



בס"ד

**שנה"ל תשפ"ב, שנתי, מועד ב**  
**שאלון בחינה בקורס: מיני פרויקט במערכות חלונות**  
**מספר קורס: 153007**

<p align="right"><b>מס' תלמיד:</b> _____</p> <p align="right"><b>קמפוס:</b> _____</p> <p align="center">למילוי ע"י הסטודנט</p>
--

- **שם המרצים:** זילברשטיין דן
- **תאריך הבחינה:** 13/06/2022, י"ד סיון תשפ"ב
- **משך הבחינה (בדקות):** 180
- **חומר עזר מותר לשימוש:** אין
- **מחשבון:** אסור
- **המבחן כולל סה"כ שני חלקים עפ"י הפירוט הבא:**
  - חלק א' - 10 שאלות מסוג רב ברירה, ללא בחירה.** בכל שאלה 4 תשובות. השאלות והתשובות מעורבלות. עליך לסמן בדף המיועד לכך את התשובה היחידה המתאימה ביותר, על ידי השחרת העיגול המתאים. בחלק זה כל שאלה שווה 6 נקודות, סה"כ 60 נקודות.
  - חלק ב' - 2 שאלות פתוחות, ללא בחירה.** יש לענות עליהן בדף תשובות נפרד עבור שאלות פתוחות, במקום המיועד לכך. בחלק זה כל שאלה שווה 20 נקודות, סה"כ 40 נקודות.
  - אם אינכם יודעים תשובה לשאלה או לחלק ממנה - יש לרשום בכתב ברור "לא יודע".** אם נרשם על כל השאלה הכיתוב הזה - יינתנו 3 נק' עבור השאלה. אם נרשם קוד לא נכון בעליל - יירשם קנס של 2- נקודות עבור כל רישום כזה וניקוד התשובה יכול להפוך לשלילי ובכך להוריד מהניקוד שהצטבר לכם בשאלות אחרות!

**נבחן/ת יקר/ה, להלן הוראות לבחינה:**

- **בסיום המבחן יש לקחת את השאלון ולשמור אותו עד תום תקופת הערעורים.**

- השאלון ודפי הטייטה לא ייבדקו
- יש לרשום ולסמן במקומות המיועדים לכך בדף התשובות את מספר התלמיד, ומספר המבחן, אין אפשרות לבדוק מבחן אשר נתונים אלה אינם מסומנים באופן מלא ונכון
- אין לכתוב שום דבר נוסף על דף התשובות של חלק א', רישומים נוספים פוסלים את הטופס לבדיקה
- נוהל הבחינות של המרכז האקדמי לב מחייב אותך, באחריותך לקוראו ולהכירו - בחינה עלולה להיפסל על כל חריגה מהנוהל

**בהצלחה רבה!**



**חלק א: 10 שאלות רב-בררה, ללא בחירה, 6 נק' כל שאלה, סה"כ 60 נק'****שאלה מספר 1:****נתונה הגדרת המבנה הבאה:**

```
struct MyStruct { public int Id { get; set; } }
```

**נתון קטע הקוד הבא מתוך התוכנית הראשית:**

```
MyStruct ms1 = new MyStruct() { Id = 5 };
MyStruct ms2 = ms1;
var v1 = new { Id = 5 };
var v2 = new { Id = v1.Id };
Console.Write (ms1.Equals(ms2) ? "yes " : "no ");
Console.Write (ms1.Equals(v1) ? "yes " : "no ");
Console.Write (ms1.Equals(v2) ? "yes " : "no ");
Console.WriteLine(v1 == v2 ? "yes " : "no ");
```

**מה יהיה הפלט של קטע הקוד הזה (משמאל לימין)?**

- א. yes no no no
- ב. yes no yes yes
- ג. yes yes yes no
- ד. yes no yes no
- ה. קטע הקוד לא יתקמפל כי יש בתוכו טעות קומפילציה

**שאלה מספר 2:****נתונה הגדרת המחלקה הבאה:**

```
public class MyClass<T>
{
    int count = 0;
    List<T> element;
    public MyClass() => element = new();
    public void Add() { element.Add(new T()); ++count; }
}
```

**איזה עדכון יש לבצע על מנת שהקוד הזה של הגדרת המחלקה יעבור קומפילציה בהצלחה?**

- א. יש להוסיף לכוותרת של המחלקה את האילוץ הבא: where T : new()
- ב. יש להוסיף לכוותרת של המחלקה את האילוץ הבאים: where T : struct, new()
- ג. יש להוסיף לכוותרת של המחלקה את האילוץ הבאים: where T : class, new()
- ד. יש להוסיף למחלקה בנאי שמקבל פרמטר מסוג T
- ה. אין צורך בשינויים, הקוד של המחלקה עובר קומפילציה בהצלחה כמו שהוא

**שאלה מספר 3:****מהו ההבדל בין קטע קוד שנכתב בתוך בלוק finally של try-catch לבין קטע קוד שנכתב לאחר try-catch ואינו תחום בבלוק finally**

- א. קוד בתוך בלוק finally יתבצע בכל מקרה גם אם בתוך בלוק try קרתה חריגה (שנתפסה או שלא נתפסה) או התבצעה פקודה return
- ב. אין שום הבדל, המילה finally – אופציונלית והקוד שלה יתבצע רק אם לא קרתה חריגה ולא בוצעה פקודה return בבלוק try
- ג. קוד בתוך בלוק finally יתבצע רק אם בתוך בלוק try קרתה חריגה שנתפסה באחד הבלוקים של catch
- ד. קוד בתוך בלוק finally יתבצע רק אם בתוך בלוק try קרתה חריגה שלא נתפסה באחד הבלוקים של catch
- ה. קוד בתוך בלוק finally יתבצע רק אם בבלוק try קרתה חריגה (שנתפסה או שלא נתפסה)

**שאלה מספר 4:****איזו טענה איננה נכונה לגבי דלגט (delegate)?**

- א. delegate עוזר למימוש עיקרון ייחוד אחריות (SRP - Single Responsibility Principle)
- ב. ניתן להגדיר delegate בתוך namespace (מחוץ למחלקות)
- ג. event היא מילת מפתח להגדרת שדה או תכונה מסוג delegate ומוסיף הגנה מזימון או דריסה של ערך השדה/התכונה מחוץ למחלקה/מבנה שבו הוא הוגדר
- ד. delegate מהווה כלי בתחביר של שפה C# למימוש תבנית עיצוב של משקיף (Observer)
- ה. ניתן להוסיף ולהציב ביטוי למבדה למימוש פעולה אנונימית רק למשתנה/שדה/פרמטר מסוג delegate

**שאלה מספר 5:****נתונות 4 הטענות הבאות:**

- טענה A: אלמנט XML בקובץ XAML, יכול להיות פקד גרפי
  - טענה B: אלמנט XML בקובץ XAML, יכול להיות תכונה (property) של פקד גרפי
  - טענה C: אטריביוט XML בקובץ XAML, יכול להיות תכונה (property) של פקד גרפי
  - טענה D: אטריביוט XML בקובץ XAML, יכול להיות אירוע (event) של פקד גרפי
- א. כל הטענות נכונות
  - ב. רק טענות A, C נכונות
  - ג. רק טענות A, D נכונות
  - ד. רק טענות A, B, D נכונות
  - ה. רק טענות A, C, D נכונות

**שאלה מספר 6:****האירוע PreviewMouseDown הוא Routed event מסוג Tunneling. עבור אילו פקדים יופעל האירוע בלחיצה על העכבר?**

- א. מתחיל משורש עץ התצוגה, אח"כ על כל הפקדים שבדרך, עד לפקד שעליו הופעל האירוע (עליו עומד העכבר)
- ב. רק על הפקד שעליו הופעל (עליו עומד העכבר)
- ג. רק על הפקד הראשי בעץ הפקדים (עץ תצוגה)
- ד. מתחיל מהפקד שעליו הופעל (עליו עומד העכבר), ממשיך לפקד שמכיל אותו וכך הלאה עד שורש עץ התצוגה
- ה. מתחיל מהפקד שעליו הופעל (עליו נמצא העכבר) וממשיך לכל הפקדים שבתוכו

**שאלה מספר 7:**

נתון קוד xaml הבא עבור החלון הראשי

```
<Window x:Class="Events.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="MainWindow" MouseRightButtonDown="f2">
    <StackPanel Margin="5" MouseRightButtonDown="f1">
        <Button Margin="5" MouseRightButtonDown="f1">
            <Image Margin="5" Source="/lev.jpg" MouseRightButtonDown="f2"/>
        </Button>
    </StackPanel>
</Window>
```

וגם נתון קטע הקוד הבא מהקוד האחורי של החלון הראשי

```
private void f1(object sender, MouseButtonEventArgs e) => e.Handled = true;
private void f2(object sender, MouseButtonEventArgs e) => e.Handled = false;
```

אלו מהמתודות תופעלנה כאשר יילחץ כפתור העכבר הימני על התמונה?

- שתי המתודות תופעלנה פעם אחת
- המתודה f2 תופעל פעם אחת, המתודה f1 לא תופעל
- שתי המתודות תופעלנה פעמיים כל אחת
- המתודה f1 תופעל פעמיים, המתודה f2 תופעל פעם אחת
- המתודה f2 תופעל פעמיים, המתודה f1 תופעל פעם אחת

**שאלה מספר 8:**

על מנת להבטיח הפרדה בין שכבות BL ו-DAL השתמשנו בכלי שפת תכנות C# ופלטפורמת דוט-נט ותבניות עיצוב. נתונה רשימה של תבניות עיצוב וכלים כדלהלן:

singleton (A)  
 simple factory (B)  
 observer (C)  
 interface עבור חוזה שירות (D)  
 struct עבור חוזה נתונים (E)  
 class עבור חוזה נתונים (F)  
 מהם הכלים שהשתמשנו כפי שפורט לעיל?

- A, B, D, E
- A, B, C, D, E
- A, B, D, F
- A, B, C, D, F
- A, B, C, D, E, F

**שאלה מספר 9:**

נתון קטע הקוד הבא מהתוכנית הראשית:

```
List<int> ls = new List<int> { 1, 2, 5, 4, 3, 6, 5, 4, 2, 2, 1, 3, 5, 4, 3, 6 };
var result = from item in ls
              group item by item into itemGroup
              where itemGroup.Ley > 2
              orderby itemGroup.Key
              select itemGroup;
foreach (var i in result)
    Conosle.WriteLine(i.Key);
```

מה יהיה הפלט של קטע הקוד הזה?

- א. 3456
- ב. 123456
- ג. 333444555
- ד. 33344455566
- ה. 333444555666

**שאלה מספר 10:**

בתכנית מסוימת השתמשנו במתודה, וקיבלנו שגיאת קומפילציה עבור השימוש במתודה. מה ייתכן שקרה?

- א. מעל המתודה היה רשום: [Obsolete("Please avoid using MyFunc()", true)]
- ב. מעל המתודה היה רשום: [Obsolete("Please avoid using MyFunc()")]
- ג. מעל המתודה היה רשום: [DebuggerDisplay("Please avoid using MyFunc()")]
- ד. מעל המתודה לא היה רשום: [Serializable] ובתוכנית ניסינו לעשות סריאליזציה עבודה
- ה. יכולנו להריץ את התוכנית והייתה שגיאת זמן ריצה בתוך מתודה

**חלק ב: 2 שאלות פתוחות, ללא בחירה, 20 נק' כל שאלה, סה"כ 40 נק'**  
 אם אינכם יודעים תשובה לשאלה או לחלק ממנה – יש לרשום בכתב ברור "לא יודע". אם נרשם על כל השאלה הכיתוב הזה – יינתנו 3 נק' עבור השאלה. אם נרשם קוד לא נכון בעליל – יירשם קנס של 2- נקודות עבור כל רישום כזה וניקוד התשובה יכול להפוך לשלילי ובכך להוריד מהניקוד שהצטבר לכם בשאלות אחרות!  
 לא יורד ציון על טעויות זניחות (להערכת הבודק) בתחביר השפה, לא יתקבלו ערעורים על הערכת "זניחות" הטעות.  
**אנא ראה את השאלות בעמודים הבאים!**

**שאלה מספר 11:**

במערכת המנהלת הספקת חבילות בין לקוחות והממומשת בתבנית עיצוב של מודל השכבות נתונים:

1. בישויות נתונים קיים מבנה Parcel, שלצורך שאלתנו מכיל את התכונות הבאות:

```
struct Parcel
{
    public int Id { get; set; } // מספר מזהה אוטומטי רץ
    public int SenderId { get; set; } // מספר מזהה של הלקוח השולח
    public int TargetId { get; set; } // מספר מזהה של הלקוח המקבל
    public DateTime Created { get; set; } // תאריך וזמן הזנת החבילה למערכת
    public DateTime? Collected { get; set; } // תאריך וזמן איסוף החבילה מהשולח
    public DateTime? Delivered { get; set; } // תאריך וזמן אספקת החבילה ליעד
}
```

2. בממשק חוזה שירות של השכבה קיימת מתודת הוספה של חבילה חדשה, כמו כן מתודות נוספות שלא חשובות עבור השאלה:

```
interface DalApi
{
    public int AddParcel(Parcel parcel);
    ... // מתודות נוספות
}
```

3. נתון מימוש כלשהו של השכבה (תניח שקיים מימוש של כל המתודות של הממשק ומתודות העזר הפנימיות), ראה גם הערות בעברית בתוך הקוד להבהרת דרישות השאלה:

```
sealed class DalSome : IDal
{
    private static DalSome? _instance = null;
    public static DalSome Instance => _instance ?? (_instance = new DalSome());
    private DalSome() { initialize(); GetNewParcelsAsync(); }
    private volatile bool _finish = false; // "תדליק" אותו
    private const string _getNewParcelPath = "xml בפורמט";
    private void GetNewParcelsAsync()
    {
        try
        {
            int sender, target;
            xmlParcel = _____;
            if (_____) // בדיקה האם אלמנט החבילה אינו ריק
            {
                // המרה ובדיקה האם המידע המתאים מתוך אלמנט החבילה קיים
                if (_____)
                {
                    _ = AddParcel(
                        new Parcel()
                        {
                            _____
                        });
                }
            }
        }
        catch (Exception ex)
        {
            // בדיקה האם החריגה עקב תקלת אבטחה או קובץ\משאב לא קיים
            if (_____)
            {
                // דיווח תקלה במערכת רישום תקלות פנימית
                // אין זריקת חריגה או סיום מתודה בבלוק הקוד הזה
            }
            else // ex שימוש במשתנה בלי שימוש במשתנה
            {
                _____
            }
        }
    }
}
```



```

        try { _____ } // השהייה של שניה אחת
        catch (ThreadInterruptedException) { }

    }

    [MethodImpl(MethodImplOptions.Synchronized)]
    public int AddParcel(Parcel parcel)
    {
        if (parcel.Id == 0)
            parcel.Id = getNewParcelId();
        else if (exists(parcel.Id)) // בדיקה האם כבר קיימת חבילה עם המזהה הנתון
            throw new InvalidOperationException("Trying to add a parcel with id that already exists");
        // ... הוספת החבילה למקום אחסון מתאים
        return parcel.Id;
    }

    // נתונים נוספים, מימושי מתודות ממשק נוספים, מתודות עזר ...
}

```

המתודה `GetNewParcelsAsync` תפעיל תהליכון שפעם בשנייה יטען משאב xml משירות רשת או מקובץ (הינו הך). תוכן המידע שמתקבל מכיל אלמנט של חבילה אחת בשם `parcel` עם מזהים של חלקי השולח והלקוח המקבל של החבילה החדשה. במקרה שאין חבילות חדשות בזמן בקשה – יחזור אלמנט חבילה ריק. דוגמאות התוכן עבור חבילה חדשה תקינה וחבילה ריקה:

```

<?xml version="1.0" encoding="utf-8" ?>
<parcel>
  <SenderId>94</SenderId>
  <TargetId>72</TargetId>
</parcel>

<?xml version="1.0" encoding="utf-8" ?>
<parcel />

```

**השלימו בדפי תשובות את קטעי הקוד החסרים במתודה `GetNewParcelsAsync` שימו לב** (אנא קראו בתשומת לב, כי אי הקפדה על ההערות תוריד מניקוד התשובה):

- (א) תניחו שכל ה-`using` הנדרשים קיימים בקוד
  - (ב) **התהליכון חייב להיות "אנונימי"** – זאת אומרת אסור לשמור את אובייקט התהליכון במשתנה או שדה כלשהו, וכמובן יש להפעיל את התהליכון בו במקום
  - (ג) יש לבצע בקשת חבילה חדשה פעם בשנייה כל עוד דגל סיום התהליכונים לא "נדלק"
  - (ד) לאחר שנדלק הדגל יש לסיים את (לצאת מ-) המתודה
  - (ה) יש להשתמש בשדה `_getNewParcelPath` עבור מחרוזת של כתובת המשאבה הקובץ של החבילה החדשה
  - (ו) בתוכן xml של חבילה חדשה חייבים להיות שני האלמנטים של מזהים עבור השולח והמקבל בפורמט שניתן להמירו למספר שלם, אך לא מובטח ששניהם אכן קיימים או הערך שלהם תקין
  - (ז) אובייקט לשמירת חבילה חדשה חייב לכלול את זמן הכנסתה למערכת וכמובן היא עוד לא נאספה ולא נמסרה
  - (ח) אם מתקבלת חבילה ריקה, לא עושים עם החבילה כלום וממשיכים להמתנה לפני הבקשה הבאה
  - (ט) תפיסת חריגה:
- בתפיסת חריגה יש להתייחס רק לשתי אפשרויות של תקלה בטעינת xml: בעיית אבטחה (`SecurityException`) או המשאבה הקובץ לא קיים (`FileNotFoundException`), בשני המקרים האלה יירשם דיווח במערכת פנימית של דיווח תקלות (אין צורך שתכתבו את קוד הדיווח).
  - במקרה שנתפסה חריגה של הזכרה לעיל, חייבים לזרוק אותה מחדש **ללא שימוש במשתנה** שמכיל את החריגה שנתפסה
  - (י) חובה לתת ערכים מתאימים לכל השדות של החבילה החדשה עבור הוספתה לשכבה, שימו לב על טיפוס התכונות והשימוש בהם במתודה `AddParcel`
  - (יא) **בתוך** בדיקת ערכי המזהים של השולח ושל המקבל יש להתייחס גם לאפשרות שאחד מהם לא קיים וגם לאפשרות שהערך של אחד מהם או של שניהם אינו תקין – **אסור** שתיזרק חריגה כלשהי מהקוד של הבדיקה

**שאלה מספר 12:**

נתונה מחלקה בסיסית עבור סטודנט (שימו לב שאין לשנות את המחלקה!):

```
public class Student
{
    public string Name { get; set; }
    public int Grade { get; set; }
}
```

נתון קוד xaml של החלון להצגת נתוני הסטודנט:

```
<Window x:Class="Q12Wpf.StudentWindow"
    ...
    xmlns:local="clr-namespace:Q12Wpf"
    Title="Show the mean student"
    DataContext="_____">
<Window.Resources>
    _____
</Window.Resources>
<_____ Orientation="_____" DataContext="_____">
    <StackPanel _____>
        <Label Content="Student name:" />
        <TextBlock HorizontalAlignment="Left" Text="_____" />
    </StackPanel>
    <StackPanel _____>
        <Label Content="Student grade:" />
        <TextBlock HorizontalAlignment="Left"
            Text="_____" />
    </StackPanel>
</_____>
</Window>
```

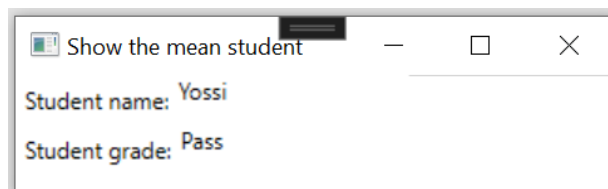
נתון קוד אחורי של החלון להצגת נתוני הסטודנט:

```
public partial class StudentWindow: Window
{
    private IBL bl = BLApi.BLFactory.GetBL();
    _____ DependencyProperty studentProperty
        = _____;
    private Student student {
        get => _____
        set => _____
    }
    public StudentWindow() => InitializeComponent();

    private void Window_Loaded(object sender, RoutedEventArgs e) =>
        student = bl.GetMeanStudent();
}

public class GradeStatusConverter : _____
{
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture) =>
        _____
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture) =>
        throw new NotImplementedException();
}
```

בתצוגה של ציון הסטודנט יש להראות "Pass" אם ציונו 60 ומעלה, אחרת יש להראות "Fail". לפניכם תצוגה של החלון כפי שהוגדר לעיל:



השלימו בדפי תשובות את החלקים החסרים הן בקוד xaml והן בקוד האחורי.