

## תרגיל mo' 3 – מערכת לניהול מוסך

### מטרות

- הטעמה של עבודה עם מחלקות, הורשה ופולימורפיזם
- שימוש ב- Collections
- שימוש enums
- פיתוח ושימוש ב- DLL (אסמבלי) חיצוני
- עבודה עם מספר פרויקטים
- עבודה עם Exceptions

### דע נדרש

- תיכנות מונחה עצמים תוך שימוש בהורשה ופולימורפיזם ב- #C#
- שימוש ב- Collections
- פיתוח ושימוש ב- DLL (אסמבלי) חיצוני
- עבודה עם מספר פרויקטים
- עבודה עם Exceptions
- הירשה מ- Object

### התרגיל

המטרה הסופית: מערכת קטנה ש"מנהלת" מוסך.

המערכת תדוע לניהול מוסך שמתפל כולם בחמשה סוגים רכבים –

- אופננו רגלי
- (2 גלגולים עם לחץ אוויר מקסימאלי 29, דלק מסווג Octan 98, 6.8 ליטר טנק דלק)
- אופננו חשמלי
- (2 גלגולים עם לחץ אוויר מקסימאלי 29, זמן מצבר מקסימאלי – 2.6 שעות)
- מכונית רגילה
- (5 גלגולים עם לחץ אוויר מקסימאלי 33, דלק מסווג Octan 95, 4.7 ליטר טנק דלק)
- מכונית חשמלית
- (5 גלגולים עם לחץ אוויר מקסימאלי 33, זמן מצבר מקסימאלי – 4.2 שעות)
- משאית
- (12 גלגולים עם לחץ אוויר מקסימאלי 26, דלק מסווג Soler, 140 ליטר טנק דלק)

#### לכל כלי רכב יש את התכונות:

- שם דגם (string)
- מספר רישוי (string)
- אחוז האנרגיה שנותרה במקור האנרגיה שלו (בשביל מד הדלק/חשמל) (float)
- אוסף של גלגולים

#### לכל גלגל יש את התכונות הבאות:

- שם יצנן (string)
- לחץ אוויר נוכחי (float)
- לחץ אוויר מקסימאלי שקבע היצנן (float)
- פעולה ניוף (מתודת שמקבלת נתון לגבי כמה אוויר להוסיף לגלגל, ומשנה את מצב לחץ האויר אם הוא לא חורף מהמקסימום)

לאופנוו (רגיל/חשמי), בנוספ לתוכנות של רכב, יש גם את התוכנות:

- סוג רישו: B, A1, A2, AA,
- נפח מנווע בסמ"ק (int)

למכונית (רגילה/חשמיית) , בנוספ לתוכנות של רכב, יש גם את התוכנות:

- צבע (הצבעים האפשרים הם: כחול, ירוק, לבן, שחור)
- כמות דלקות (2, 3, 4, או 5)

למשאית, בנוספ לתוכנות של רכב, יש גם את התוכנות:

- האם מוביל חומרים מסוכנים
- נפח מטען - float

**בכל רכב שעבוד על דלק ניתן למצוא את המידע הבא ולהפעיל את הפעולות הבאות:**

- סוג הדלק (Soler, Octan95, Octan96, Octan98)
- כמות הדלק הנוכחית בליטרים (float)
- כמות הדלק המקסימלית בליטרים (float)

פעולת תלוק (מתודה שמקבלת כמות ליטרים להוסף וסוג דלק, ומשנה את מצב הדלק  
במידה וסוג הדלק מתאים ואין חריגה מגודל הטנק)

**בכל רכב חשמי ניתן למצוא את המידע הבא ולהפעיל את הפעולות הבאות:**

- זמן מצבר שנותר בשעות (float)
- זמן מצבר מקסימלי בשעות (float)
- פעולה טעינה מצבר (מתודה שמקבלת נתון שהוא מספר שעות להוסיף למצבר ו"טענת" את המctrber בהתאם כל עוד מספר השעות לא חורגת מהמקסימום)

**על כל רכב שנמצא במוסך יש לשמר את הנתונים הבאים:**

- שם בעליים (string)
- טלפון בעליים (string)
- מצב הרכב במוסך (המצבים האפשרים: בתיקון, תוקן, שולם)
  - כל רכב שנכנס למוסך מצבו ההתחלתי הוא 'בתיקון'.

### **המערכת תציג למשתמש תפריט עם האפשרויות הבאות:**

1. לטען את המערכת עם נתונים רכבים מתוך קובץ (מצ"ב קובץ עם נתונים רכבים – db)
2. "להכניס" רכב חדש למוסך -

המשתמש יתבקש להזין את מספר הרישוי של הרכב שמעוניינים להכניס למוסך.  
אם מדובר ברכב שכבר נמצא במוסך (עפ"י מספר רישוי), המערכת תוציא הודעה מתאימה  
ותשתמש ברכב שכבר נמצא במוסך (ותעביר את מצב הרכב לו - "בתיקון").  
אם לא -

המשתמש יתבקש לבחור את סוג הרכב שהוא מעוניין "להכניס למוסך", אז הוא יתבקש להזין את "מצבו", קרי את הנתונים שלו, קרי - כמות הדלק שיש בו או מצב המctrber (באחוזים), כמות האויר  
בגלגים, צבע (אם מדובר במכונית), סוג רישו (אם מדובר באופנוו), האם מוביל חומרים מסוכנים  
(אם מדובר במשאית), וכו'ב.

\* כדי להקל על המשתמש, ניתן לאפשר לו להזין מצב זהה לכל הגלגים בפעם אחת (במקום  
לבקש ממנו להזין את המצב של כל הגלגים בנפרד).

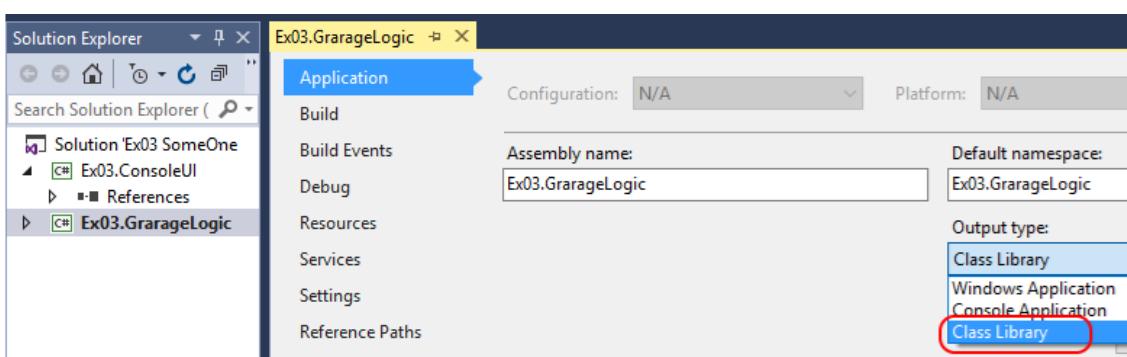
\* כדי להקל על המשתמש (למרות זהה 문제가 על המשתמש), ניתן למשמש באופן שהודעה על  
קלטיים לא תקין/נים תציג רק אחרי שהמשתמש הzin את כל הгалטים, ולא אחרי הקלט השגוי.

3. להציג את רשימת את מספרי הרישוי של הרכיבים ב מוצר, עם אפשרות לסינון לפי המצביעם ב מוצר.
4. לשנות מצב של רכב ב מוצר (הנתונים המבוקשים מהמשתמש הם מספר רישוי והמצביעב החדש).
5. לנפח אויר בಗלאלים של רכב למקסימום (המשתמש מזין מספר רישוי).
6. לתדלק רכב שמווע ע"י דלק (המשתמש מזין מספר רישוי, סוג דלק למילוי, כמות למילוי)
7. להטען רכב חשמלי (המשתמש מזין הנתונים הם מספר רישוי, שם דגם, שם בעלים, מצב במוצר, פירוט הגלגים (לחץ אויר ויצרן), מצב דלק + סוג דלק / מצב מצבר, ושאר הפרטים הרלוונטיים לסוג הרכב הספציפי)
8. להציג נתונים מלאים של רכב לפי מספר רישוי (מספר רישוי, שם דגם, שם בעליים, מצב במוצר, פירוט הגלגים (לחץ אויר ויצרן), מצב דלק + סוג דלק / מצב מצבר, ושאר הפרטים הרלוונטיים לסוג הרכב הספציפי)

לצורך פיתוח המערכת, יש ליצור **Solution** שיכיל **שני** פרויקטים:

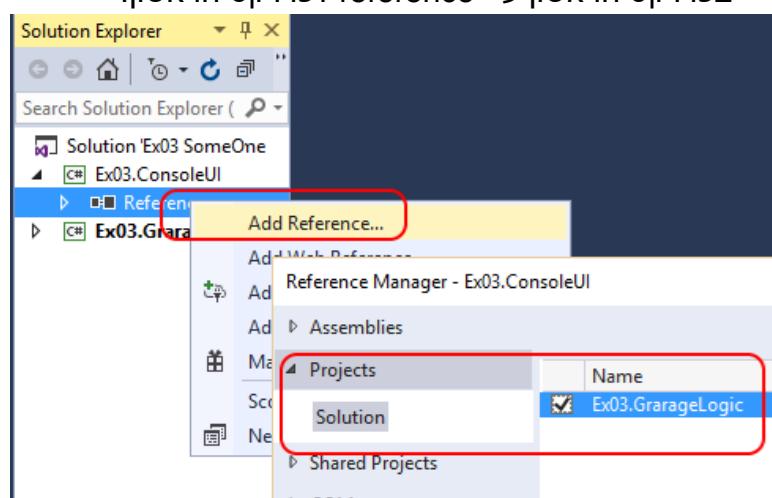
### Ex03.GarageLogic .1

זהו פרויקט שמייצר **dll** (אוסף של שאים PE) שמכיל רק את מודול האובייקטים והשכבה הלוגית של המערכת (השכבה שנרצה לעשות בה שימוש חוזרת גם אם נפתח ממשק משתמש אחר מסחר **Console**).  
השכבה זו אינה יודעת להדפיס למסך או לקלוט נתונים מקונסול.  
כדי ליצור **dll** יש לlectת למאפיינים (Properties) של הפרויקט ולבחר באפשרות:  
Output type: Class Library



### Ex03.ConsoleUI .2

זהו פרויקט שמייצר את ה- **exe** ובו יש שימוש של ממשק המשמש במערכת ניהול מוצר בקונסול. מערכת זו עשוה שימוש במודול שמומש בפרויקט הראשון ע"י reference לפרויקט הראשוני:



**הערות:**

1. יש לתכנן את המערכת בצורה הטובה ביותר כך מבחינת חלוקה למחקות ולמהותיות, הרושה ופולימורפיזם.
2. יש להפריד בין משק המשמש והשכבה הלוגית אשר מנהלת את מודל האובייקטים ואת הישויות הלוגיות של המערכת (מחלקה כמו Vehicle לא אמורה להכיר את המחלקה Console, גם לא באופן עקיף).
3. יש לנסות ולהשוו על פתרונות לצורך הפרדה זו.

**מחלקה שאחראית על ייצור אובייקטי רכבים במערכת, ועל רשימת הסוגים הנתמכים, סופקה**

**לכם חלק מהתרגיל (מצ"ב קובץ VehiclesCreator.cs). יש בה מתודה שמייצרת (new)**

**אובייקטים של מחלקות רכבים.**

- a. חובה להשתמש בה קוד
- b. אסור לשנות או להוסיף בה קוד
- c. אסור ליצור אובייקטים של רכבים בשום מקום אחר במערכת.
4. יש לכתוב את המערכת באופן זה שהוסףה של סוג רכב חדש ("טרקטור") תוכל להתבצע **לא שינוי בקוד** (למעט 2 התוספות המתבקשות במחלקה מסעיף 3 שמייצרת אובייקטים)
5. יש להימנע משכפל קוד/לוגיקה ומימוש בטכניות / פיצרים בשפה שלא הכרנו בקורס.
6. לצורך טיענת הרכבים מהקובץ יש להשתמש במתודה System.IO.File.ReadAllLines
7. אסור לשנות את הפורמט שבו מיוצגים הנתונים של הרכבים בקובץ.
8. **חובה להשתמש ב:**

Dictionary<K,V> List<T> ○

Enum ○

String formatting ○

ParseException (тиירק במקרה של קלט לא מתאים ברמת ה-Parsing).

ArgumentException (тиירק במקרה של קלט לא מתאים מבחינה לוגית, למשל בחירה למילוי

דלק מהסוג הלא נכון)

ValueOutOfRangeException – **מחלקה שאתם כתבו** (тиירק במקרה של קלט לא מתאים

מבחינה של טווח ערכים רצוי, למשל מילוי דלק מעיל המotor).

זהו מחלקה שיורשת מהמחלקה Exception ומחזיקה את השודות הנוספים הבאים:

float MaxValue ▪

float MinValue ▪

**הגשה:**

**יש לזרף להגשה בתוך התקינה את הקובץ (מצ"ב) אחרי שמלאתם בו:**

- שמות המגישים + ת"ז
- רשימת הטיפוסים העיקריים שהגדתם (class/enum) עם הסבר **קצרץ'** על כל אחד מהם.
- **Diagramma של מחלקות (Class Diagram)** שמציגת את היררכיות הירושה והקשר בין הרכבים העיקריים. לדוגמה

