מבחן WPF - TASK PROGRAMMING

<u>חלק א'</u>

1. מה היתרון בשימוש ב WPF על פני WINFORMS ?
   א. WPF תומך ב vector graphics
   ב. WPF מאפשר שימוש ב MVVM
   ג. השימוש ב XAML נותן אפשרות לכתוב תצוגה "חכמה" יותר
   ד. WPF יודע להשתמש ביכולות של כרטיס המסך (במידה ויש)
   ה. WPF הכתוב בצורה מבנית נכונה מאפשר UNIT TEST
   ו. כל התשובות נכונות

2. למה מיועד ה Expression Blend ?
   א. כלי לעיצוב גראפי המאפשר לייצר XAML
   ב. סביבת פיתוח לעיצוב XAML למטרות וב בלבד
   ג. חלק מטכנולגיית SILVER LIGHT
   ד. להכנת שייקים (קיצור של בלנדר)

3. מה היתרון בשימוש ב- XAML על פני שימוש בצייר רגיל (כגון הצייר של WINFORM) ?
   א. מאפשר בנייה של התצוגה בצורה חכמה ועשירה בשילוב "קוד"
   ב. עוזר לתמוך במודל MVC ו- MVVM
   ג. בפיתוח צוותי, מאפשר לגרפיקאים ללא יכולות סי שארפ גבוהות לבנות את התצוגה
   ד. מאפשר לעצב את התצוגה בצורה יותר מודולרית (במקום לגרור רכיבים- לכתוב קוד)
   ה. כל התשובות נכונות (או שפשוט תסמן א, ב, ג, ד....)

4. בחר את **כל סוגי** ה BINDING ההגיוניים ביותר (two way, one way, one way to source, one time) – עבור **כל אחד** מהזוגות הבאים:
   (source) Slider **ל-** (target) TextBlock
   (source) TextBlock **ל-** (target) Slider
   (source) TextBox **ל-** (target) TextBox

5. בחר ברכיב התצוגתי המתאים ביותר (StackPanel, DockPanel, Grid, WrapPanel)**-** עבור **כל אחד** מהתצוגות הבאות:
   הצגה של מידע בצורה טבלאית
   הצגה של מידע אנכי "מצטבר", כאשר כל רכיב יתפוס את כל רוחב הקונטרול
   הצגה של מידע אנכי "מצטבר", כאשר כל רכיב יתפוס את רוחב הקונטרול המקסימלי
   הצגה של תפריט צדדי, כיתוב תחתון ותמונה שתיתפוס את יתר התצוגה

6. מה עשוי להיות יתרון כאשר אני מייבא מדיה כ- Resource לאפליקציה? (אחד או יותר)
   א. למנוע תלות בנתיב המקומי כאשר אעתיק את האפליקציה ללקוח
   ב. ה- Resource מתקמפל לגודל קטן יותר ולכן יעבוד מהר יותר
   ג. כדי שבקוד אייצר קישור רלטיבי (מניעת תלות "בהזזת" מיקום התמונה בהרדיסק)
   ד. כדי לתמוך ב- MVVM
   ה. שקר כלשהו
   ו. אף תשובה לא נכונה

7. מה נכון לגבי- DataContext? (אחד או יותר)
   א. חובה להגדירו באפליקציית WPF
   ב. כאשר אגדיר אותו אוכל להגדיר Binding בין הרכיב ב- XAML לבין ה- Property
   ג. ניתן להגדיר אותו בנפרד עבור כל רכיב ב- XAML (דרך הקוד)
   ד. ניתן להגיד אותו עבור החלון כולו
   ה. עבור ListView - אם אגדיר ItemSource עבורו, חובה להגדיר גם DataContext

8. למה משמש ה- Dispatcher Thread?
   א. זה ה- Thread של הצייר באפליקציית WPF
   ב. לשלוח הודעות למשתמש
   ג. לחבר בין ה- XAML לבין הקוד הגראפי
   ד. כל התשובות נכונות

9. מדוע כדאי להשתמש ב- "SafeInvoke" ?
   א. כדי למנוע שגיאה בציור מ- Thread שהוא לא ה- Dispatcher
   ב. כדי למנוע מה- Dispatcher לתת משימה לעצמו לצייר
   ג. כדי למנוע ציור פעמיים של אותו רכיב
   ד. א' ו- ב' נכונות
   ה. א' ו- ג' נכונות
   ו. א', ב' ו- ג' נכונות

10. מה נכון לגבי ICommand ? (תשובה אחת או יותר)
   א. אינטרפייס המכיל בדיקה אם הפעולה אפשרית או לא ואת הפעולה לביצוע
   ב. אינטרפייס שעוזר במימוש MVVM
   ג. אינטרפייס החוסך/ מונע קוד במחלקה של החלון לדוגמא עבור- Button click event
   ד. אינטרפייס שיכול לסייע בטסטים, כדי למנוע צורך בלחיצה "אמיתית" על הכפתור
   ה. משמש רק כפתורים
   ו. כל התשובות נכונות

11. מה נכון לגבי Routed event ? (תשובה אחת או יותר)
   א. פעולה יכולה "לקרות" בכמה רכיבים ביחד – לדוגמא לחיצה על כפתור וגם על הפאנל שמתחתיו
   ב. פעולה כלפי מעלה בעץ נחשבית- Routed bubble event
   ג. פעולה כלפי מטה בעץ נחשבית- Routed tunnel event
   ד. לא אפשרי למנוע מה- event להגיע לכל הרכיבים ולפעול עליהם (רכיבים שהם אחד על השני)
   ה. כל התשובות נכונות

12. מה היתרונות בשימוש ב- Delegate Command? (אחד או יותר)
    א. מאפשר להשתמש ב Delegate בימקום ב- Action
    ב. מאפשר לממש את מה שנדרש ב- ICommand מבלי לייצר מחלקה יורשת
    ג. כיף לעבוד עם Nugget
    ד. מאפשר לחשב מחדש אם הכפתור (לדוגמא) צריך להיות Enabled או לא, באמצעות RaiseCanExecuteChanged
    ה. מאפשר להריץ את הפעולה ישירות באמצעות ExecuteChanged
    ו. מגיע מספריית Prism
    ז. כל התשובות נכונות
    ח. אף תשובה לא נכונה (נו באמת...אתה לא ברצינות חושב לסמן את זה...)

13. מה נכון לגבי ObservableCollection ? (אחד או יותר)
    א. מאפשר לאחסן רשימה של פריטים עבור קונטרולרים כדי לבדוק אם נלחצו או לא
    ב. מאפשר לאחסן רשימה של פריטים שהוספה/ הורדה של פריט יעודכן אוטומטית בתצוגה
    ג. מאפשר לאחסן רשימה של פריטים ששינוי התוכן שלהם יעודכן אוטומטית בתצוגה
    ד. משמש לאחסן רשימה שתהיה Thread Safe

14. עבור **כל אחד** מהמקרים הבאים בחר האם ה- Value Converter יהיה חד-כיווני, דו-כיווני או שבכלל לא צריך אותו:
    TextBlock שיציג את ערכו של ה Slider
    TextBox שיציג את ערכו של הסליידר
    כפתור שיהיה Enabled כאשר מספר התווים ב- TextBox יהיה גדול מאפס (אחרת Disabled)
    TextBlock שיציג את הטקסט שמוצג ב TextBox
    צבע החלון – שיהיה אדום כאשר סכום הערכים ב TextBox גדול מהיתרה המותרת בחוק

15. למה משמש INotifyPropertyChanged? (אחד או יותר)
    א. אינטרפייס שמאפשר בקשה לעדכון התצוגה
    ב. אינטרפייס הנותן לי אפשרות לדעת מתי התעדכנה התצוגה
    ג. אינטרפייס העוזר לממש MVVM
    ד. אינטרפייס העוזר ל TESTING מכיוון שאז אפשר להפעיל פעולות גם כשאין תצוגה
    ה. כל התשובות נכונות (מוזר שכל כך הרבה פעמים כל התשובות נכונות .... אולי זה טריק?)

16. מדוע נשתמש ב- DataTemplate ? (אחד או יותר)
    א. כי אז מעצבים את תבנית תצוגת המידע פעם אחת וניתן להשתמש בה פעמים רבות
    ב. כי איתי המרצה אמר שכדאי (בונוס +5 נקודות למי שבוחר בזה)
    ג. כי אז אין צורך לעדכן DataContext לרכיב התצוגתי
    ד. כל התשובות נכונות

17. מה נכון לגבי השימוש ב Material ? (אחד או יותר ... כרגיל)
    א. מקבלים חינם ומהר ערכת תצוגה מגניבה בפרוייקט
    ב. סגול זה הצבע האהוב עלי
    ג. דורש הורדה באמצעות Nugget
    ד. דורש עדכון של ה- Resource בפרוייקט ברמת האפליקציה וברמת החלון
    ה. דורש שימוש ב MVVM

18. מה נכון לגבי DataGrid ? (שניים או יותר...)
א. מאפשר באופן מהיר לייצר רכיב ויזואלי להצגת / עדכון מידע רשימתי
ב. מאפשר הרכבה ידנית (Manual) של העמודות (אם רוצים) – לדוגמא שינוי של שמות העמודות
ג. מאפשר עבודה במצב של ReadOnly – רק תצוגה ללא עדכון
ד. בהרכבה ידנית (Manual) יש לעדכן את רשומות ה- Enum בתוספת קוד
ה. ניתן להוסיף ValueConverter לשדות, בהרכבה ידנית (Manual)
ו. מעדכן את הרשימה אליה הוא מחובר (מאחורי הקלעים)

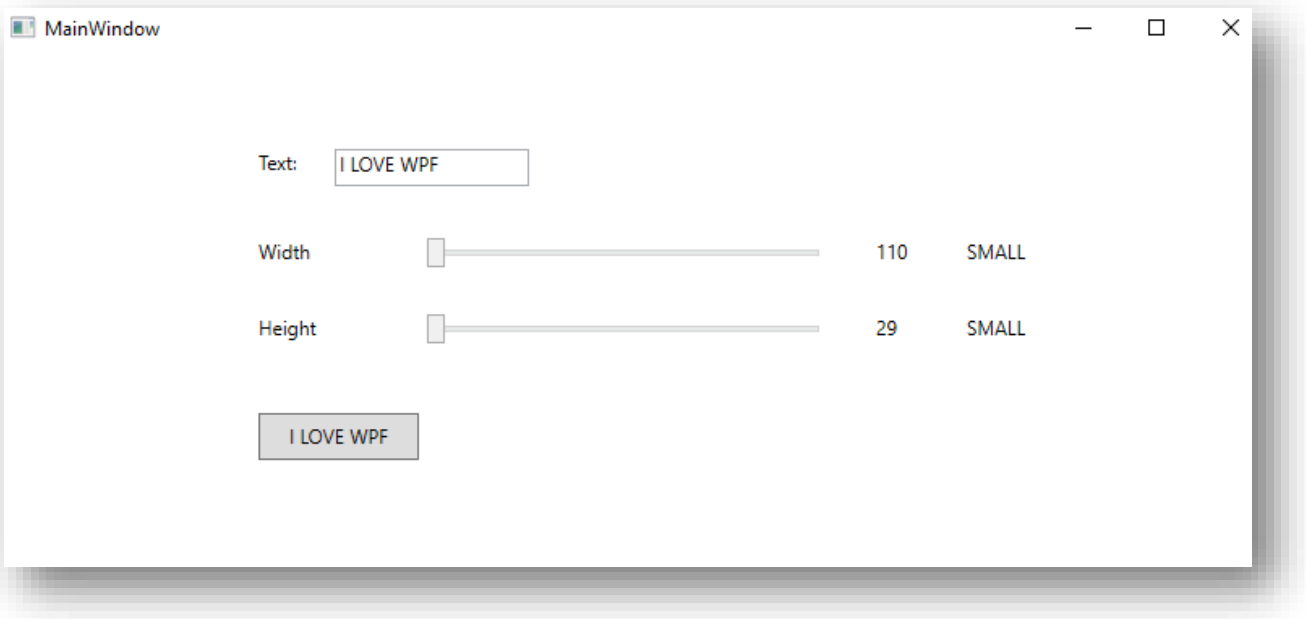
19. מה נכון לגבי UserControl (אחד או יותר)
א. מאפשר כתיבה של רכיב תצוגתי + קוד וקריאה אליו במקומות רבים מחלונות האפליקציה
ב. ניתן להעביר פרמטרים ל- UserControl בעת הוספתו ב- XAML
ג. ניתן להוסיף פרמטרים כגון גדול/ צבע ל- UserControl בעת הוספתו ב- XAML
ד. דורש מימוש של ICommand

20. מה נכון לגבי IDataErrorInfo (אחד או יותר)
א. משמש לולידציה של ערכים ועדכון בתצוגה
ב. מאפשר הוספה של אלמנטים לתצוגה במקרה והולידציה נכשלה
ג. מאפשר שינוי תצוגתי (כגון החלפת צבעים) במקרה והולידציה נכשלה
ד. מחליף את ה- Value Converter במקרים שהמידע שגוי

שאלת בונוס (25 נקודות):
איך היה? (שלוש או יותר)
א. מצויין
ב. היום המושלם ביותר בחיי
ג. היה קל מידי. מתי מגיע החלק הקשה?
ד. שליש גן עדן

<u>חלק ב'</u>

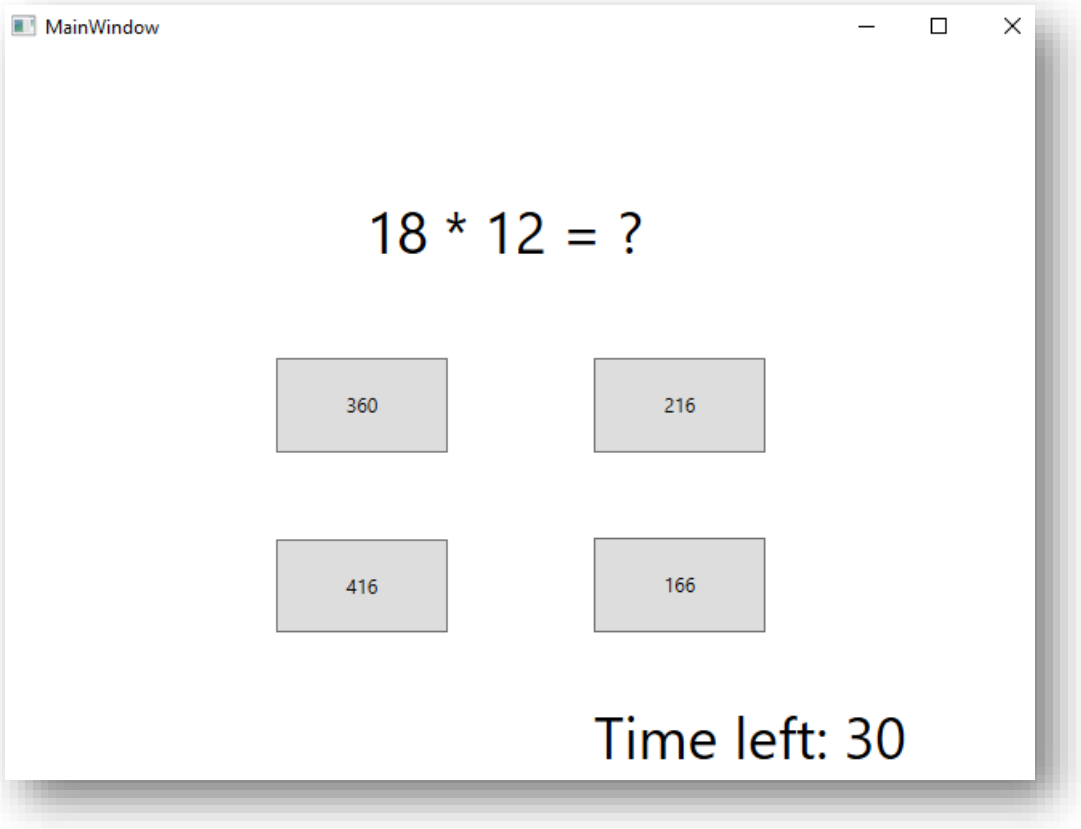
שאלה 1-



כתוב אפליקציית WPF המאפשרת לערוך כפתור (גודל + טקסט). הסליידר הראשון ישמש לבחירת הרוחב של הכפתור, והסליידר השני ישמש לעריכת הגובה של הכפתור. בעת הזזת הסליידרים – הרוחב והגובה של הכפתור ישתנו בהתאמה. בעת שינוי הטקסט בתיבת הטקסט – הטקסט בכפתור יתעדכן (אוטומטית)

- שים לב שהערך ההתחלתי של הסליידרים משקף אורך וגובה הגיוניים (ולא 0), בדוגמא 29, 110
- הערך שמופיע בצד ימין של הסליידר זה הערך הנוכחי של הסליידר (והוא מתעדכן בעת הזזת הסליידר)
- המילה מימין לסליידר מייצגת את גודל הבחירה. עד ל 25% מגודל הסליידר תופיע המילה **SMALL**, עד ל- 50% מגודל הסליידר תופיע המילה **MEDIUM**, עד ל 75% מגודל הסליידר תופיע המילה **LARGE**, מעל 75% תופיע המילה **EXTRA LARGE**
- בעת לחיצה על הפתור – יפתח חלון WPF חדש ובתוכו יוצגו הערכים שנבחרו עבור הרוחב, גובה והטקסט (בבקשה בלי משתנים סטטיים)

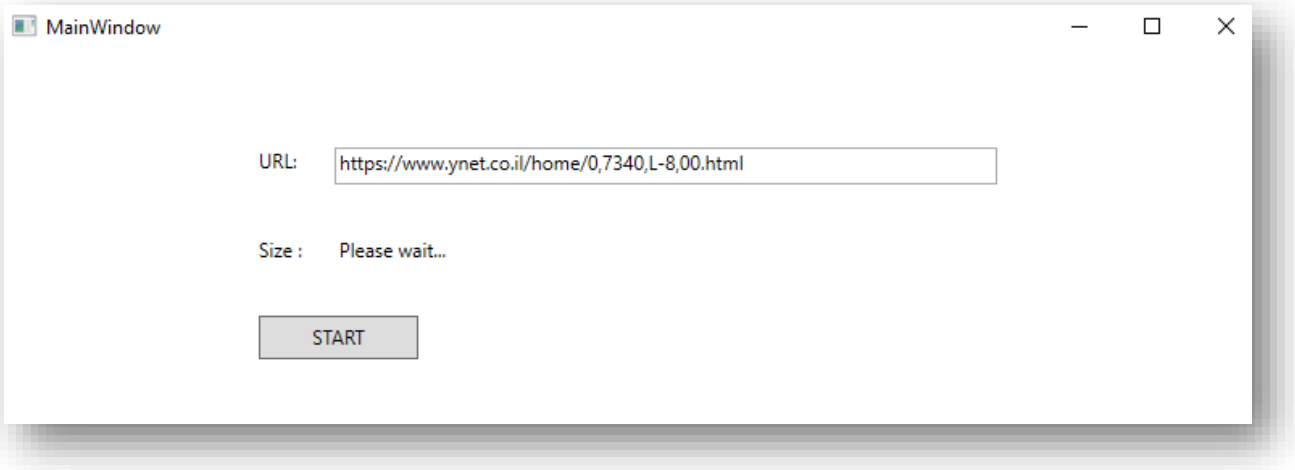
שאלה 2-



כתוב אפליקציית WPF עבור לומדה לחשבון. בלומדה יוצג תרגיל וארבע תשובות אפשרויות. כאשר רק תשובה אחת היא נכונה. המשתמש צריך לבחור את התשובה הנכונה ע"י לחיצה על הכפתור עם המספר הנכון. במהלך התרגיל, ישנו טיימר היורד מ- 30 לאפס המייצג את מספר השניות שנותרו.

- אם מספר השניות הגיע לאפס (ללא בחירת תשובה), הטיימר יפסק, הכפתורים יהיו Disabled וצבע החלון יהפך לאדום בהיר
- אם המשתמש בחר תשובה לא נכונה, הטיימר יפסק, הכפתורים יהיו Disabled וצבע החלון יהפך לאדום בהיר
- אם המשתמש בחר בתשובה נכונה, הטיימר יפסק, הכפתורים יהיו Disabled וצבע החלון יהיה ירוק בהיר
- במהלך הספירה לאחור, אם מספר השניות גדול מחמש עשרה, צבע הכיתוב של השניות יהיה **שחור** אך אם קטן שווה לחמש עשרה- ישתנה צבע הכיתוב של השניות **לאדום**
- בונוס: השתמש ב- ICommand וב- MVVM
- בונוס יותר גדול: העבר את מה שכתבת ל UserControl, והצג שני תרגילים על המסך בו זמנית
- בונוס סופר גדול (אין מספיק זמן לזה אז עדיף שלא תעשה, ראה הוזהרת!): יהיה מאגר של תרגילים ובכל פעם שתרגיל אחד יסתיים יהיה Pause של 3 שניות ואז יוצג התרגיל הבא

שאלה 3-



כתוב אפליקציית WPF אשר פונה ל URL באינטרנט, קוראת את התוכן ומדפיסה את גודל המידע

המשתמש יקליד את הנתיב ב- URL, וילחץ על הכפתור בכדי להתחיל את התהליך. כאשר התהליך יסתיים תופיע כמות המידע שנקראה בתוך תגית הטקסט.

- לאחר שהמשתמש ילחץ על הכפתור, הכפתור יהיה ב- Disabled עד לסיום התהליך
- העדכון של הטקסט בתגית יתבצע באמצעות INotifyPropertyChanged
- הכפתור יעבוד באמצעות ICommand או DelegateCommand
- ממש את הפיתרון בשני דרכים: 1) באמצעות Async Await 2) Dispatcher.Invoke ב- Task נפרד
- בונוס: כל עוד ה- TextBox ריק או שאינו מכיל מתחיל במילה "http", אז הכפתור יהיה ב- Disabled
- בונוס: הוסף תגית זמן (במילישניות) ובה הדפס כמה זמן עבר מהרגע שהמשתמש לחץ על הכפתור וייעצר כאשר הקריאה מה URL תסתיים

קוד עזר:

```
WebRequest webRequest = WebRequest.Create( url );
WebResponse response = webRequest.GetResponseAsync();
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
        string text = reader.ReadToEndAsync();
        // text.Length == is the length of the result
}
```

בהצלחה