

# אפליקציות אסינכרוניות

## גרסה 5.0

### מבוא לתוכנית הלימודים

בשנים האחרונות הולכת וגוברת הדרישה לשימוש ביכולות אסינכרוניות של תשתיות תוכנה בפיתוח מערכות תוכנה שונות.

הדבר נובע הן מהצורך והן משיפורים טכנולוגיים.

**הצורך** – מערכות תוכנה מטפלות בנפח הולך וגובר של מידע המגיע ממקורות שונים ורבים (מסדי נתונים חיצוניים, מסדי נתונים פנימיים, שירותי רשת, קבצי XML ועוד).

מערכות תוכנה עושות שימוש הולך וגובר ברכיבים גרפיים עתירי חישובים.

שימוש הולך וגובר במידע המגיע מרשת האינטרנט.

**שיפורים טכנולוגיים** – באים לידי ביטוי הן בחומרה והן בתוכנה.

בחומרה - השיפור העיקרי הם מעבדים מרובי ליבות המאפשרים מקביליות אמתית.

בתוכנה - שיפור ושדרוג תשתיות תוכנה המאפשרות ניצול היכולת של המעבדים החדשים תוך הקלה על הקושי הרב שהיה בעבר נחלתם של מתכנתים שעסקו בפיתוח ובתחזוקה של תהליכים אסינכרוניים.

מטרת העל של ההתמחות זו היא לרכוש את הידע הנדרש בתכנות מבוסס תהליכים אסינכרוניים, תוך למידה ובחינה של האספקטים השונים המעורבים בתכנות אסינכרוני:

- תכנות אסינכרוני שמטרתו שיפור חווית משתמש.
  - תכנות אסינכרוני המנצל טוב יותר של משאבי המערכת.
  - תכנות אסינכרוני בניהול מידע רב.
  - תכנות אסינכרוני בצריכת שרותי רשת.
  - תכנות אסינכרוני בתקשורת שמבוססת שרת/לקוח ולקוח/לקוח.
  - תכנות אסינכרוני כמסד להרצת עיבודים כבדים.
  - מאפייני התכנות האסינכרוני במכשירים ניידים.
- וכל זאת תוך למידה והקפדה על תכנות מונחה עצמים כמתודולוגית עיצוב ופיתוח מובילה ומרכזית.

ממטרת העל הזו ניתן לגזור את מטרות המשנה הבאות:

1. הבנת עקרונות התכנות האסינכרוני.
2. הכרות עם מקורות מידע מרכזיים כמסדי נתונים וקבצי XML, הכרות עם שפת השאילתות SQL ושיטות לעיבוד מידע באפליקציות.
3. הבנת עקרונות פיתוח תוכנה במתודולוגית תכנות מונחה עצמים.
4. הבנת עקרונות תקשורת מבוססת מודל OSI.

5. יכולת פיתוח של שרותי רשת.
6. יכולת בניית אפליקציה למכשיר נייד בשפת תכנות מונחת עצמים המשלבת תהליכים אסינכרוניים ותכנות בסביבה מבוזרת.
7. הכרת תשתיות הפיתוח WinRT ו-WP8.

בסיומה של התוכנית, התלמידים יתכננו, יעצבו ויכתבו אפליקציה למכשיר נייד המשתמשת בתהליכים אסינכרוניים בניהול מידע, בתקשורת מחשבים חיצונית (Peer to Peer או Service) ובעיבודים כבדים.

**במסגרת לימודי המעבדה רצוי לבחור אחת מהחלופות הבאות לפי סדר העדיפויות:**

1. תכנות בסביבת אינטרנט
2. מבוא למערכות מידע

במסגרת לימודי היחידה החמישית רצוי לבחור בחלופה של "תכנות מונחה עצמים"

### **המלצה לשיטת ההוראה**

המורים יתמקדו בעיקר בהוראה מן הפרט אל הכלל. כלומר, הצגת דוגמה, צורך והסקת מסקנה לתמונה הכוללת.

חלק נכבד מהתרגול יעשה כשיעורי בית ובמעבדה. התלמידים יריצו את שיעורי הבית ויציגו תקלות, בעיות ואפשרויות באופן התכנות. המורה יכול להשתמש בעבודות התלמידים כדי להסביר את טכניקות התכנות והשימוש ב- MSDN לצרכי למידה עצמית.

חלק משיעורי הבית יכללו את לימוד העזרה באמצעות האינטרנט ושליטה בהשגת ידע לצורך תכנות ושימוש במחלקות קיימות.

### **חלוקה השעות ושעות הפרויקט**

מעצם טבעה של תוכנית לימודים זו, נדרשת התנסות רבה של התלמיד בעבודה מעשית. חלקה המעשי של התוכנית הינה 67% מהשעות הנלמדות, דהיינו 10 ש"ש. חלקה העיוני של התוכנית הינו 33% מהשעות הנלמדות, דהיינו 5 ש"ש.

**שעות הלימוד העיוניות ושעות ההתנסות צריכות להיות משולבות זו בזו.**

יש לשים לב שהנושא "הנחיית פרויקט" נכלל בשעות ההתנסות. את הכנת עבודת הגמר (883599) או פרויקט הגמר (סמלי בחינה 883300 או 883500) יש לבצע בשלבים ולא בסיום

שנת הלימודים. לדוגמה, יש לבקש את תכנון מסד הנתונים בשלבים מתקדמים (תוך אפשרות לשינויים בהמשך).

הערכה לביצוע פרויקט צריכה להיעשות ע"י מורה תוך כדי למידה של נושאי הלימוד שונים במסגרת שעות התנסות.

### חלוקה מומלצת

כיתה י"א – 180 שעות ( 6 ש"ש )

כיתה י"ב – 270 שעות ( 9 ש"ש )

אפשר לבצע חלוקה אחרת פנימית של ביה"ס.

## ראשי פרקים

הערות	שעות	נושא
התמקדות בהורשה ופולימורפיזם, הנחת מוצא - מכירים כימוס	80	OOP
התמקדות ב-RSS	20	XML
לימוד עקרונות ה-XAML ו-Event Driven Programming	40	פיתוח אפליקציות Windows 10 - שלב א
בניית DB, טבלאות, קשרים בין טבלאות, מפתחות, אינדקסים, שאילתות.	40	SQL Server
בניית שכבת מידע המייצגת את מבנה ה-DB בקוד ושכבת גישה לקריאה ועדכון נתונים ב-DB. בניית שכבת BL	20	מודל השכבות - בניית ORM
הבנת עקרונות התכנות עם Delegates	10	Delegates
מימוש Delegates באמצעות מתודות אנונימיות וביטוי למבדא	20	Generics, מתודות אנונימיות וביטוי למבדה
	10	טיפול בחריגים
Threads על קצה המזלגTasks, await, async, ,	40	תכנות אסינכרוני
פיתוח שירותי רשת ומערכות Peer to Peer באמצעות WCF	20	WCF
	70	פיתוח אפליקציות Windows 10 - שלב ב
	80	הגשת עבודת גמר
	450	סה"כ שעות

## פרק ראשון - תכנות מונחה עצמים – 80 שעות

### מטרות כלליות

הבנה ויכולת תכנון וכתיבה של תוכניות מחשב על פי מתודולוגיית תכנות מונחה עצמים (OOP) על מרכיביה השונים: כימוס, הורשה רב-צורתיות.

### מטרות ביצועיות

1. התלמיד ייצור מחלקה על פי עקרונות הכימוס הכוללת תכונות, בנאים, מתודות תוך הקפדה על הגנה על הנתונים.
2. התלמיד יתכן ויכתוב היררכית מחלקות על פי עקרונות ההורשה הכוללת מחלקת בסיס אחת ולפחות שתי מחלקות נגזרות.
3. התלמיד ידע לזהות את הצורך ולממש שיטות וירטואליות המביאות לידי ביטוי את ההתמחות של המחלקות הנגזרות.
4. התלמיד יכיר את הצורך בממשקים, את מטרת השימוש בהם וכיצד לממש אותם הלכה למעשה.

### מושגים והכוונה

לימוד תכנות מונחה עצמים מבוסס על תיאוריה לצד מעשה. מיד עם סיום פרק תיאורטי יש להדגים קוד פשוט המציג את העיקרון הנלמד ורק לאחר מכן להדגים קוד מורכב יותר המביא מתאים פיתרון לבעיה אמיתית. ישנם חלקים במתודולוגיה שהם מאוד אבסטרקטיים ולכן מומלץ מאוד לתת דוגמאות אמיתיות מחיי היום יום המציגות בעיה או צורך ואת הפיתרון הנכון מתודולוגית. מומלץ להציג דוגמה "מתגלגלת" בשלבים של מערכת תוכנה מבוססת OOP, בסיום כל נושא לימוד להוסיף למערכת את היכולות והתכונות שנלמדו. דוגמה לתרגיל שכזה: יומן בית ספר - שמורכב מיומני הכיתה, תלמידים, שיעורי בית, הרשאות וכדומה. בנוסף מומלץ לדרוש מהתלמידים כתרגול מעבדה או שיעורי בית לכתוב בעצמם דוגמה בשלבים, דוגמה למערכת "מתגלגלת" שכזו: מערכת לחישוב שכר. רצוי שהרצת התרגילים בשלב זה תעשה בפרויקט Console application.

המורה יעבור על המושגים הבאים תוך הסבר תיאורטי ומעשי:

### **כימוס - Encapsulation**

1. מבוא תיאורטי ועקרונות
2. הגדרת מחלקה בסיסית
3. האובייקט
4. מתודות
5. בנאים (Constructors)
6. בנאים ורשימות אתחול (Init List)
7. הייחוס this
8. חברי מחלקה סטאטיים
9. בנאים סטאטיים
10. קבועים ותכונות לקריאה בלבד
11. הגדרת קבוע (const data member)
12. הגדרת תכונה לקריאה בלבד (read-only data member)
13. מאפיינים (Properties)
14. מחלקות ומערכים
15. מערך של אובייקטים
16. מחלקת מערך
17. יחסים בין מחלקות: Aggregation, Composition, Association

### **הורשה - Inheritance**

1. מבוא תיאורטי ועקרונות
2. תחביר הורשה
3. גישה למרכיבי מחלקת הבסיס
4. רשימת אתחול
5. כלליות מול התמחות
6. הסתרת מתודות (Method Hiding)
7. הרשאת גישה Protected
8. מחלקות חתומות (Classes Sealed)

### **רב צורתיות - Polymorphism**

1. מבוא תיאורטי ועקרונות
2. מתודות וירטואליות (Virtual methods)
3. כללים במימוש מתודות וירטואליות
4. "קוד גנרי" מול "קוד מתמחה"
5. המחלקה Object
6. מחלקות אבסטרקטיות
7. מתודות אבסטרקטיות
8. תכנון נכון של ההיררכיה
9. מדוע ממשקים?
10. מהו ממשק?
11. מימוש ממשקים.
12. הורשה מרובה של ממשקים
13. היררכית ממשקים
14. אוספים הטרוגניים.

### **דרכי הערכה**

1. בחינה עיונית - תכנון מערכת הכוללת לפחות מחלקת בסיס אחת, שתי מחלקות נגזרות ומחלקת אוסף.
2. בחינה מעשית - עבודת כיתה משולבת עבודת בית – בניית מערכת המיישמת את סעיף 1.

## פרק שני - XML – 20 שעות

### מטרות כלליות

התלמיד יכיר את תקן XML על מרכיביו השונים וירכוש יכולת שימוש במסמכים הכתובים על פי פורמט זה המגיעים מהמחשב המקומי או ממקורות שונים ברשת. התלמיד יכיר את מבנה הקובץ, את מרכיביו, יבין את המשמעות של מבנה מידע היררכי ויכיר את קבצי ה-Schema.

### מטרות ביצועיות

1. התלמיד יכיר את מרחב השמות System.Xml ואת המחלקות XmlDocument, XmlNode, XmlReader.
2. התלמיד יפענח דף XML + Schema.
3. התלמיד יצור קובץ XML ו-Schema מאובייקטים.
4. התלמיד יכתוב תכנית הקוראת נתונים מתוך קובץ XML לוקאלי וקובץ XML מרוחק.
5. התלמיד יסביר ויתאר את מרכיבי הקובץ הבאים: elements, namespaces ו-attributes.
6. התלמיד ימיר elements ל-attributes והיפך.

### מושגים והכוונה

המורה יעבור על המושגים הבאים תוך הסבר תיאורטי ומעשי:

1. מהו תקן XML.
2. יתרונות XML.
3. חסרונות XML.
4. מבנה היררכי – למה? ואיך?
5. מרחבי שמות – Namespace.
6. ייצוג מידע באלמנטים – Elements.
7. ייצוג מידע במאפיינים – Attributes.
8. מהי Schema ומדוע צריך Schema.
9. מרכיבי ה-Schema (מבלי להיכנס לפירוט לתכני מרכיבי הסכימה).

XML הוא מסמך טקסטואלי ויש להדגיש את היותו Firewall Friendly. יש לתרגל את התלמידים ביצירת קבצי XML מתוך מחלקות/היררכית מחלקות. מומלץ לשלב בתרגול פיענוח של קבצי RSS כדוגמה לשימוש נרחב ופופולארי בפורמט.

### דרכי הערכה

1. בחינה עיונית - יצירת קובץ XML ו-Schema XML מהיררכיית מחלקות.
2. בחינה מעשית – מימוש סעיף 1 בקוד.

# פרק שלישי – מבוא לפיתוח אפליקציות – 40 שעות

## מטרות כלליות

Universal Window Platform (UWP) הנה פלטפורמה לפיתוח אפליקציות מבית מיקרוסופט. הפלטפורמה היא Cross Platform, אפליקציות UWP יכולות לרוץ על כל מחשב המריץ Windows 10: מחשבי PC, טאבלטים, טלפונים חכמים, שעון חכם, קונסולת המשחק XBOX One, מכשירי Raspberry PI ו-Arduino, רכיבי IoT (כמו מחשב לביש) ועוד. אותו הקוד שנכתוב עבור מחשב מסוג אחד יוכל לרוץ על שאר המכשירים.

פרק זה מהווה את הבסיס הלימודי של פרק פיתוח אפליקציות בתוכנית הלימודים. באמצעותו נלמד את עקרונות תכנות האפליקציות, נכיר את התכנות מונחה האירועים, נתנסה בתכנות גרפי, שימוש בפקדים ועוד. נעזר בידע הנרכש בו בלימוד תכנות אסינכרוני והתחברות למקורות מידע באמצעות שכבות (נשתמש ב-UWP למימוש נושאים מתוך פרקים אלו שילמדו בהמשך). כמו כן פרק זה מהווה הכנה הכרחית לפרק ההמשך העוסק בפיתוח אפליקציות ואשר יתמקד יותר בשימוש בהתקני החומרה הקיימים במכשירים השונים וביכולות מתקדמות.

הידע הנצבר בפרק זה והניסיון בכתיבה הוא הכרחי בפיתוח אפליקציות. כמו כן, הידע הנרכש בפרק זה ישמש בהמשך להדגמת יכולת צריכת שרותי רשת.

המלצה - כבר בשלב הזה המורה יטיל על התלמידים שיעורי בית שבהם הם יידרשו ללמוד ולשלב בתרגול רכיבים נוספים באמצעות ה-MSDN או מקורות מידע אחרים מהאינטרנט.

## מטרות ביצועיות

1. התלמיד יכיר את המבנה של אפליקציות UWP.
2. התלמיד יכתוב ויריץ תכנית הכוללת טופס המעוצב ב-XAML המכיל פקדי סידור (Layout), פקדי טקסט, פקדי פקודה, פקדי בחירה ורשימות.
3. התוכנית תדע לטפל באירועי הפקדים השונים שבטופס.
4. התלמיד יכתוב תוכנית גרפית המכילה מחלקות גרפיות שונות תוך הקפדה על היררכיית מחלקות כפי שנלמד בפרק אחד (OOP).
5. לתכנן ולבנות טופס רספונסיבי המתאים לסוגי מסכים שונים.
6. התלמיד ידע לקרוא קובץ XML מהאינטרנט (רצוי בפורמט RSS) ולהציג את תוכנו בטופס.
7. התלמיד ידע להריץ אפליקציה בסיסית על מחשבי PC, טאבלט וסמארטפון.

## מושגים והכונה

### מבוא ל-UWP

1. מה זה UWP
2. ממשק משתמש (UI) מול חווית משתמש (UX).
3. מבנה תוכנית UWP – חלוקה בין תצוגה לקוד.
4. מחזור החיים של תוכנית UWP
5. תכנות מונחה אירועים – Event Driven Programming

## XAML

1. מה זה XAML
2. יתרונות שימוש ב- XAML לייצוג גרפי
3. מבנה בסיסי של קובץ XAML
4. XAML – מקבץ דוגמאות בסיסיות.

## עימוד וסידור באמצעות פנלים – Layout and Panels

1. עקרונות עימוד חלון באמצעות פנל.
2. רכיבי Layout
3. StackPanel
4. Grid
5. RelativePanel
6. SplitPanel
7. Canvas

## פקדים

1. מה זה פקד?
2. מבוא לפקדים:
  - i. פקד TextBlock.
  - ii. פקד TextBox.
  - iii. פקד Button.
  - iv. טיפול באירועי הפקדים.
3. פקדי טקסט (TextBlock, TextBox, PasswordBox)
4. פקדי תוכן – Content Control (לדוגמה Button)
5. פקדי אוסף (ItemsControl, ListBox, ComboBox, ListView, TreeView)
6. יכולת ה- Content בפקדי אוסף
7. פקדי בחירה (CheckBox, RadioButton)
8. תפריט "המבורגר"
9. Dependency Property

## התאמת הממשק למכשירים שונים

1. Adaptive UI
2. Adaptive Code
3. מעבר בין חלונות - Navigation

## הספרייה הגרפית

1. Shapes
2. Geometries
3. Transforms
4. Brushes
5. Opacity

## אנימציה – רשות/לימוד עצמי



- עקרונות האנימציה ב- UWP
- Linear Animations
- Storyboards

## כריכת נתונים – רשות/לימוד עצמי

1. עקרונות כריכת נתונים ב- UWP
2. השיטה
3. עקרונות כריכת נתונים באמצעות DataContext
4. הממשק INotifyPropertyChanged
5. המאפיין UpdateSourceTrigger
6. האוסף ObservableCollection<T>

### דרכי הערכה

1. בחינה מעשית - יצירת אפליקציה תוך שימוש בפקד או רכיב שלא נלמד בכיתה.
2. בחינה מעשית – התלמיד יצרוך קובץ XML לתוך מחלקות מתאימות שהוא כתב ויצג את תוכנם בטופס.
3. התלמיד יריץ אפליקציה אשר תתאים גם לרזולוציה של מחשב PC וגם לזו של טלפון חכם.

## פרק רביעי - שפת SQL ו- SQL Server – 40 שעות

### מטרות כלליות

הכרת עקרונות השימוש במסדי נתונים.  
בניית מערכת טבלאות על פי כללי הנרמול.  
הכרת שפת SQL.

### מטרות ביצועיות

1. התלמיד יסביר את הצורך בשימוש במסד נתונים ואת יתרונותיו על פני שימוש בקבצים רגילים.
2. התלמיד יתכן ויכתוב מסד נתונים המכיל מספר טבלאות וקשרי יחיד-רבים ורבים-רבים ביניהן.
3. התלמיד יסביר את השימוש במפתחות.
4. התלמיד יבצע נרמול במסד נתונים שבו שלוש טבלאות לפחות.
5. התלמיד יבצע שאילתה מורכבת ויצג את נתונים מתוך מסד הנתונים.
6. התלמיד יבחין בין סוגי המשתנים במסד הנתונים ( int, nchar, nvarchar, ntext, longint).
7. התלמיד יבצע שאילתה מורכבת היוצרת טבלת ביניים חדשה כדי לבצע עליה שאילתה נוספת.

### מושגים והכונה

### עקרונות

1. שיטות לניהול נתונים (קבצים, מס נתונים)
2. הכרות כללית עם תוכנת מסד נתונים (ב- SQL Server יש להציג דרך SQL Server Management Studio)
3. יצירת מסד נתונים באמצעות דיאגרמה.
4. הטבלה – מושגים, מאפייני הטבלה, מפתחות והוספת טבלה אחת.
5. הוספת נתונים באמצעות SQL Server Management Studio

6. הצגת נתונים באמצעות SQL Server Management Studio

## שאלות

1. מבוא לשפת SQL
2. כללי תחביר כלליים.
3. Select
  1. שאילתת Select פשוטה
  2. הוספת תנאי (Where) לשאילתת Select
  3. סינון שדות בשאילתת Select
  4. מיון עולה, מיון יורד
  5. הוספת שדות חישוביים.
  6. כינויים לשדות
  7. מניעת כפילויות, הגבלת כמות, קיבוץ
  8. פונקציות סקאלריות
2. שאילתת Insert
3. שאילתת Update
4. שאילתת Delete

## נרמול נתונים

1. חמשת הרמות של נרמול נתונים (השלוש הראשונות חובה)
2. בניית מסד נתונים עם מספר טבלאות
3. קשר יחיד לרבים.
4. קשר רבים לרבים.

## שאלות צירוף

1. Inner Join
2. Cross Join
3. Composite Join

## דרכי הערכה

1. בחינה עיונית - ניתוח בעיה אמיתית מהחיים ותכנון מסד נתונים עם מספר טבלאות וקשרים ביניהם (לדוגמה – ניהול חנות אינטרנט: מוצרים, קטגוריות, ספקים .....).
2. בחינה מעשית – מימוש של סעיף 1, כתיבת מסד הנתונים בתוכנת מסד הנתונים.
3. בחינה מעשית - ניתוח שאילתה והצגת תוצאות על טבלה.

# פרק חמישי - מודל השכבות - בניית Model ו- View

## Model – 20 שעות

### מטרות כלליות

מטרת הפרק להכיר את השילוב של המידע בתוכניות תוך שימוש בשכבות. הבנת הבעייתיות שבחיבור ישיר בין התצוגה למידע. הבנת הפתרון – בניה של שכבות ייעודיות. התלמיד יבין שביישומים מודרניים המידע יכול להגיע ממספר מקורות מידע שונים ומגוונים (מסד נתונים פנימי, מסד נתונים חיצוני, קבצי XML, שרות רשת, קבצים ומקורות אחרים). התלמיד ידע כיצד ליצור "מחסן נתונים" (Model) לתוכנית המכיל מידע המגיע ממסד נתונים, קבצי XML או מקורות מידע נוספים. התלמיד ידע כיצד לבנות את השכבה שמקשרת בין מקורות המידע, "מחסן המידע" של התוכנית ושכבת התצוגה.

### מטרות ביצועיות

1. התלמיד יבצע התחברות ישירה בין התצוגה למקורות נתונים (מסד נתונים ו-XML) ללא שכבות.
2. התלמיד יסביר את הבעייתיות בגישה ישירה בין המידע לתצוגה.
3. התלמיד יבנה שכבת Model המכילה מידע משני מקורות מידע שונים (מסד נתונים, XML).
4. התלמיד יבנה שכבת ViewModel אשר נגשת למסד הנתונים, קוראת וכותבת מידע. התלמיד יוסיף לשכבת ה-ViewModel יכולת קריאת נתונים מקובץ XML (רצוי פורמט RSS)

### מושגים והכוונה

1. חיבור מסד נתונים לקוד ב-C# באמצעות ADO.NET - Connected Architecture.
2. ארכיטקטורת תוכנה מבוססת שכבות.
3. בניית Model מבוסס מחלקות מקושרות למסד הנתונים ולמקורות מידע נוספים.
4. בניית ViewModel.

### דרכי הערכה

1. בחינה עיונית - פירוט הסיבות שדורשות שכבות.
2. בחינה מעשית – כתיבת תוכנית מלאה הכוללת את שלושת השכבות שהוסברו בפרק ומכילה לפחות שני מקורות מידע שונים.

## פרק שישי - Delegates- 10 שעות

### מטרות כלליות

מנגנון ה-Delegate מהווה מרכיב חשוב ושימושי בפיתוח ב-.NET Framework. בכלל ובכתיבת מתודות אסינכרוניות בפרט. לימוד והבנה של Delegate חשוב להמשך התוכנית ובייחוד לחלק של תכנות אסינכרוני מבוסס Task. התלמיד יכיר את ה-Delegates ואת ה-Multicast Delegate.

### מטרות ביצועיות

התלמיד יסביר תהליכים הדורשים והממומשים באמצעות Delegate. התלמיד ידע כיצד להגדיר ולהשתמש ב-Delegate בקוד התוכנית.

## מושגים והכוונה

1. מה זה Delegate? מה זה MulticastDelegate.
2. שימוש Delegate בקוד (הגדרה, מופע, הפעלה).

## דרכי הערכה

1. בחינה עיונית - ניתוח תכנית הכוללת Delegate
2. בחינה מעשית - כתיבת תכנית הכוללת משימה הממומשת באמצעות Delegate

# פרק שביעי - Generics, מתודות אנונימיות וביטויי למבדה – 20 שעות

## מטרות כלליות

התלמיד יכיר יכולות מתקדמות של השפה ושל תשתית הפיתוח כהכנה ללימוד תכנות אסינכרוני.  
התלמיד יכיר שיטות מתקדמות להגדרת והרצת Delegates באמצעות מתודות אנונימיות וביטויי למבדה.

## מטרות ביצועיות

1. התלמיד יתכן ולכתוב מתודות, מחלקות, וממשקים גנריים.
2. התלמיד יגדיר Delegate באמצעות מתודות אנונימיות.
3. התלמיד יכיר את התחביר של ביטויי למבדה.
4. התלמיד יגדיר Delegate באמצעות ביטויי למבדה.
5. התלמיד ישתמש בצורה נכונה ב- Delegates המובנים בשפה: Predicate, Action, Func
6. התלמיד יגדיר באלו מקרים להשתמש בכל אחד מה-Delegates הללו ולהתאים לו ביטוי למבדה מתאים.
7. התלמיד יכיר וידע להשתמש בביטויי למבדה מובנים של השפה.

## מושגים והכוונה

1. Generics.
  1. תחביר כללי.
  2. רשימות גנריות.
  3. רשימות גנריות.
  4. מתודות גנריות.
  5. ממשקים גנריים.
  6. Delegates גנריים.
2. מתודות אנונימיות.
3. ביטויי למבדה.
  1. מבנה תחבירי.
  2. ביטוי למבדה פשוט.
  3. ביטוי למבדה בן מספר שורות.
  4. ערך מוחזר מביטויי למבדה.
  5. Predicate<T>
  6. Action<T>
  7. Func<T>

8. שימושים בלמבדה ( , Where, OrderByDescending, OrderBy, Where, )  
(ForEach, Average, Select, Sort, GroupBy)

### דרכי הערכה

1. בחינה עיונית – המרת שימוש ב-Delegate מהשיטה המסורתית למתודות אנונימיות ולביטויי למבדה.
2. בחינה עיונית – התלמיד ידע להגדיר את ההבדל בין Predicate, Action, Func ומתי ראוי להשתמש בכל אחד מהם.
3. בחינה מעשית – התלמיד ידע לכתוב Delegate בשלושת השיטות שנלמדו בפרק זה.
4. בחינה מעשית – התלמיד ידע לכתוב קוד העושה שימוש ב- Predicate, Action, Func.
5. בחינה מעשית – התלמיד ידע להשתמש בלפחות 5 פונקציות בנויות העושות שימוש בביטויי למבדה ובשלושת ה-Delegates הבנויים של השפה: Predicate, Action, Func.

## פרק שמיני - טיפול בחריגים – 10 שעות

### מטרות כלליות

היכרות ראשונית עם טיפול בחריגים ושימוש ב- Try Catch, ההבנה שבתכנון ובניית תוכנה יש להתייחס גם לתקלות ולא רק לזרימה תקינה של התוכנית היא חשובה על מנת לבנות תוכניות אמינות וחסינות מהתרסקויות.

### מטרות ביצועיות

1. התלמיד יפרט אלו פעולות תלויות בגורמים אקראיים שיכולים לגרום להתרסקות התוכנית.
2. התלמיד יפרט אלו שגיאות משתמש עלולות לגרום להתרסקות התוכנית.
3. התלמיד יפרט אלו בעיות רשת עלולות לגרום להתרסקות התוכנית.
4. התלמיד יכתוב קוד המגן על הקוד ממגוון התקלות הנ"ל.

### מושגים והכונה

1. מהם חריגים ומדוע צריך לטפל בהם?
2. עקרונות הטיפול בחריגים?
3. כיצד מערכת ההפעלה וה- CLR מטפלים בחריגים.
4. תחביר (try, catch, finally, throw)
5. המחלקה Exception והמחלקות הנגזרות שלה.
6. בניית מחלקת Custom Exception.

### דרכי הערכה

1. בחינה עיונית - ניתוח קוד שיוצר שגיאה, לכידת השגיאה ותיאור התוצאה
2. בחינה מעשית - כתיבת מחלקה המטפלת בשגיאות מתמטיות, שגיאות משתמש ושגיאות רשת. כתיבת מחלקת Custom Exception (למתקדמים כולל ירושת מחלקה המטפלת בחריגים) – עבודת בית הכוללת הגנה בכיתה.

## פרק תשיעי - תכנות אסינכרוני – 40 שעות

### מטרות כלליות

אחד הפרקים המרכזיים והחשובים בתוכנית הלימודים, הסבר תיאורטי לצד מעשי של פיתוח תוכניות אסינכרוניות המתבססות על Task ועל ה-"מתודה האסינכרונית" כפי שהוצגה לראשונה ב-.NET Framework 4.5

חשוב להקדיש זמן גם לשיטות הישנות של פיתוח אסינכרוני (המחלקה Thread, ThreadStart Delegate) בשביל להבין את הבסיס של הפיתוח האסינכרוני אולם אין טעם להתעמק בכך מעבר לנדרש.

הדגש בפרק זה הוא על פתרונות אסינכרוניים בבעיות של איטיות וקפיאה של ממשק משתמש, ניהול מידע רב ממקורות שונים, עבודה רשתית (Peer to Peer Service), והרצת אלגוריתמים כבדים ואו שילוב ביניהם.

### מטרות ביצועיות

1. התלמיד ידע לזהות מטלות שרצוי לבצעם בצורה אסינכרונית.
2. התלמיד יכתוב קוד אסינכרוני בגישה המסורתית תוך שימוש ב-Thread ו-ThreadStart Delegate.
3. התלמיד יכתוב מטלות אסינכרוניות תוך שימוש במתודה אסינכרונית.
4. התלמיד ידע לסנכרן בין מטלות אסינכרוניות מרובות הרצות במקביל.
5. התלמיד יכיר את הבעיות שיכולות להתקיים בעבודה אסינכרונית כגון: deadlock, starvation, race condition.

### 6. מושגים והכוונה

1. מבואות
  - i. מהי מקביליות.
  - ii. למה מקביליות?
  - iii. מקביליות אמיתית מול פסאודו מקביליות.
  - iv. הבנת המושגים: Parallel, Concurrency, Multithreaded.
  - v. ההבדלים בין Process, AppDomain, Threads.
2. המחלקה Thread - רשות
  - i. כתיבת מטלה אסינכרונית באמצעות המחלקה Thread ו-ThreadStart Delegate
  - ii. הפעלת מספר מטלות אסינכרוניות במקביל.
  - iii. מה מתרחש מאחורי הקלעים? (Thread Scheduler, TimeSlicing).
  - iv. מחזור החיים של Thread.
  - v. שליחת פרמטר ל-Thread.
  - vi. Foreground and Background Thread.
  - vii. הגדרת רמת עדיפות ל-Thread.
3. סנכרון בין Threads - רשות
  - i. שמירה משאבים משותפים
  - ii. Sleep, Join
  - iii. Lock
  - iv. בעיית Deadlock
4. ספריית ה-Task Parallel Library (TPL)
  - i. החסרונות בגישה הקודמת של תכנות אסינכרוני.
  - ii. היתרונות בגישה החדשה.
  - iii. המחלקה Task
  - iv. העיקרון של awaitable והמחלקה Task<T>
5. Async\await

- i. תחביר כללי
  - ii. ערכים מוחזרים
  - iii. פרמטרים
  - iv. שמירה על משאבים משותפים.
  - v. טיפול בחריגים
  - vi. השהיה, ביטול, התקדמות משימות אסינכרוניות.
6. תסריטים אסינכרוניים
- i. הורדת/העלאת מידע מהרשת.
  - ii. עבודה מול מסדי נתונים.
  - iii. חישובים כבדים.

### דרכי הערכה

1. בחינה עיונית – זיהוי מטלות שרצוי לבצע אסינכרונית מבין אוסף המטלות של התוכנית.
2. בחינה עיונית – זיהוי הבעיות שיכולות להיגרם בריצה אסינכרונית של קוד נתון.
3. בחינה מעשית – כתיבת קוד אסינכרוני ב-UWP המריץ מטלות אלגוריתמיות כבדות.
4. בחינה מעשית – כתיבת קוד אסינכרוני ב-UWP לקריאה/כתיבה של מידע ממקורות מידע חיצוניים.
5. בחינה מעשית – כתיבת קוד ב-UWP המסנכרן בין תהליכים אסינכרוניים התלויים אחד בשני תוך מניעת תקלות כמו deadlock, starvation ו-race condition.

## פרק עשר- WCF – 20 שעות

### מטרות כלליות

WCF הנה טכנולוגיה מודרנית לבניית שירותי רשת הבאה להחליף את הטכנולוגיה הוותיקה יותר שהייתה בשימוש החל מהגרסה הראשונה של .NET Framework.

WCF הינה מעטפת נוחה לתכנות (High Level) של התכנים שנלמדו בפרק 10 (Sockets).

התלמיד יבין את היתרונות והחסרונות של שימוש בתשתית High Level כמו WCF אל מול שימוש בתשתית Low Level כמו Sockets.

באמצעות WCF ניתן לכתוב שירותי רשת העושים שימוש בפרוטוקול SOAP (כמו ב-WS) או בפרוטוקולים נוספים כגון: oData, Rest.

התלמיד יכיר את הפרוטוקולים הללו, יתרוניהם ומתי ניתן ורצוי להשתמש בכל אחד מהם. במסגרת הפרק התלמידים ילמדו ויתרגלו בניית שירותי רשת ותוכנת לקוח והתחברות ישירה בין מחשבים תוך שינוי של קובץ הגדרות בלבד.

### מטרות ביצועיות

1. התלמיד יסביר את חשיבותן של שירותי רשת ותכנות מבוזר.
2. התלמיד יבנה שרות רשת מבוסס WCF.
3. התלמיד יגדיר קובץ קונפיגורציה ולשנותו.
4. התלמיד יבנה תוכנת לקוח העושה שימוש בשירות (סנכרוני ואסינכרוני).
5. התלמיד יסביר את ההבדלים בין שלושת הפרוטוקולים Soap, Rest, oData.

### מושגים והכונה

1. מבוא ל-WCF.
2. מדוע צריך שירותי רשת.
3. יתרונות WCF.

4. (Address, Binding, Contract) ABC.
5. בניית שרות רשת מבוסס WCF
6. בניית צרכן.
7. קובץ הקונפיגורציה.
8. פרוטוקול SOAP.
9. פרוטוקול Rest ו- WCF Data Services (oData)

### דרכי הערכה

1. בחינה עיונית – ניתוח שירות רשת ומסמך קונפיגורציה.
2. בחינה עיונית – התלמיד יסביר מה הבדל בין Soap, Rest, oData.
3. בחינה מעשית – כתיבת שרות רשת ותוכנת לקוח אשר צורכת אותו..

## **פרק אחד עשר - פיתוח אפליקציות מתקדם – 70 שעות**

### מטרות כלליות

הכרות מעמיקה עם טכנולוגיה לפיתוח אפליקציות ל-Windows 10 תוך הסתמכות ושימוש בכל התכנים של פרקי הלימוד הקודמים והטמעתם בדגש על תכנות אסינכרוני. בפרק לימוד זה ילמדו היסודות של פיתוח המתבסס על יכולות חומרה כמו חיישנים הקיימים במכשירים. שימוש במסד נתונים לוקאלי (SQLite) בנוסף למסד נתונים חיצוני. התחברות לשרות רשת. התלמידים ילמדו כיצד לתכנת תוך שימוש ביכולות המתקדמות של Windows 10, את הספרייה הגרפית, ניהול משימות ברקע ויכולות מובנות נוספות של מערכת ההפעלה (כגון: LiveTiles, Sharing, Charms, Badge, Toast Notification וכדומה) וכיצד להשתמש בהם לתועלת הלקוח.

### מטרות ביצועיות

1. התלמיד יכיר את ההבדל בין תכנות למחשב PC למכשירים ניידים.
2. התלמיד יכיר את רכיבי החומרה הייחודיים למכשירים ניידים המריצים Windows 10.
3. התלמיד ידע לעצב מסכי ממשק משתמש מתקדמים באמצעות XAML.
4. התלמיד יכיר את סדר פעולות האתחול, חידוש וסיום אפליקציה ואת הדרכים לניצולם לתועלת האפליקציה.
5. התלמיד ידע לשלב רכיבי חומרה וסנסורים באפליקציה.
6. התלמיד ידע לשלב מדיה באפליקציה.
7. התלמיד ידע לייצר מטלות למשימות רקע.
8. התלמיד ידע לנהל מידע לוקאלי ומידע מרוחק באמצעות שרותי רשת.
9. התלמיד ירכוש את הידע הנדרש על מנת להשתמש ביכולות מערכת ההפעלה (כגון: LiveTiles, Sharing, Charms, Badge, Toast Notification וכדומה)

### מושגים והכונה

1. מבוא לפיתוח עבור מכשירים ניידים (סמארטפונים וטאבלטים).
  - i. פיתוח ל-PC מול פיתוח לניידים - הדומה והשונה
  - ii. NET Core – שפה משותפת, עקרונות משותפים, תשתיות הקצה שונות.
  - iii. מסכים וממשקי משתמש.
  - iv. חומרה וחיישנים.
2. מחזור החיים של האפליקציה.
  - i. סדר אירועי אתחול האפליקציה (Activated Event).



- .ii סדר אירועי חידוש ריצה של האפליקציה (Resuming Event).
- .iii אירועי סיום החיים של האפליקציה (Termination Event).
- .3 שילוב סנסורים (GPS, Accelerometer, Camera, Compass)
- .4 שימוש מדיה
- .5 הגדרת משימות רקע.
- .6 ניהול המידע.
  - .i שמירת מידע בקבצים - רשות
  - .ii שמירת מידע לוקאלי באמצעות SQLite
  - .iii שמירת מידע במסד נתונים באמצעות WCF.
  - .iv ניהול מידע משתמש.
- .7 שימוש ביכולות המערכת.
  - .i Tiles ו-Lives Tiles
  - .ii Charms - , Search Charm , Sharing Charm , Setting Charm.
  - .iii Badge Notification
  - .iv Toast Notification
  - .v Lock Screen Notification

### דרכי הערכה

- בחינה עיונית – התלמיד ידע להסביר את ההבדלים בין תשתיות פיתוח למכשירים ניידים לבין NET framework.
- בחינה עיונית – התלמיד יכיר את רכיבי החומרה והסנסורים של מכשירים ניידים ואת תפקידם.
- בחינה מעשית – התלמיד יכתוב אפליקציה בת 3 מסכים העושה שימוש בסנסור אחד לפחות, יכולת מובנית אחת של מערכת ההפעלה, צריכת מידע בשירות רשת בגישה אסינכרונית ושמירת מידע לוקאלי ב-SQLite.

## עבודת גמר – 80 שעות

### מטרות כלליות

עבודת הגמר היא גולת הכותרת של לימוד החלופה. במסגרת הפרויקט התלמיד יציג ידע תיאורטי לצד יכולת מימוש מעשית שיבואו לידי ביטוי בתכנון וכתובת אפליקציה אשר תרוץ על מכשירים שונים המריצים Windows 10.

בעבודת הגמר יינתן דגש מיוחד על תכנון מחלקות על פי עקרונות OOP שנלמדו, על העברת משימות כבדות למתודות אסינכרוניות, שימוש אסינכרוני במקורות מידע חיצוניים (מסד נתונים, XML, שירותי רשת) וניצול נכון של חומרת ויכולות המערכת.

הפרויקט ייכתב בשלבים:

1. בחירת הנושא הפרויקט ותיאור כללי.
  2. כתיבת תיק פרויקט ראשוני הכולל: תיאור המערכת, הצרכים שהיא עונה עליהם, תיאור של הממשק, תיאור של מקורות המידע ומבנה המידע הנדרש, תיאור המודולים השונים.
  3. התחלת הפיתוח.
  4. עדכון תיק הפרויקט בהתאם לשינויים מהכוונה המקורית שנעשו במהלך הפיתוח.
- המורה צריך להקפיד על עבודה בשלבים לעודד חשיבה יצירתית ופתרונות "מחוץ לקופסא".
- הערה** – המורה יקפיד שהיקף הפרויקט יהיה סביר, לא גדול מדי על יכולותיו של התלמיד ולא קטן ופשוט מדי שיכשיל אותו במבדק.