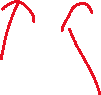
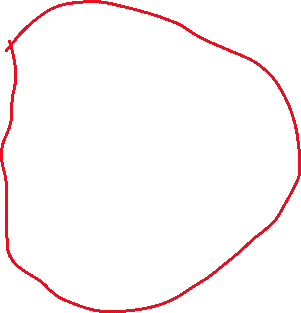
**לעבור על מוסכמות שמות ב C#**

**יום שלישי- כ"ט כסליו**

**עבודה על פרויקטים DalObject ו ApiDal**



LineStation

LineNumber..

numberInPath..

BusLine

Number…

Erea….

Path..

הטיפוסים ב DO:

אוטובוס

אוטובוס בנסיעה (רשות- לשלב הבא)

תחנת אוטובוס

קו אוטובוס

יציאת קו (רשות- לשלב הבא)

זוג תחנות עוקבות

תחנת הקו

משתמש (רשות)

נסיעת משתמש (רשות- בונוס לשלב הבא)

פעולות idal:

CRUD עבור כל יישות:

* הוספה
* עדכון
* מחיקה
* בקשה
* אובייקט בודד
* אוסף כולו
* אוסף לפי תנאי סינון

**IDal פירוט הפונקציות בממשק:**

**הוספה: יקבלו את הטיפוס מאותו הסוג אותו מוסיפים יחזירו void**

* הוספת אוטובוס (טכנית) (2 בדיקות- אם האוטובוס קיים זורק חריגה, אם האוטובוס היה קיים ונמחק זורק חריגה)
* הוספת תחנת אוטובוס (2 בדיקות- אם התחנה קיימת זורק חריגה, אם התחנה הייתה קיימת ונמחקה זורק חריגה)
* הוספת קו אוטובוס
* הוספת תחנה בקו מסוים (לבדוק שלא קיימת תחנה אחרת במקום הזה במסלול של האוטובוס?)
* הוספה לשתי תחנות עוקבות (לבדוק שלא קיימים כבר נתונים)
* הוספת משתמש (בדיקה- שם משתמש זהה לא קיים במערכת)

**עדכון: יקבלו את הטיפוס מאותו הסוג אותו מוסיפים יחזירו void**

* עדכון אוטובוס (טכנית) (2 בדיקות- אם האוטובוס לא קיים זורק חריגה, אם האוטובוס היה קיים ונמחק זורק חריגה)
* עדכון תחנת אוטובוס (2 בדיקות- אם התחנה לא קיימת זורק חריגה, אם התחנה הייתה קיימת ונמחקה זורק חריגה)
* עדכון קו אוטובוס (אם הקו לא קיים- זורק חריגה)
* עדכון תחנה בקו מסוים (נשאר קבוע- מספר קו, מספר תחנה במסלול) (לבדוק אם התחנה לא קיימת (מקום במסלול)– זורק חריגה)
* הוספה לשתי תחנות עוקבות (לבדוק שקיים מידע על 2 התחנות)
* עדכון משתמש (עדכון סיסמא) (בדיקה- שם משתמש זהה קיים במערכת)

**מחיקה: יקבלו את המספר המזהה מאותו הסוג אותו מוסיפים יחזירו void**

* מחיקה אוטובוס (טכנית) (2 בדיקות- אם האוטובוס לא קיים זורק חריגה, אם האוטובוס היה קיים ונמחק זורק חריגה)- עדכון deleted
* מחיקה תחנת אוטובוס (2 בדיקות- אם התחנה לא קיימת זורק חריגה, אם התחנה הייתה קיימת ונמחקה זורק חריגה)- עדכון deleted
* עדכון קו אוטובוס (אם הקו לא קיים- זורק חריגה)- עדכון deleted
* עדכון תחנה בקו מסוים (נשאר קבוע- מספר קו, מספר תחנה במסלול) (לבדוק אם התחנה לא קיימת (מקום במסלול)– זורק חריגה) - עדכון deleted
* הוספה לשתי תחנות עוקבות (לבדוק שקיים מידע על 2 התחנות)- עדכון deleted
* עדכון משתמש (עדכון סיסמא) (בדיקה- שם משתמש זהה קיים במערכת)- עדכון deleted

**בקשה : יקבלו לפי סוג הבקשה מחזירים את סוג הטיפוס המבוקש (IEnamrable במקרה של אוספים)**

**אובייקט בודד: יקבלו את המספר המזהה, יחזירו איבר בודד מהטיפוס המבוקש**

* בקשת אוטובוס (טכנית) (2 בדיקות- אם האוטובוס לא קיים זורק חריגה, אם האוטובוס היה קיים ונמחק זורק חריגה)- מחזיר Bus
* בקשת תחנת אוטובוס (2 בדיקות- אם התחנה לא קיימת זורק חריגה, אם התחנה הייתה קיימת ונמחקה זורק חריגה)
* בקשת קו אוטובוס (אם הקו לא קיים- זורק חריגה)
* בקשת תחנה בקו מסוים (נשאר קבוע- מספר קו, מספר תחנה במסלול) (לבדוק אם התחנה לא קיימת (מקום במסלול)– זורק חריגה) -
* בקשת שתי תחנות עוקבות (לבדוק שקיים מידע על 2 התחנות)-
* בקשת משתמש (בדיקה- שם משתמש זהה קיים במערכת)

**אוסף שלם: לא יקבלו כלום, יחזירו IEnamerable של הטיפוס המבוקש**

**לשים לב שמזירים רק את אלו שלא deleted!**

* בקשת רשימת אוטובוסים (טכנית)
* בקשת רשימת תחנות אוטובוס
* בקשת רשימת קווי אוטובוס
* בקשת רשימת תחנות בקווים
* בקשת רשימת כל התחנות העוקבות
* בקשת רשימת המשתמש (בדיקה- שם משתמש זהה קיים במערכת)

**אוסף לפי תנאי מסויים: לקבל פרדיקט (פונקציה מסננת), יחזירו IEnamerable של הטיפוס המבוקש (לאחר הסינון).**

* סינון על רשימת אוטובוסים (טכנית)
* סינון רשימת תחנות אוטובוס
* סינון רשימת קווי אוטובוס
* סינון רשימת תחנות בקווים
* סינון רשימת כל התחנות העוקבות
* סינון רשימת המשתמש (בדיקה- שם משתמש זהה קיים במערכת)

צעדים הבאים:

1. להתעסק עם הds לאתחל נתונים בתוך הרשימות.
2. לפתוח פרויקט חדש זמני. Console. מתקשר לDS וגם ל dalApi.

וככה כל פעם שנכתוב פונקציה נוכל לבדוק אותה ספציפית עם הconsole ולפי הגישה לנתונים..

**יום רביעי א' טבת**

**שכבת ה BL:**

**ישויות ב BO:**

* **קו אוטובוס:**
* **נתונים בסיסיים- מס' מזהה, מספר קו.**
* **אוסף התחנות במסלול של הקו: כל תחנה כוללת את המרחק מהתחנה הקודמת**
* **תחנת אוטובוס:**
* **נתונים בסיסיים- קוד תחנה, שם תחנה.**
* **רשימת הקווים שעוברים בתחנה כל קו כולל את מזהה הקו ואת מספר הקו ואת תחנת הסיום שלו?**
* **תחנה במסלול של קו:**
* **קוד תחנה**
* **שם תחנה**
* **מרחק מהתחנה הקודמת**
* **זמן נסיעה מהתחנה הקודמת**
* **אוטובוס בודד:**
* **נתונים בסיסיים- מס' רישוי**
* **נתונים מה Dal**

**פעולות ב IBl: (לפי התצוגה שנרצה להציג למשתמש)**

* **הוספת אוטובוס**
* **מחיקת אוטובוס**
* **עדכון פרטים של אוטובוס**
* **הוספת קו אוטובוס**
* **מחיקת קו אוטובוס**
* **עדכון פרטים של קו אוטובוס**
* **עדכון תחנות קו של האוטובוס**
* **הוספת תחנות קו לאוטובוס (מספר תחנה שעדיין לא קיים)**
* **מחיקת תחנה מסוימת מקו אוטובוס- עדכון של כל התחנות אחריה לשנות מיקום..:(**
* **הוספה של פרטים של 2 תחנות שונות**
* **הוספת תחנת אוטובוס**
* **מחיקת תחנת אוטובוס**
* **עדכון פרטים של תחנת אוטובוס**
* **טיפול באוטובוס ?**
* **תדלוק של אוטובוס**
* **שליחת פרטים בסיסיים של כל האוטובוסים (רשימה של טיפוס BO)**
* **שליחת פרטים של אוטובוס בודד**
* **שליחת פרטים בסיסיים של כל הקווים (רשימה של טיפוס BO)**
* **שליחת פרטים של קו בודד**
* **שליחת פרטים בסיסיים של כל התחנות (רשימה של טיפוס BO)**
* **שליחת פרטים של תחנה בודדת**
* **עדכון למרחק/זמן בין 2 תחנות**

DS – האוספים..

DAL – השכבה שניגשת לאוסף

הוספה, מחיקה, בקשה, עדכון

BL – השכבה הלוגית

כל החישובים והפעולות של המשתמש על האוסף

PL-UIתצוגה- חלונות..

מה שהמשתמש רואה

לפני שעוברים ל wpf נשאר:

* לאתחל את כל הנתונים ב DS עם נתונים עדכניים ונכונים (להיעזר במוביט..) או באתר של משרד הרישוי
* לסיים פונקציות בBL ולהוסיף פונקציות לפי הנדרש
* להוסיף tostring לכל המחלקות (ב BO)
* לעשות בדיקה שהכל עובד כמו שצריך בBL ובDL
* להוסיף תיעוד לכל הפונקציות ב IBL

**יציאת קווים לנסיעה:**

ישויות חדשות ב BL:

1. **לוח זמנים קו אוטובוס:**

* רשימה של כל זמני היציאה של הקו )מהתחנה הראשונה) עבור יממה.
* רשימה של זמני נסיעה בין התחנות (time span)
* מזהה הקו

1. **יציאת קו אוטובוס: (מסוים)**

* מזהה הקו
* שעת יציאת האוטובוס
* רשימת תחנות עם זמני הגעה (מדויקים) לכל תחנה
* זמן נסיעה כולל (מהתחנה הראשונה ועד האחרונה)