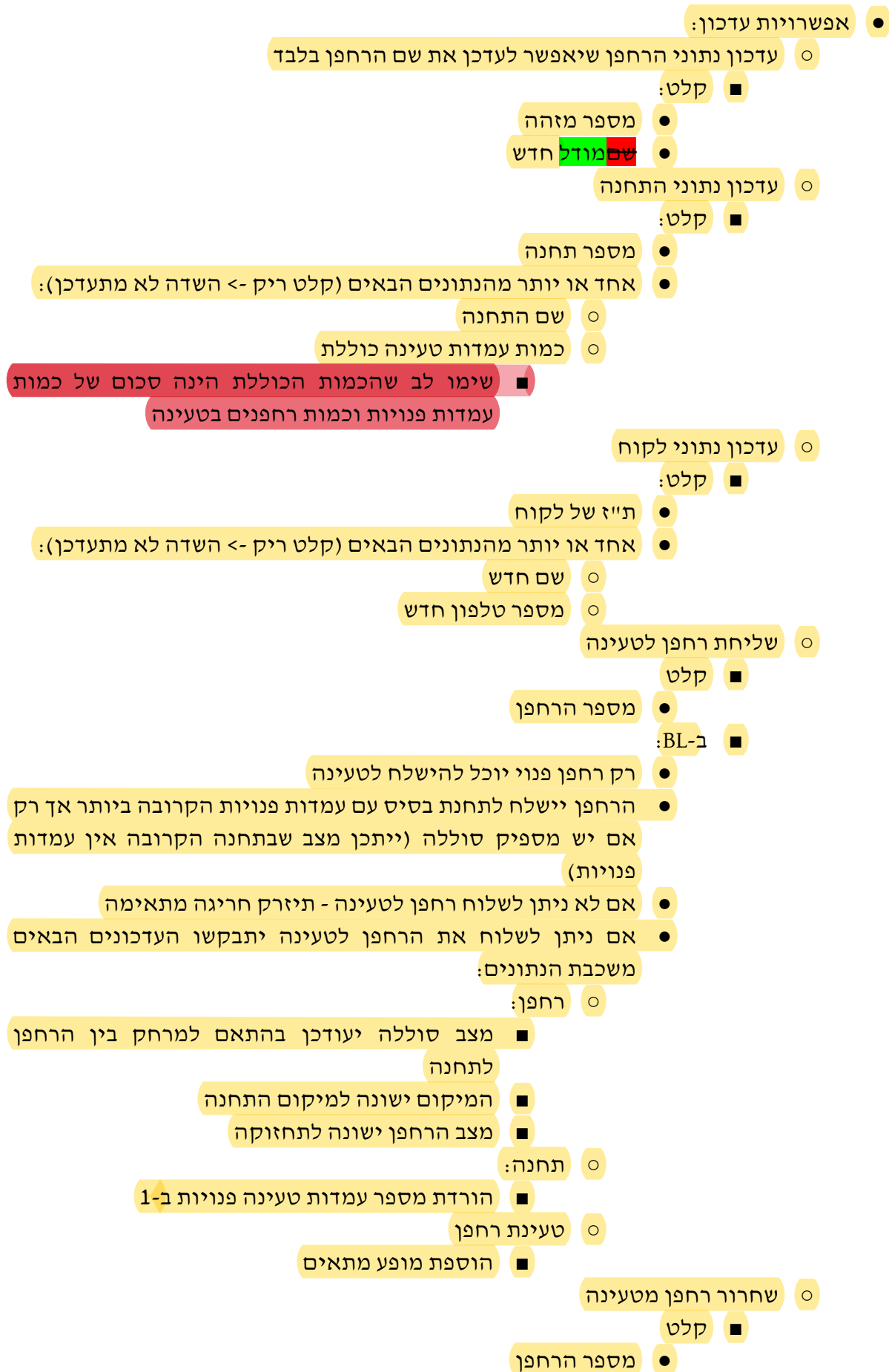
• משקל (בחירת קטגוריה מתת-תפריט?)

• עדיפות (בחירה מתת-תפריט?)

■ ב-BL כל הזמנים יאותחלו לזמן אפס למעט תאריך יצירה שיאותחל

ל-DateTime.Now

■ הרחפן יאותחל ל-null



- פרק זמן בטעינה
- ב-BL:
- רק רחפן בתחזוקה יוכל להשתחרר מטעינה
- אם לא ניתן לשחרר את הרחפן - תיזרק חריגה מתאימה
- אם הרחפן משתחרר יתבקשו העדכונים הבאים משכבת הנתונים:
- רחפן:
- מצב סוללה יעודכן בהתאם לזמן שהיה בטעינה
- מצב הרחפן ישונה לפני
- תחנה:
- העלאת מספר עמדות פנויות ב-1
- טעינת רחפן:
- מחיקת מופע מתאים
- נ.ב. מי שעשה לבנוס של תרגיל 1 שמירת היסטוריית טעינה - במקום מחיקה - עדכון נתונים מתאימים
- שיוך חבילה לרחפן
- קלט:
- מספר רחפן
- ב-BL:
- יש לוודא שהרחפן פנוי (כמובן שלפני כן יש לבקש את נתוני הרחפן משכבת הנתונים)
- יש לחפש חבילה לפי התנאים הבאים לפי הסדר הרשום:
 - בעדיפות הגבוה ביותר
 - משקל חבילה מרבי אפשרי עבור הרחפן
 - החבילה הקרובה
 - כל הנ"ל רק בתנאי שהרחפן יצליח להגיע לשולח, להעביר את החבילה ליעד ולהגיע לתחנה הקרובה ביותר (מיעד המשלוח) על מנת להיטען (אם יהיה צורך בכך)
 - נ.ב. שימו לב שיש מספר גישות לבנות את הלוגיקה הנ"ל, חלקן מסורבלות ולא יעילות, וחלקן יפות ויעילות יותר.
- אם נמצאה חבילה מתאימה, יש לעדכן את שכבת הנתונים כלדקמן:
 - יש לשנות את מצב הרחפן למבצע משלוח
 - בחבילה יש להוסיף את הרחפן ולעדכן את זמן השיוך
- איסוף חבילה ע"י רחפן
- קלט:
- מספר רחפן
- ב-BL:
- רק רחפן המבצע משלוח של חבילה ששויכה אליו אבל עוד לא נאספה יוכל לאסוף אותו
- במקרה שאי אפשר לאסוף את החבילה תיזרק חריגה מתאימה
- אם החבילה נאספת, יש לעדכן את שכבת הנתונים כלדקמן:
- רחפן:

- עדכון מצב סוללה לפי המרחק בין מיקום מקורי לבין מיקום השולח
- עדכון מיקום למיקום השולח
 - חבילה:
 - עדכון זמן איסוף
 - אספקת חבילה ע"י רחפן
 - קלט:
 - מספר רחפן
 - ב-BL:
 - רק רחפן שאסף אך עוד לא סיפק את החבילה יוכל לספק אותה
 - במקרה שאי אפשר לספק את החבילה תיזרק חריגה מתאימה
 - אם החבילה מסופקת, יש לעדכן את שכבת הנתונים כדלקמן:
 - רחפן:
- עדכון מצב סוללה לפי המרחק בין מיקום מקורי לבין מיקום יעד המשלוח
- עדכון מיקום למיקום יעד המשלוח
- שינוי מצב רחפן לפנוי
 - חבילה:
 - עדכון זמן אספקה
- אפשרויות תצוגה (כולן ע"פ מספר מזהה מתאים, תצוגה של כל הנתונים מישות לוגית, בעזרת ToString של ישות לוגית):
 - תצוגת תחנת-בסיס
 - שימו לב שב-BL יש לאכלס את רשימת הרחפנים בטעינה בתחנה
 - תצוגת רחפן
 - שימו לב שב-BL יש לאכלס גם את החבילה שבהעברה (במקרה שהרחפן במצב ביצוע משלוח)
 - תצוגת לקוח
 - שימו לב שב-BL יש לאכלס את רשימת החבילות שהלקוח שולח ואת רשימת החבילות שהלקוח מקבל
 - תצוגת חבילה
 - שימו לב שב-BL יש לאכלס את הרחפן, אם החבילה כבר שויכה
- אפשרויות הצגת הרשימות
 - **נ.ב. הרשימות מוצגות ע"י רשימות הכוללות את מופעי הישויות לפי "יישות לרשימה" מתאימה**
 - הצגת רשימת תחנות-בסיס
 - הצגת רשימת הרחפנים
 - הצגת רשימת הלקוחות (תצוגה מקוצרת - מספר מזהה, שם, ומיקום)
 - הצגת רשימת החבילות
 - הצגת רשימת חבילות שעוד לא שויכו לרחפן
 - הצגת תחנות-בסיס עם עמדות טעינה פנויות
- יציאה

בהצלחה!

ישויות לוגיות:

יש להדגיש שהאובייקטים לא נשמרים (למעט הרחפנים) אלא מחושבים בעת הצורך

• מיקום - יש להדגיש שבכל הישויות שזה נדרש קיימת תכונה שמחזיקה אובייקט מסוג זה

0 קו אורך (longitude)

0 קו רוחב (latitude)

• לקוח

0 מספר מזהה ייחודי

0 שם הלקוח

0 מספר טלפון

0 מיקום

0 רשימת משלוחים **חבילות** **אצל לקוח** - מהלקוח

0 רשימת משלוחים **חבילות** **אצל לקוח** - אל הלקוח

• לקוח לרשימה

0 מספר מזהה ייחודי

0 שם הלקוח

0 טלפון

0 מספר חבילות ששלח וסופקו

0 מספר חבילות ששלח אך עוד לא סופקו

0 מספר חבילות שקיבל

0 מספר חבילות שבדרך אל הלקוח

• לקוח במשלוח **חבילה**

0 מספר מזהה ייחודי

0 שם הלקוח

• חבילה

0 מספר מזהה ייחודי

0 **לקוח בחבילה** (השולח)

0 **לקוח בחבילה** (המקבל)

0 קטגוריית משקל (קל, ביניים, כבד)

0 עדיפות (רגילה, מהיר, חירום)

0 **רחפן בחבילה**

0 זמן יצירת המשלוח **חבילה**

0 זמן שיוך

0 זמן איסוף

0 זמן אספקה

• חבילה לרשימה

0 מספר מזהה ייחודי

0 שם לקוח שולח

0 שם לקוח מקבל

0 קטגוריית משקל (קל, ביניים, כבד)

0 עדיפות (רגילה, מהיר, חירום)

ס מצב חבילה (הוגדרה, שויכה, נאספה, סופקה)

• **משלוח חבילה אצל לקוח**

ס מספר מזהה ייחודי

ס קטגוריית משקל (קל, ביניים, כבד)

ס עדיפות (רגילה, מהיר, חירום)

ס מצב (נוצר, שויך, נאסף, סופק)

ס **לקוח במשלוח חבילה** - המקור\היעד (הצד השני של המשלוח חבילה) - המקבל עבור השולח

והשולח עבור המקבל)

• **חבילה בהעברה**

ס מספר מזהה ייחודי

ס מצב **משלוח חבילה** (בולאני - ממתין לאיסוף \ בדרך ליעד)

ס עדיפות (רגילה, מהיר, חירום)

ס קטגוריית משקל (קל, ביניים, כבד)

ס **לקוח במשלוח חבילה** - השולח

ס **לקוח במשלוח חבילה** - המקבל

ס **מיקום** איסוף

ס **מיקום** יעד אספקה

ס מרחק הובלה

• **רחפן**

ס מספר מזהה ייחודי

ס מודל הרחפן

ס קטגוריית משקל (קל, ביניים, כבד)

ס מצב סוללה

ס מצב רחפן

ס **משלוח חבילה בהעברה**

ס **מיקום** (נוכחי)

• **רחפן לרשימה**

ס מספר מזהה ייחודי

ס מודל הרחפן

ס קטגוריית משקל (קל, ביניים, כבד)

ס מצב סוללה

ס מצב רחפן

ס **מיקום** (נוכחי)

ס מספר חבילה מועברת (אם יש)

• **רחפן בחבילה**

ס מספר מזהה ייחודי

ס מצב סוללה

ס **מיקום** (נוכחי)

• **רחפן בטעינה**

ס מספר מזהה ייחודי

ס מצב סוללה

• תחנת-בסיס

מספר מזהה ייחודי	0
שם התחנה	0
<u>מיקום</u>	0
מספר עמדות הטענה (פנויות)	0
רשימת <u>רחפנים בטעינה</u>	0

• תחנת-בסיס לרשימה

מספר מזהה ייחודי	0
שם התחנה	0
מספר עמדות טעינה פנויות	0
מספר עמדות טעינה תפוסות	0