

קורס בדוט-נט תשע"ח

מיני פרויקט

2	מבוא
2	תיאור הארגון
4	תיאור ישויות המערכת
5	תיאור חלקי הפרויקט בקצרה
6	שלב א – הגדרת הפרויקטים עבור מודל השכבות ומימושן
6	חלק א – ישויות – שכבת ה BE
9	חלק ב' – DAL במימוש רשימות
10	חלק ג' – שכבת ה-BL
12	שלב ד' – שכבת ה-UI
13	חלק ב – עדכון שכבת ה-DAL
13	מימוש באמצעות LINQ-To-XML
15	הדרכה ל שימוש ב-GOOGLE MAPS
17	תאריכי הגשות:

מבוא

היעד: כתיבת תוכנה (=פרוייקט) מתפתח במהלך הקורס.

המטרה: תירגול ויישום של:

- ❖ עקרונות שפת C#,
- ❖ עקרונות ארכיטקטורה של תוכנה,
- ❖ יצירת ממשקי משתמש באמצעות תשתית הפתוח המודרנית WPF
- ❖ חשיבה תכנותית ולמידה עצמית.

הערה: ההכוונה במסמך זה היא כללית בלבד, שכן חלק מן הדרישה היא להפעיל חשיבה יצירתית. כמו כן, חלק מהמבחן הסופי יתבסס על נושאים תכנותיים שתבקשו להתמודד איתם במהלך התרגול. חשוב להדגיש שציון מתחת ל-85 בקורס זה כמו בכל קורס מעשי אחר איננו משמעותי בעיניהם של המעסיקים בתעשייה...

דגשים:

- ❖ העבודה תתבצע אך ורק בזוגות (לא בבודדים ולא בשלישיות). הזוגות יהיו קבועים לכל אורך הסמסטר (לא ניתן לעבור במהלך הסמסטר מקבוצה לקבוצה. לא ניתן להיות שותף עם חבר שרשום בקבוצה שונה. בכל אחד מן המקרים האלה התרגיל ייפסל).
- ❖ כל אחד מהשותפים חייב להיות שותף מלא בכל אחד מן השלבים. (במידה ויתברר כי שותף אחד עשה שלב מסוים ואלו שותף אחר עשה את החלק האחר, הציון יהיה פרופורציונלי למספר השלבים שנעשו ע"י כל אחד מן השותפים)

תהליך ההגשה:

- ❖ הפרויקט מחולק לשלושה שלבים המבוססים זה על זה.
- ❖ בסיום כל אחד מן השלבים נקבע מועד הגשה והצגה של הנעשה עד כה.
- ❖ בדיקת שני השלבים הראשונים תעשה בצורה כללית (בעיקר תיבדק הרצה תקינה, הערות על איכות העבודה, ועמידה מדויקת בלוח הזמנים הנדרש). הציון על שני שלבים אלו לא יועבר לידי הסטודנטים.
- ❖ עם הגשת השלב האחרון והמסכם תתבצע בדיקה יסודית של העבודה שתכלול גם הגנה (מעין בחינה בע"פ) של שני השותפים על העבודה, כ"א בנפרד. הציון של ההגנה יהיה שבר בין 1..0, שיוכפל בציון הכללי שישוקלל מבדיקת כל השלבים. הציונים הסופיים עבור כל אחד מהשלבים יינתנו רק בסוף הסמסטר.
- ❖ ציון המעבדה ישוקלל עם ציון המבחן רק בתנאי שהציון במבחן הוא לפחות 60%.

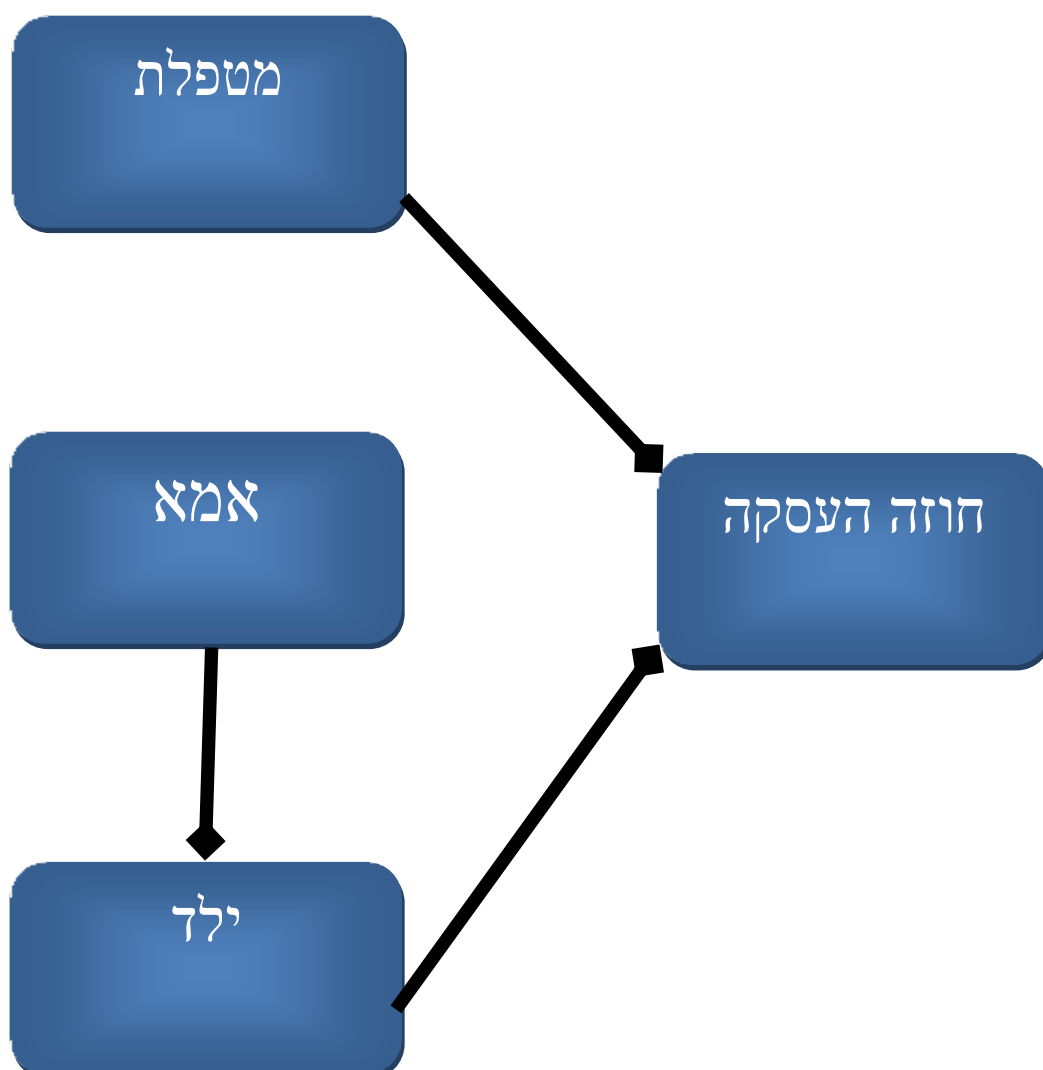
הנחיות להגשת הקבצים במערכת המודל:

בכל אחד מן השלבים, יש להעלות, עוד לפני ההגנה, קובץ zip עם הפתרון למודל (לפי ההנחיות בתרגיל המבוא):
 שם הקובץ יהיה dotNet5778_Project01_xxxx_yyyy שם הקורס והשנה העברית
 dotNet5778 = שם הקורס והשנה העברית
 01 = שם השלב (01,02,03)
 xxxx = 4 ספרות אחרונות בתעודת הזהות של בן הזוג הראשון
 yyyy = 4 ספרות אחרונות בתעודת הזהות של בן הזוג השני
 נא להקפיד על פורמט זה על מנת למנוע מצב של אי קבלת ציון על תרגיל מסוים.

תיאור הארגון

בפרויקט זה נכתוב מערכת (חלקית בלבד) לניהול "לוח מטפלת"

האפליקציה מנהלת מאגר מידע המכיל נתונים של מטפלות. לוח המטפלות מנהל אתר אינטרנטי, שבאמצעותו כל מטפלת יכולה להציע את שירותיה. שמות המטפלות ופרטיהן הרלוונטיים נגישים לקהל הלקוחות שהן בעצם האימהות שמעוניינות בשירותיהן של המטפלות עבור אחד הילדים שלהן. המערכת מאפשרת לאם ליצור קשר עם המטפלת בצורה בלתי אמצעית וכן לחתום איתה על חוזה העסקה. אמנם, את הפרויקט הזה עדיין לא נכתוב כאתר אינטרנטי בתבנית של שרת\לקוח (מפני שזה חורג מנושאי הקורס) אך נכתוב את רוב השכבות בצורה כזאת שהרוב יהיה חופף למערכת אינטרנטית מקבילה.



תיאור ישויות המערכת

מטפלת

מכילה פרטים כגון: מספר ת.ז., שם משפחה, שם פרטי, תאריך לידה, טלפון, כתובת, האם ישנה מעלית בבניין, מספר שנות ניסיון, מספר ילדים מקסימלי, גיל הילדים המטופלים, ימי עבודה בשבוע, שעות עבודה, תעריף לשעה, תעריף חודשי, ימי חופש (ע"פ משרד החינוך\משרד תעשייה-מסחר-תעסוקה) המלצות כשדה טקסטואלי, פרטי חשבון בנק

אמא

מכילה פרטים כגון: מספר ת.ז., שם משפחה, שם פרטי, מספר טלפון נייד, מספר טלפון בבית, כתובת (רחוב, מספר רחוב, עיר) ימים שבהם נידרש הטיפול, שעות שבהם נידרש הטיפול.

ילד

מכילה פרטים כגון: מספר ת.ז. של האם, מספר ת.ז. של הילד, גיל הילד בחודשים, האם ילד רגיל\צרכים מיוחדים?

חוזת העסקה

ישות המקשרת בין המטפלת לילד ומכילה את הפרטים הבאים: מספר חוזת העסקה, מספר ת.ז. של מטפלת, מספר ת.ז. של הילד, האם נחתם חוזת העסקה (שדה בוליאני), השכר הסופי (תשלום בחישוב לפי מספר שעות או תשלום חודשי, בהתאם לבחירה) תאריך תחילת העסקה, תאריך סוף העסקה, ימי העסקה, שעות העסקה.

תיאור חלקי הפרויקט בקצרה

שלב ראשון:

נבנה את הפרויקט על פי מודל השכבות.
מקור הנתונים יופיע כפרויקט נפרד ויכיל רשימות של אובייקטים.
הגישה למקור הנתונים תתבצע באמצעות שכבת ה-DAL.
שכבת ה-BL תכיל את הלוגיקה הנדרשת,
ושכבת ה-UI תהיה ממשק console פשוט, או לחילופין ממשק גרפי WPF פשוט

שלב שני:

פיתוח ממשק המשתמש לממשק ויזואלי גרפי.
תשתית הפתוח הנדרשת של הממשק הוויזואלי הינה WPF.

שלב שלישי:

נשנה את שכבת ה-DAL כך שמקור הנתונים יתבסס על קבצי xml.

שלב א – הגדרת הפרויקטים עבור מודל השכבות ומימוש

בשלב זה של הפרויקט נגדיר את כל אחת מן השכבות שיהוו יחד את המיני פרויקט שיאפיין את לוח המטפלות.

שלב זה כולל 4 חלקים כדלהלן:

- חלק א' - הגדרת המחלקות עבור הישויות, כלומר שכבת ה- BE.
- חלק ב' - הגדרת דרך העבודה מול מקור הנתונים, כלומר שכבת ה- DAL.
- חלק ג' - הגדרת הלוגיקה של המערכת, כלומר שכבת ה- BL.
- חלק ד' - הגדרת ממשק המשתמש של המיני פרויקט, כלומר שכבת ה- UI.

חלק א – ישויות – שכבת ה BE

להלן נפרט שדות שיכללו במחלקות. **חובה** להוסיף שדות ופעולות נוספים.
שים לב! יש להימנע משימוש בפונקציות public במחלקות (מלבד toString), הלוגיקה המרכזית מתבצעת בשכבת ה- BL כפי שיפורט בהמשך.

הגדר פרויקט **Class Library** בשם **BE** ובו המחלקות הבאות. חובה להגדיר כל מחלקה בקובץ נפרד.

בקובץ נפרד יש להגדיר את כל ה Enum הנדרשים.

1. מחלקה בשם *Nanny* שמייצגת מטפלת ותכלול:

- א. מאפיין שמציין את מספר ת.ז. של המטפלת.
- ב. מאפיין שמציין את שם המשפחה של המטפלת.
- ג. מאפיין שמציין את השם הפרטי של המטפלת.
- ד. מאפיין שמציין את תאריך הלידה של המטפלת.
- ה. מאפיין שמציין את הטלפון של המטפלת.
- ו. מאפיין שמציין את הכתובת של המטפלת (שדה string אחד לכל הכתובת מופרד לפסיקים בין רחוב עיר וארץ, כי כך שולחים את זה לגוגל מפות)
- ז. מאפיין שמציין האם יש מעלית בבניין (שדה בוליאני)
- ח. מאפיין המציין באיזו קומה בבניין היא גרה
- ט. מאפיין שמציין את מספר שנות הניסיון של המטפלת.
- י. מאפיין שמציין את מספר הילדים המקסימלי האפשרי של המטפלת.
- יא. מאפיין שמציין את גיל הילדים המינימלי שאותו מקבלת המטפלת (בחודשים)
- יב. מאפיין שמציין את גיל הילדים המקסימלי שאותו מקבלת המטפלת (בחודשים)
- יג. מאפיין בוליאני המציין האם היא מאפשרת תעריף לפי שעה
- יד. מאפיין שמציין את התעריף לשעת עבודה של המטפלת.
- טו. מאפיין שמציין את התעריף לחודש עבודה של המטפלת.
- טז. מאפיין שהוא מערך בוליאני שכל איבר מציין יום ובו שומרים אם המטפלת עובדת ביום זה או לא.
- יז. מאפיין שהוא מטריצה, של שש עמודות ושתי שורות, כל עמודה היא יום בשבוע, בשורה הראשונה שעת התחלה ובשורה השנייה שעת סיום
- זזה למעשה מציין את שעות העבודה של המטפלת בכל יום.
- (ניתן לחשוב גם על מבנה אחר כרצונכם לציון שעות העבודה של המטפלת בכל יום)
- יח. מאפיין בוליאני שמציין האם ימי החופשה שהמטפלת דורשת מבוססים ע"פ ימי החופשה של משרד החינוך או ימי החופשה של משרד תעשייה-מסחר-תעסוקה (תמ"ת).
- יט. מאפיין מסוג מחרוזת המכיל המלצות
- כ. מאפיינים נוספים לפי הצורך.
- כא. *ToString*.

2. מחלקה בשם Mother שמייצגת אמא שרוצה להעסיק מטפלת ותכלול:

- א. מאפיין שמציין מספר מזהה של האם.
- ב. מאפיין שיציין את שם משפחת האם.
- ג. מאפיין שיציין את השם הפרטי של האם.
- ד. מאפיין שיציין את מספר הטלפון הנייד של האם.
- ה. מאפיין שמציין את הכתובת של האם (שדה string אחד לכל הכתובת מופרד לפסיקים בין רחוב עיר וארץ, כי כך שולחים את זה לגוגל מפות)
- ו. מאפיין שמציין את הכתובת שבסביבתה האם מחפשת מטפלת.
- ז. מאפיין שהוא מערך בוליאני שכל איבר מציין יום ובו שומרים אם האם צריכה מטפלת ביום זה או לא.
- ח. מאפיין שהוא מטריצה, של שש עמודות ושתי שורות, כל עמודה היא יום בשבוע, בשורה הראשונה שעת התחלה ובשורה השנייה שעת סיום
- ז. זה למעשה מציין את שעות הטיפול הנדרשות של האם בכל יום.
(ניתן לחשוב גם על מבנה אחר כרצונכם לציון שעות העבודה של המטפלת בכל יום)
- ט. מאפיינים נוספים לפי הצורך.
- י. מאפיין הערות מסוג מחרוזת
- יא. ToString.

3. מחלקה בשם Child שמייצגת ילד ותכלול:

- א. מאפיין שמציין מספר מזהה של הילד.
- ב. מאפיין שיציין מספר ת.ז של האמא שלו (חייב להופיע כזה)
- ג. מאפיין שיציין את השם הפרטי של הילד
- ד. מאפיין שיציין את תאריך הלידה של הילד.
- ה. מאפיין בוליאני שיציין האם הילד הוא בעל צרכים מיוחדים.
- ו. מאפיין מחרוזת שמציין את הצרכים המיוחדים.
- ז. מאפיינים נוספים לפי הצורך.
- ח. ToString.

4. מחלקה בשם: Contract שמייצגת חוזה העסקה (כלומר את הקשר בין המטפלת לילד המטופל) ותכולת:

- א. מאפיין שמציין את מספר חוזה העסקה. (קוד רץ ייחודי בן 8 ספרות)
- ב. מאפיין שמציין מספר מזהה של המטפלת. (חייב להופיע כזה)
- ג. מאפיין שמציין מספר מזהה של הילד. (חייב להופיע כזה)
- ד. מאפיין בוליאני שמציין האם נערכה פגישת היכרות ? (כן/לא)
- ה. מאפיין בוליאני שמציין האם נחתם חוזה העסקה ? (כן/לא)
- ו. מאפיין שמציין את השכר בתשלום לפי שעות. פר שעה
- ז. מאפיין שמציין את השכר החודשי בתשלום חודשי.
- ח. מאפיין המציין האם החוזה הוא לפי שעה או חודשי
- ט. מאפיין שמציין את תאריך תחילת ההעסקה.
- י. מאפיין שמציין את תאריך סוף ההעסקה.
- יא. מאפיינים נוספים לפי הצורך.
- יב. ToString

הערה: כל תקלה, כתוצאה מפעולה כלשהי תגרור אחריה טיפול בחריגות לפי מנגנון החריגות הקיים ב-C#. לדוגמא: ייתכן ובחוזה ההעסקה יירשם ילד למטפלת שגילו אינו בטווח הגילאים שהמטפלת מקבלת.

חלק ב' – DAL במימוש רשימות

להלן נפרט פעולות אפשריות. **חובה** להוסיף פעולות נוספות.

הוסף פרויקט מסוג `class library` בשם `DAL` ובו בצע את הפעולות הבאות:

א. הגדר (interface) ממשק בשם `Idal`.

בממשק הנ"ל הגדר חתימה של פונקציות שימושיות עבור האפליקציה כגון:

- הוספת מטפלת.
- מחיקת מטפלת.
- עדכון פרטי מטפלת קיימת.
- הוספת אם.
- מחיקת אם.
- עדכון אם קיימת.
- הוספת ילד.
- מחיקת ילד.
- עדכון ילד.
- הוספת חוזה העסקה
- עדכון חוזה העסקה
- מחיקת חוזה העסקה
- קבלת רשימת כל המטפלות.
- קבלת רשימת כל האמהות.
- קבלת רשימת כל הילדים ע"פ האמהות.
- קבלת רשימת כל החוזי העסקה.

ב. צור פרויקט חדש בשם: `DS` והגדר בתוכו מחלקה בשם `DataSource` שתכיל את הנתונים (רשימות) של הישויות שלנו.

מחלקה זו תכיל 4 רשימות סטטיות (אוספים מסוג `list<>`) של המחלקות הנמצאות ב- `BE`.

ג. הגדר מחלקה בשם: `Dal_imp` אשר מממשת את ממשק ה- `Idal` הנ"ל.

הפונקציות של המחלקה הזאת יעבדו מול הרשימות שבמחלקה `DataSource`.

הערה:

- על שכבת ה- `DAL` לוודא שמספר הת.ז. איננו קיים כבר. בישויות חוזה העסקה שכבת ה- `DAL` היא זו שמוסיפה לישות מספר זיהוי הרץ.
- הוספת חוזה העסקה תתאפשר רק ע"י בדיקה שאכן המטפלת הזאת קיימת במאגר המטפלות, ושהאם קיימת במאגר האימהות.

חלק ג' – שכבת ה- BL

הגדר ממשק (interface) בשם: IBL שבו החתימות של השיטות בדיוק כמו ב- IDAL

וכן, על שכבת ה BL לבצע את הלוגיקה הבאה:

- אין אפשרות להוסיף מטפלת מתחת לגיל 18
- אין אפשרות לחתום חוזה העסקה עבור ילד שקטן מגיל 3 חודשים.
- בחישוב השכר לחוזה, נלקח בחשבון האם לילד יש אחים נוספים אצל המטפלת, במקרה כזה תהיה הנחה בתשלום חודשי של 2% . (ראה בפסקה הבאה ...)
- חישוב השכר נטו לחודש נעשה ע"פ שתי אפשרויות, לפי בחירת האם: במידה והבחירה היא בשכר שעתי, יחושב מספר השעות השבועי כפול 4. במידה והבחירה היא בשכר חודשי, השכר יהיה לפי התעריף החודשי של המטפלת
- בנוסף, התשלום לוקח בחשבון את מספר הילדים של אותה האם שכבר מטופלים אצל אותה מטפלת (הרעיון הוא שעבור כל ילד נוסף, שיעור ההנחה גדל ב-2%)
- לא ניתן לחתום חוזה עם מטפלת שכבר מילאה את מכסת הילדים שאותם לוקחת (כלומר – יש לבדוק לפני חתימה חוזה כמה חוזים כבר נחתמו עם מטפלת זאת)

בנוסף על שכבת ה BL להוסיף את חתימת הפונקציות הבאות:

- **שימוש ב-Google Maps - כתיבת פונקציה המשתמשת ב-google maps ומחזירה מרחק בין שתי כתובות נתונות.**
(שים לב שבהמשך קריאה מה UI שגוררת שימוש בפונקציה זו מחויבת להתבצע בתהליכון נפרד)
- פונקציה שמקבלת את פרטי האם ומחזירה את כל המטפלות המתאימות לאילוצי האם או נכללות בהן, לפי ימי עבודה נצרכים ושעות עבודה נצרכים. הפונקציה הזאת בעצם יוצרת את התיאום הפוטנציאלי הראשוני בין המטפלת לאם. הערה: גם מטפלת שעובדת יותר ממה שאם צריכה זו אופציה.
- שכלול של הפונקציה הקודמת – אם אין התאמה מלאה בין דרישות האם והמטפלת יש להחזיר את 5 התוצאות הקרובות ביותר, בהגדרה קרובות כל אחד יפעל לפי שיקול דעתו וינמק אותה בהערות למנחה הפרויקט שלו.
- פונקציה שמקבלת את אילוצי האם ומחזירה את כל המטפלות שנמצאות בטווח קילומטרים רצוי מהכתובת שציינה האם. שים לב! במידה והאם לא ציינה כתובת, הכתובת לגביה תתבצע הבדיקה היא כתובת ביתה של האם.
הערה: את זה נבצע בעזרת שימוש בפונקציה שהוגדרה לעיל לחישוב מרחק בעזרת Google Maps ולכן צריך לשים לב שהקריאה לפונקציה זו מה UI תתבצע בתהליכון נפרד.
- פונקציה שמחזירה את כל הילדים שעדיין לא נמצאו להם מטפלת.

- פונקציה שמחזירה את כל המטפלות שעובדות לפי חופשות של משרד התמ"ת.
- פונקציה שמחזירה את כל חוזי ההעסקה שמתאימים לתנאי מסוים (הכוונה שהפונקציה מקבלת delegate שמתאים לפונקציות שמקבלות חוזה העסקה ומחזירות bool וכך מוגדר התנאי)
- פונקציה שמחזירה את מספר חוזי ההעסקה שמתאימים לתנאי מסוים (הכוונה שהפונקציה מקבלת delegate שמתאים לפונקציות שמקבלות חוזה העסקה ומחזירות bool וכך מוגדר התנאי)

שימוש ב Grouping

- הגדר פונקציות המחזירות את הקבוצות הבאות (ע"י שימוש ב-Grouping)
- כל אחת מהפונקציות הבאות תקבל בנוסף משתנה בוליאני שיציין האם להחזיר את התוצאות בצורה ממוינת (איך למיין זה לפי שיקול דעתכם) ערך ברירת מחדל של פרמטר זה יהיה false (לא ממוין)
- רשימת מטפלות מקובצת (Grouping) ע"פ גיל הילדים. הפונקציה תקבל משתנה בוליאני שיציין אם להחזיר את התוצאות לפי הגיל המקסימלי או הגיל המינימלי. לשיקול דעתכם באיזה טווח להגדיר קבוצת גיל (כל חודש, כל שלושה חודשים וכו..), אך בכל מקרה יהיה ערך מספרי אחד בהתאם לכל אחת מקבוצות הגיל שחילקתם.
 - חוזי העסקה של הגמ"ח מקובצים (Grouping) ע"פ מרחק בין בית המטפלת לבין הכתובת שבה האם מחפשת מטפלת, לדוגמה כל חוזי ההעסקה לפי מרחק עד 5 ק"מ, כל חוזי ההעסקה לפי מרחק של עד 10 ק"מ וכן הלאה. כדי לחשב מרחק יש להשתמש בפונקציה המחשבת מרחק. הערה: את זה נבצע בעזרת שימוש בפונקציה שהוגדרה לעיל לחישוב מרחק בעזרת Google Maps ולכן צריך לשים לב שהקריאה לפונקציה זו מה UI תתבצע בתהליכון נפרד.

חובה לעשות שימוש בכישרון היצירתיות שלכם ולהוסיף לפחות עוד 6 פונקציות שמתאימות להיות בשכבת ה-BL ועובדות על הנתונים המוחזרים מה-DAL.

הגדר מחלקה חדשה בשכבת ה-BL שתממש את הממשק IBL הנ"ל המימוש יכלול שימוש ב- Linq to object וביטויי למבדה.

יש לעשות שימוש בלפחות 4 ביטויי Linq בשכבת ה-BL וה-DAL כל אחת. יש לעשות שימוש בלפחות 4 ביטויי למבדה בנוסף לביטויי Linq דלעיל.

כאן תתבצע ההגנה הראשונה – אפשר בממשק קונסול או גרפי פשוט.

לשם הבדיקה של הדברים אנו ניצור פרויקט זמני בשם PL. תוכל לממש אותו או בעזרת console application או ממשק גרפי WPF פשוט. אפשר בשלב זה לעשות הכל בחלון אחד. בכל מקרה עליך לקרוא לפונקציות הנמצאות ב-BL ולבדוק אותן.

הערה:

כדי לעבוד בשלבים מומלץ בהתחלה ליצור משהו שבדוק את ה-BE אח"כ את הפונקציונליות ב-DAL כדי לראות שאכן זה עובד ואז לחבר את ה-BL.

כמו כן יש להגדיר ב-BL פונקציה לאתחול והוספת מופעים ראשוניים של המחלקות שהגדרת (הרעיון שבהפעלה הראשונית, יהיה כבר מידע בסיסי זמני על מנת שנוכל לעבוד עם התוכנה)

שלב ד' – שכבת ה- UI

בשלב זה ניצור ממשק גרפי לפרויקט באמצעות תשתית הפתוח WPF.

ניצור פרויקט חדש מסוג : WPF

ונקרא לו **PLWPF**.

כמובן שיהיה עליך ליצור לו reference מתאים ל-BL וכן ל-BE .
עליך להגדיר ישות בשם: BL מטיפוס: IBL כפי שראית בדוגמה בהרצאה. לדוגמא:

```
IBL bl
public Window1()
{
    bl = FactoryBL.getBL();
}
```

למעשה עליך לתכנן את המסכים אשר יקראו ויאפשרו למשתמש לגשת לפונקציונאליות הנמצאת ב-BL.

עליך לתכנן מסך לכל ישות בדומה למה שראינו בכיתה עם "הסטודנטים והקורסים", כאשר יהיו לפחות המסכים הבאים:

- מסך הוספת, עדכון, מחיקה של מטפלת.
- מסך הוספת, עדכון, מחיקה של אם.
- מסך הוספת, עדכון, מחיקה של ילד.
- מסך הוספת, עדכון, מחיקה של חוזה העסקה. כאשר מעוניינים להוסיף "חוזה העסקה" יש לחבר בין האם לבין אחת מן המטפלות האפשריות ע"פ אילוצים רלוונטיים כמו למשל: ימי עבודה, מרחק, מחיר.
- בנוסף – לפחות עוד 3 מסכים המציגים שאילתות שמימשתם ב-BL.

התכנית תיפתח במסך ראשי המפנה לאפשרויות השונות. יש לשים לב שאנו לא יוצרים כל פעם מופע חדש של ה-BL (איך אפשר לעשות זאת?)

הערה: כאשר ישנו מסך המאפשר הוספת ערך כמו מטפלת, אם וכדו' שכבר קיימים במערכת או עדכון, יש לאפשר בחירה של שדה זה ע"י פקד ComboBox שבו נקבל רשימה ממנה יש לבחור. כמו כן כאשר אנו מאפשרים עדכון, לאחר בחירה מתוך רשימת הישויות לעדכון, יופיע המפתח על המסך ויתמלאו יתר השדות באופן אוטומטי, כאשר לא ניתן יהיה כמובן לשנות את ערך המפתח. ישנה אפשרות שהוספה עדכון ומחיקה יהיו למעשה אותו מסך, בהתאם לצורך באותו רגע.

(אין להוסיף פונקציונאליות מעבר ל- CRUD בסיסי ל-DAL !)

כמו כן עליך לדאוג להוסיף זריקה של חריגות במקומות המתאימים ותפיסה של החריגות כדי שהתכנית לא "תעוף" במקרה שפעולה מסוימת נכשלה בזמן הרצה.

לאחר השלמת לימוד כל ה-WPF יש צורך לדאוג שכל האלמנטים שנלמדו ב-WPF מוטמעים בפרויקט. היכן ואיך לממש, זה תלוי בכל אחד ואחת לפי יצירתיותם.

הערה חשובה: יש לעשות שימוש ב:

Resources, data binding, converter, data context, dependency property, trigger,

כאן תתבצע הגנה השניה במספר.

חלק ב – עדכון שכבת ה-DAL

מימוש באמצעות Linq-To-XML

הוספה ומימוש מחלקה נוספת ב-DAL המממשת את ה-IDAL
באמצעות Linq – to –XML:

עליך להוסיף לפרויקט DAL מחלקה נוספת בשם Dal_XML_imp

יש להכין קבצי xml שיחליפו את האוספים שיצרנו כבר (כלומר יחליפו את מקור הנתונים) אחד לכל אחת מהישויות, אשר ייכתבו בפורמט המתאים למבנה הישות אותה הוא מייצג. קבצים אלו יהיו בתוך תיקיה נפרדת ב-solution.

יש לעשות אותו תהליך שעשינו בכיתה כדי ליצור את האתחולים, אפשרות שמירה וטעינה של קובץ וכן אפשרות תשאול ע"י שאילתת Linq.
לאחר מכן יש לממש את כל המתודות של הממשק של ה-IDAL.

לגבי עבודה עם קבצי XML מקומיים, ניתן להיעזר בהדרכה להלן:

עבור אתחול קבצי הXML:

לדוגמה:

`studentRoot = new XElement("students");`
 כעת נוכל כל פעם להוסיף ולהוריד ולעדכן אלמנטים בקובץ במידה והקובץ כבר קיים, פשוט לא ניצור אותו. הקוד יהיה בערך כך:

```
public XmlSample()
{
    if (!File.Exists(FPath))
        CreateFiles();
}

private void CreateFiles()
{
    studentRoot = new XElement("students");
}
```

עבור המספר הרץ של חלק מהמחלקות ועוד כמה הגדרות:

הבעיה:

כשמימשנו את המספר הרץ השתמשנו במשתנה סטטי שגדל במהלך התוכנית. כעת שנשמור את הנתונים ב- xml בפעם הבאה שנפתח את התוכנית אותו משתנה יתאפס שוב ואז המספר הרץ יתחיל מ- 0

הפתרון:

נגדיר קובץ XML בשם config.xml ושם נגדיר את המספר הרץ ועוד הגדרות שנרצה כל פעם שנשמור אובייקט שמשתמש במספר הרץ נעדכן גם את קובץ ה config.xml בפעם הראשונה שה dal נוצר הוא ידאג לעדכן זאת במחלקה.

הערה:

עבור קובץ ה- config ועוד קובץ אחד של אחת המחלקות, יש להשתמש ב- linq to xml עבור כל פעולות ההוספה עדכון ומחיקה ושליפה.

במימוש של שאר המחלקות ניתן להשתמש ב- serialize, כלומר לעבוד עם הנתונים ב- list ובסופו של דבר לשמור את זה ל- XML באמצעות xmlSerialize

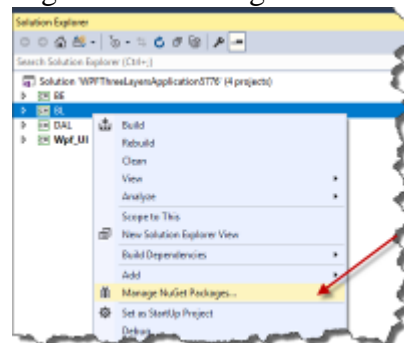
כאן תתבצע בעז"ה ההגנה הסופית

בהצלחה !

הדרכה ל שימוש ב-Google Maps

דבר ראשון, עלינו להוסיף ספריות (DLL) לפרויקט.

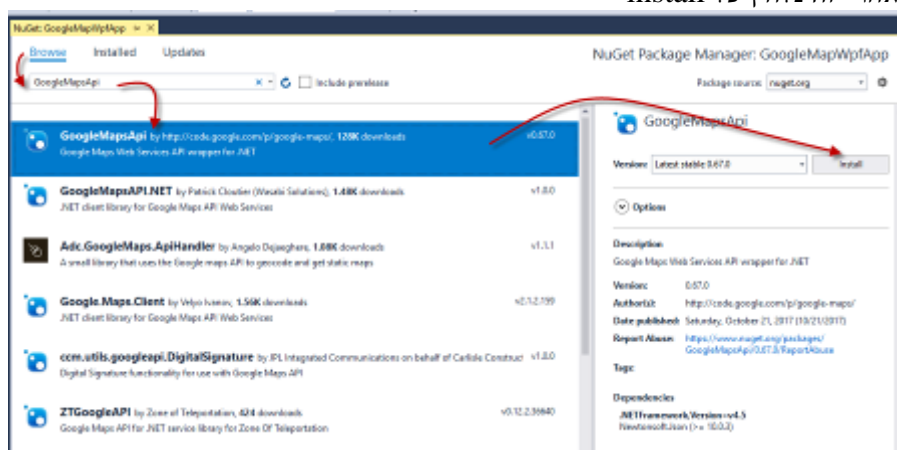
נלחץ מקש ימני על הפרויקט שבו נרצה להוסיף את הספריות לשימוש ב-Google Maps (במקרה שלנו ב BL) ונבחר ב Manage NuGet Packages ...



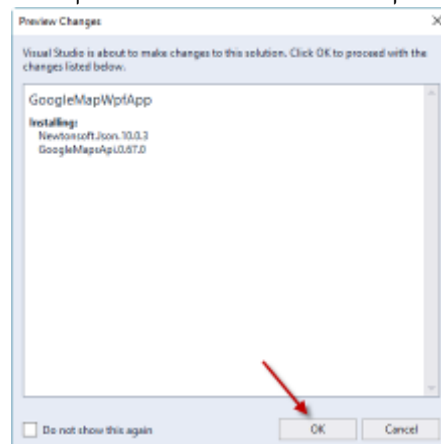
בחלון שיפתח נבחר ב Brows

בחיפוש נכתוב GoogleMapsApi

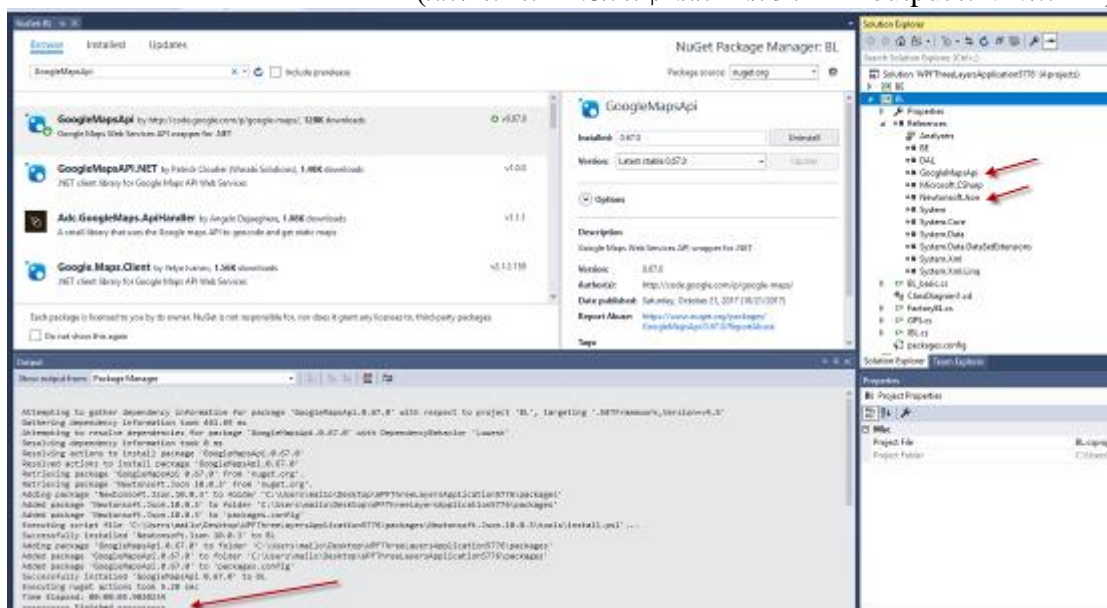
נבחר ב Google maps api web service
ואחרי זה נלחץ על install



בחלון הבא נראה שאמורים להתווסף לנו שני קבצי DLL ונלחץ על OK



בסופו של דבר נראה שנוספו לנו שני קבצי DLL לפרויקט (גם בחלונית ה output יש מידע האם ההתקנה הסתיימה בהצלחה)



דוגמה בסיסית, לחישוב מרחק הליכה בין שתי כתובות:

```
public static int CalculateDistance(string source, string dest)
{
    var drivingDirectionRequest = new DirectionsRequest
    {
        TravelMode = TravelMode.Walking,
        Origin = source,
        Destination = dest,
    };

    DirectionsResponse drivingDirections = GoogleMaps.Directions.Query(drivingDirectionRequest);
    Route route = drivingDirections.Routes.First();
    Leg leg = route.Legs.First();
    return leg.Distance.Value;
}
```

GoogleMaps.Directions.Query מקבל אובייקט מסוג DirectionsRequest שמכיל מוצא ויעד ומחזיר אובייקט מסוג DirectionsResponse שמציין את תוצאת המסלולים

עבור אובייקט מסוג DirectionsResponse ניתן לקבל את כל המסלולים בעזרת המאפיין Routes במקרה אנחנו רוצים את הראשון האפשרי.

מהמסלול נקבל את כל נקודות העצירה שלנו (Legs) שלמעשה היה ניתן להגדיר יותר מנקודת עצירה אחת, אבל זה לא רלוונטי לפרויקט שלנו (לדוגמה, אני רוצה לנסוע למכון ולעבור דרך תחנת דלק שאני אוהב לפני שאני מגיע למכון, אז במקרה כזה יהיו לי שני legs בתוצאת המסלול) כמובן אני בוחר את העצירה הראשונה, שזה למעשה היעד.

מהתוצאה ניתן לקבל את המרחק בעזרת המאפיין Distance. למעשה ניתן לקבל גם את הזמן ועוד נתונים על המסלול ...

תאריכי הגשות:

- הגשה 1 – חלק א עד שלב ג. תאריך הגשה: 10.12 יום ראשון כ"ב כסלו
- הגשה 2 – חלק א שלב ד. תאריך הגשה: 4.1 יום חמישי י"ז טבת
- הגשה 3 – לאחר חלק ג' סופי – **תאריך הגשה** : מפגש אחרון בסמסטר

(ההגנה בשבוע האחרון בשעות המעבדה). לאחר סיום הסמסטר לא יתבצעו בדיקות.