

## תרגיל מס' 4 – ממשקים, משלחות, אירועים ותפריטים

### מטרות

- תכונות מונחה עצמים ו- polymorphism
- הטעמה של עובודה עם ממשקים - interfaces
- הטעמה של עובודה עם מצביעים לפונקציות ומשלחות Action<T> – (delegates)

### ידע נדרש

- עובודה עם ממשקים - interfaces
- עובודה עם משלחות – (Action<T>) delegates
- שימוש ב- Collections
- עובודה עם מספר פרויקטים

### התרגיל

עליכם למשרר רכיב תוכנה שמשיע בהציגו וניהול של **תפריטים היררכיים לאפליקציות Console**. אם היה זה רכיב, הייתם יכולים למשל להשתמש בו בתרגיל מס' 3 כדי להציג את התפריטים של המערכת למשתמש, במקומם לכתוב בעצמכם את הקוד שמציג אותם ומנהל את בחירות המשמש כפ' שבתח עשייתם בתרגילים רבים בעבר..).

עליכם למשרר מחלוקת בשם **MainMenu** אשר תאפשר (لتוכניתן שישתמש בה) לבנות תפריט לפי צרכי (למשל במערכת לניהול מוסך שמימושם בתרגיל הקודם), ע"י הגדרה של פריטי-תפריט בתפריט הראשי או לכל תת-תפריט.

אפליקציה שרצה להציג תפריטים למשתמש (למשל האפליקציה לניהול מוסך מהתרגיל הקודם) צריכה להזין מופיע של מחלוקת **MainMenu**.

מחלקה **MainMenu** תאפשר למשתמש בה (למשל התוכניתן שבונה את האפליקציה של המוסך) **לבנות את היררכיית התפריטים** (נניח בשלב האתחול באפליקציה שלו) ע"י הגדרת והוספה **פריטים** (מוספים של מחלוקת בשם **MenuItem**) לתפריט ותת-פריטים לכל פריט וכן הלאה..

כודגמא לתפריט עם פריטי תפריט שבוני בצורה היררכית, קחו למשל את התפריט של visual studio – התפריט הראשי נמצא בראש החלון ומכיל מספר פריטים ברמה הראשונה (File, Edit, View, ..) שמכילים פריטים בעצמם. לחיצה על חלק מהפריטים גורמת לפעולה מסוימת בתוכנה. לחיצה על פריטים אחרים גורמת להופעה של תת-תפריט (פריטי תפריט נוספים שמאוגדים מתחת לפריט שבחרנו) וכן הלאה..

המשך בעמוד הבא..

**אפיון**

כאמור, עליוכם למשוך רכיב תוכנה שמספק הצגה וניהול של תפריטים היררכיים לאפליקציות Console. הפעלת הפונקציה () `MainMenu.Show` תגרום להציג התפריט הראשי (רימה ראשונה) ובפועל תקיים את הלולאה הריאטיבית שכל אפליקציית console זקוקה לה כדי להתקיים. בכל שלב:

1. תוצג הרמה הנוכחית של התפריט למשתמש
2. תוצג בקשה למשתמש לבחור באחד הפריטים.
3. תקלט הבחירה מהמשתמש
4. תבצעו בדיקת תקינות קלט + הצגת הודעה מתאימה במקרה של קלט לא תקין
5. ניוט לחת-תפריט / הפעלת הפעולה שנבחרה:
  - a. בחירה בפריט שמקיל פריטי משנה תנקה את המסר ותציג את פריטי המשנה הרלוונטיים (כלומר את הרמה הבאה בתפריט, כלומר חזרה לשלב 1).
  - b. בחירה בפריט שאינו מכיל פריטי משנה, כלומר זו בחירה שבגינה צריכה ל��ות פעולה כלשהיא במערכת (למשל – האפשרות להכנס רכב חדש למוסך) תנקה את המסר ותפעיל את הפונקציה במערכת שמספקת את משך המשתמש שתומך בפונקציונליות הצעת של המערכת. לאחר השלמת הפעולה במערכת, תוצג שוב הרמה של התפריט שבה היינו קודם וחוזר חלילה עד לרגע שהמשתמש יבחר באופציית יציאה מהתוכנית (ע"י פריט מיוחד בתפריט)

שימוש לב: התפריט מציג באופן קבוע בסדר זהה (בכל רמה שהיא):

- כותרת (ברמה ראשונה - כותרת ראשית, בכל רמה אחרת – כותרת הפריט שנבחר ברמה הקודמת)
- רשימת פריטים שישיכם לרמה הנוכחיית (מספרים מ-1)
- פריט שכותוב בו 'Back' או 'Exit' במידה וזה התפריט הראשי (מספר 0)
- הודעה שմבקשת מהמשתמש לבחור מבין הפריטים או לבחור בפריט 'Back'

```
** Drinks Machine **  
-----  
1. Tea  
2. Coffee  
3. Juice  
0. Exit  
Please enter your choice (1-3 or 0 to exit):  
>> 2
```

**<Console.Clear()>**

```
** Coffee **  
-----  
1. Americano  
2. Espresso  
3. Capuchino  
4. Late  
0. Back  
Please enter your choice (1-4 or 0 to go back):  
>> 3
```

המערכת שמשתמשת במחלקה `MainMenu` תציג כתעת למשתמש את משך המשתמש הרלוונטי להכנס אספרסו, ובסיום הפעולה תציג שוב הרמה הנוכחית בתפריט:

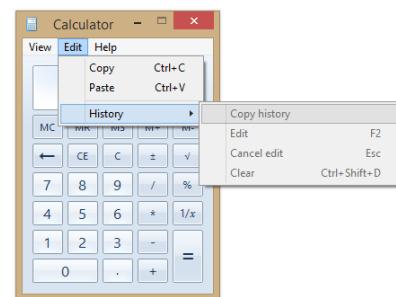
## הסברים נוספים:

הרכיב מאפשר למי משתמש בו (מתקנת אחר, שבונה למשל מערכת לניהול מוסך לקונסול...) לבנות תפריט לפי צרכי, עם תת-תפריטים בהתאם, ולהשתמש בו במקום למש את התפריט למשתמש עצמו.

הדרך לעשות זאת היא ע"י כך שהרכיב יחזק אוסף של פריטי תפריט, וכל פריט יחזק אוסף של תת-פריטים במקרה הצורך (שייהו את תת-התפריט שלו) וכן הלאה.  
במערכת משתמשת במנגנון התפריטים זהה (למשל המערכת לניהול מוסך), צריכה להיות פונקציה שמאתחלת את מבנה התפריט שלה.

לגביה הפונקציות שעל הפריטים הסופיים ("עלים") להפעיל - צריכה אכן להיות דרך לאפשר להם להפעיל את הפונקציה המתאימה באפליקציה (פריט סופי = פריט שאין לו תת-פריטים, כלומר פריט שמטרתו להפעיל פעולה מסוימת ולא להציג תת-תפריט).

קחו לדוגמה את התפריט המוצע במחשבון (במקרה הזה: 3 פריטים בתפריט, 3 פריטים בפריט השני, 4 פריטים בפריט השלישי שלו...):



## הוראות לימוש

עליכם למשך זאת בשתי הטכניקות שהוסבו לפתרון בעית "עובד מודיע שהוא חולה"  
(רמז – במקרה זהה, פריט בתפריט מודיע למערכת שבחרו בו)

### 1. ע"י שימוש בממשקים – interface

איזה משק יש להגדיר? אילו פעולה/ות הוא יחשוף? היכן (באיזה אוסף) יש להגדיר את המשק? מי אמור למשך את המשק? מי אמור להשתמש במשק? מה צריכה להיות רמת החשיפה שלו (access modifier)? כיצד תקבל המחלקה משתמש במשק את האובייקט שמשמש את המשק? וכו'.

### 2. ע"י שימוש ב- Action

מי יציג מופיע שלו? מי יפעיל אותו? מי מייצר מופיע שלו? וכו'.

פתרון התרגיל יכולsolution ובו 3 פרויקטים:

#### 1. Ex04.Menus.Interfaces – יכול מימוש לפי הטכניקה הראשונה.

זהו אינו פרויקט שמייצר קובץ הרצה אלא סדרית מחלקות בלבד (.NET Assembly).

#### 2. Ex04.Menus.Events – יכול מימוש לפי הטכניקה השנייה.

זהו אינו פרויקט שמייצר קובץ הרצה אלא סדרית מחלקות בלבד (.NET Assembly).

#### 3. Ex04.Menus.Test – אפליקציה (EXE) שמדגימה שימוש בכל אחד משני הפרויקטים מהסעיפים הקודמים:

בממודה Main יש לבנות שני תפריטים בעלי 2 רמות כל אחד (אחד עbor כל אחת משתי השיטות).  
תחילה יציג התפריט הראשון והמשתמש יוכל לנוט בו. כשהמשתמש יבחר ליצאת מתפריט (Exit)  
יוצג לו התפריט השני.

### להלן המפורט של התפריטים:

ברמה הראשונה יהיו 2 פריטים.

#### a. פריט 1 יציג תת-תפריט עם הכותרת – "Version and Lowercase" שבו יהיו 2 פריטים:

##### i. פריט פועלה – "Show Version"

מפעיל (בעקיפין) מותודה ב- Test שמציגה את הטקסט הבא:

App Version: 26.1.4.5940

##### ii. פריט פועלה – "Count Lowercase"

מפעיל (בעקיפין) מותודה ב- Test שմבוקשת מהמשתמש לכתוב משפט

והמערכת אומרת לו כמה אותיות קטנות (lower-case) יש במשפט.

#### b. פריט 1 יציג תת-תפריט עם הכותרת "Show Current Date/Time" שבו יהיו 2 פריטים:

##### i. הפריט הראשון יהיה פריט פועלה – "Show Current Time"

מפעיל (בעקיפין) מותודה ב- Test שמציגה את השעה נוכחית.

##### ii. הפריט השני יהיה פריט פועלה – "Show Current Date"

מפעיל (בעקיפין) מותודה ב- Test שמציגה את התאריך של היום.

שים לב, אין צורך להימנע משיכפול קוד בין שני הפרויקטים הראשונים.

נא להתייחס אליהם כאלו שני תרגילים שונים, העשויים להכיל קטעי קוד דומים.

**פלט לדוגמא:**

```
** Delegates Main Menu **  
-----  
1. Version and Lowercase  
2. Show Current Date/Time  
0. Exit  
Please enter your choice (1-2 or 0 to exit):  
>> 1  
  
<clear screen>  
  
** Version and Lowercase **  
-----  
1. Show Version  
2. Count Lowercase  
0. Back  
Please enter your choice (1-2 or 0 to go back):  
>> 1  
App Version: 26.1.4.5940  
  
** Version and Lowercase **  
-----  
1. Show Version  
2. Count Lowercase  
0. Back  
Please enter your choice (1-2 or 0 to go back):  
>> 1  
I Wonder if 3 cats are Better than 4 dogs  
> There are 27 lowercase letters in your text  
  
** Version and Lowercase **  
-----  
1. Show Version  
2. Count Lowercase  
0. Back  
Please enter your choice (1-2 or 0 to go back):  
>> 0  
  
<clear screen>  
  
** Delegates Main Menu **  
-----  
1. Version and Lowercase  
2. Show Current Date/Time  
0. Exit  
Please enter your choice (1-2 or 0 to exit):  
>> 1  
  
** Show Current Date/Time **  
-----  
1. Show Current Time  
2. Show Current Date  
0. Back  
Please enter your choice (1-2 or 0 to go back):  
>> 1  
> Current Time is 18:45:32
```

## הערות נוספת

1. כותרות בתפריט יש להציג בצבע סגול. לצורך כך, ניתן להשתמש ב- `Console.ForegroundColor`.
2. יש לתכנן את המערכת בצורה הטובה ביותר של קוד מבנית חלוקה למחלקות ולמетодות, הורשה ופולימורפיזם. יש להימנע משכפל קוד/לוגיקה ע"י חלוקה למתחדות והפרדה של לוגיקה לרכיבים. שימוש לבב, אלו הם רכיבים שמשמשים מערכת תפריטים **לקונסול**. لكن מן הסתם מותר להם לכתוב ולקראם מהקונסול.
3. ניתן להיעזר בקבוצת הפיסבוק של הקורס כדי לשאול שאלות בנוגע לתרגיל.
4. יש לעמוד בתקנים לכנתיבת קוד כפי שפורסם במסמך הרלוונטי שנייתן למצוא באתר הקורס.  
נקודות ירדו למי שלא עומד בתקנים אלו.
5. **שימוש לבן המסמך "Coding Standards" מכיל סעיף בנוגע למוסכמות בכתיבת קוד הקשור למשלחות ויריעים (delegates) ו- events.** נא לקרוא ועלמוד במוסכמות אלו.
6. יש לעמוד בהוראות ההגשה כפי שפורסם במסמך הרלוונטי שנייתן למצוא באתר הקורס.  
נקודות ירדו למי שלא יפעיל לפי הוראות אלה. אם יש אי הבנה ניתן לשאול בקבוצת הפיסבוק.
7. **יש למשתמש את העבודה באופן עצמאי לחולטן ולהימנע מהעתקות (הן מתגלות מאוד בקלות).**  
**אפשר לגשת קוד שנוצר ע"י כל בינה מלאכותית ואסור להשתמש ביכולות של השפה ושל ה- Framework שלא הכרנו בקורס.**

בהצלחה!