

תרגיל מס' 4 – ממשקים, משלחות, אירועים ותפריטים

מטרות

- תכנות מונחה עצמים – polymorphism
- הטעמה של עבודה עם ממשקים – interfaces
- הטעמה של עבודה עם מצלבים לפונקציות ומשלוחות (delegates)

ידע נדרש

- עבודה עם ממשקים – interfaces
- עבודה עם משלוחות – delegates – יש לצפות [ב嗣ון החדרה חיה](#)
- שימוש ב- Collections
- עבודה עם מספר פרויקטים

התרגיל

עליכם למשתמש מחלקה שמשיעת בהציגו וניהול של תפריטים הייררכיים לאפליקציות `Console`.

(אם הייתה כזו מחלוקת, הייתם יכולים למשתמש בה בתרגיל מס' 3 כדי להציג את התפריטים של המערכת למשתמש, במקומם לכתוב את הקוד שמציג אותם ומנהל את בחירות המשמש כפי שבתח עשיתם בתרגילים רבים בעבר..).

מחלקה זו צריכה לאפשר (لتוכניתן שימושה בה) לבנות תפריט לפי צרכי (למשל המערכת לניהול מוסך שככבותם בתרגיל הקודם), ע"י הגדרה של פריטי-תפריט בתפריט הראשי או לכל תת-תפריט.

מחלקה זו נקרא `MainMenu`. אפליקציה שרצה להציג תפריטים למשתמש (למשל האפליקציה לניהול מוסך מהתרגיל הקודם) צריכה להחזיק מופיע של מחלקת `MainMenu`.

מחלקה `MainMenu` מאפשרת למשתמש בה (למשל התוכניתן שבונה את האפליקציה של המוסך) לבנות את היררכיית התפריטים (נניח בשלב האתחול באפליקציה שלו) ע"י הגדרת והוספת פריטים (`MenuItem`) לתפריט וכמובן שיווכן להוסיף תת-פריטים לכל פריט וכן הלאה..

כדוגמה לתפריט עם פריטי תפריט שבניו בצורה היררכית, קחו למשתמש 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 לתפריט שבחרנו) וכן הלאה..

המשך בעמוד הבא..

阿富ין

כאמור, עליוכם למשוך רכיב תוכנה שמספק הצגה וניהול של נתונים היררכיים לאפליקציות sole .Console. הפעלת הפונקציה ()Show _MainMenuBar _Show תגרום להציג התפריט הראשי (הרמה הראשונה) ובפועל תקיים את הלולאה שתוארה לעיל הולאה הראשיתichel אשר console זקופה לה כדי להתקיים. בכל שלב:

1. תוצג הרמה הנוכחית של התפריט למשתמש
2. תוצג בקשה למשתמש לבחור באחד הפריטים.
3. תקלט הבחירה מהמשתמש
4. תתבצע בדיקת תקינות קלט + הצגת הודעה מתאימה במקרה של קלט לא תקין
5. ניוט לתת-תפריט / הפעלת פעולה שנבחרה:
 - a. בחירה בפריט שמכיל 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 לקרוות 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 הינו 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):

- כוורת (ברמה הראשונה – כוורת ראשית, בכל רמה אחרת – כוורת הפריט שנבחר ברמה הקודמת)
- רשימת פריטים שישיכים לרמה הנוכחית (מוספרים מ-1)
- פריט שכותוב בו 'Back' או 'Exit' במידה וזה התפריט הראשי (מוספר 0)
- הודעה שմבוקשת מהמשתמש לבחור מבין הפריטים או לבחור בפריט 'Back' / 'Exit'

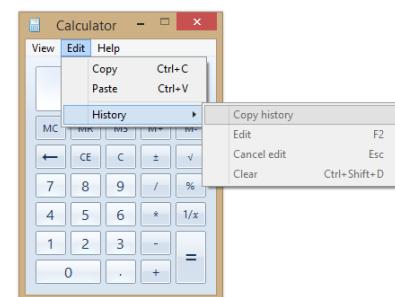
הסברים נוספים למי שלא הבין

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

הדרך לעשות זאת היא ע"י כך שהרכיב יחזק אוסף של פריטי תפריט, וכל פריט יחזק אוסף של תת-פריטים במרקחה הצורך (שייהו את תת-התפריט שלו) וכן הלאה. במינימאלית שמשתמש במינימאלית התפריטים הזה (למשל המערכת לניהול מוסך), נדרש להיות פונקציה שמאתחلت את מבנה התפריט שלה.

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

קחו לדוגמה את התפריט המוצג במחשבון (במרקחה זהה: 3 פריטים בתפריט, 3 פריטים בפריט השני, 4 פריטים בפריט השלישי שלו...):



הוראות למימוש

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

1. ע"י שימוש בממשקים – interface

איזה ממשק יש להגדיר? אילו פעולה/ות הוא יחווף? היכן (באיזה אוסף) יש להגדיר את הממשק? מי אמור למשוך את הממשק? מי אמור להשתמש בממשק? מה צריכה להיות רמת החשיפה שלו (access modifier)? כיצד תקבל המחלקה משתמשת בממשק את האובייקט שמשמש את הממשק? וכו'.

2. ע"י שימוש ב- delegates

איזה delegate יגדיר? באיזה פרויקט? מי יחזיק מופע שלו? מי יפעיל אותו? מי מייצר מופע שלו? וכו'.

פתרון התרגיל יכול solution ובו 3 פרויקטים:

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

זהו איננו פרויקט שמייצר קובץ הריצה (exe) אלא **ספריית מחלקות** בלבד (dll / NET Assembly).

2. Ex04.Menus.Delegates – יכול מימוש לפי הטכניקה השנייה.

זהו איננו פרויקט שמייצר קובץ הריצה (exe) אלא **ספריית מחלקות** בלבד (dll / NET Assembly).

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

הקדומים:

במתודה Main יש לבנות **שני** תפריטים בעלי 2 רמות כל אחד (אחד עבר כל אחת משתי השיטות).
תחילה יוצג התפריט הראשון והמשתמש יוכל לנוט בו. כשהמשתמש יבחר לצאת מהתפריט (Exit) יוצג לו התפריט השני.

להלן המפרט של התפריטים:

ברמה הראשונה יהיו 2 פריטים.

a. הפריט הראשון עם הcotract – "Actions and Info" יציג תת-תפריט שבו יהיו 2 פריטים:

ו. הפריט הראשון יהיה פריט פעולה – "Display Version"

(App Version: 17.2.4.0) פעיל מתודה שמציגה את הטקסט הבא:

ii. הפריט השני יציג תת-תפריט – "Actions" שבו יהיו 2 פריטים:

1. הפריט הראשון יהיה פריט פעולה – "Count Spaces"

פעיל מתודה באפקצייה (כלומר לא מתודה שמומשת בתפריט עצמו אלא מתודה באפקציה שתופעל אם בוחרים בפריט זהה בתפריט) שתפקידו מהמשתמש לכתוב משפט והמערכת אומרת לו כמה רווחים יש במשפט.

2. הפריט השני יהיה פריט פעולה – "Chars Count"

פעיל מתודה באפקצייה (כלומר לא מתודה שמומשת בתפריט עצמו אלא מתודה באפקציה שתופעל אם בוחרים בפריט זהה בתפריט) שתפקידו מהמשתמש לכתוב משפט והמערכת אומרת לו כמה אותיות יש במשפט.

b. הפריט השני עם הcotract - "Show Date/Time" יציג תת-תפריט שבו יהיו 2 פריטים:

ו. הפריט הראשון יהיה פריט פעולה – "Show Time"

(פעיל מתודה באפקצייה שמציגה את השעה הנוכחי).

ii. הפריט השני יהיה פריט פעולה – "Show Date"

(פעיל מתודה באפקצייה שמציגה את התאריך של היום).

שים לב, אין צורך להימנע משיכוף קוד בין שני הפרויקטטים הראשונים.

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

הערות נוספות

1. יש לתכנן את המערכת בצורה הטובה ביותר של קוד מבחינת חלוקה למחלקות ולמתקודות, הורשה ופולימורפיזם. יש להימנע משכפל קוד/לוגיקה ע"י חלוקה למתקודות והפרדה של לוגיקה לרכיבים. שימוש לבב, אלו הם רכיבים שמשמשים מערכת טריטיים **לקונסול**. لكن מן הסתם מותר להם לכתוב ולקראם מהקונסול.
2. ניתן להיעזר בקבוצת הפיסבוק של הקורס כדי לשאול שאלות בנוגע לתרגיל.
3. יש לעמוד בתקנים לכנת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. נקודות ירדו למי שלא עומד בתקנים אלו.
4. **שימוש לבן המסמך "Coding Standards" מכיל סעיף בנוגע למוסכמות בכתיבת קוד הקשור למשלחות ויריעות (delegates - events).** נא **לקראם ו לעמוד במוסכמות אלו.**
5. יש לעמוד בהוראות ההגשה כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. נקודות ירדו למי שלא יפעיל לפיה הוראות אלה.
6. נא **להימנע מהעתיקות** (הן מתגלות מאוד בקלות).

בהצלחה!