

Delegates

Delegate הוא מצביע לפונקציות. נשתמש בו בכדי להצביע על פונקציות ולהפעיל אותן. שימוש ב delegate נעשה בשלושה שלבים (ודי בדומה לשימושים שעשינו במחלקות).

1. Declaration

2. Instantiation

3. Invocation

בואו נסביר את השלבים השונים ונדגים אותם:

1. Declaration - הצהרה על חתימה מסויימת של פונקציות שהמופעים שלנו יצביעו עליה. עושים את זה כך (בדרך כלל מחוץ ל class מסויים):

```
public delegate void MyDelegate();
```

2. Instantiation – בתוך מתודה נוכל לאתחל מופע של המחלקה, ולהצביע על פונקציה שאותה אנחנו רוצים להפעיל דרך המצביע. אפשר להצביע על מתודה סטטית או מתודה ששייכת למופע מסויים- העיקר, שהמתודה עונה על החתימה שלנו ועם אותו ערך מוחזר.

```
public static void Stam()
{
    Console.WriteLine("X={0}", x++);
}
```

```
//Points on static method
MyDelegate myDelegate1 = new MyDelegate(Stam);
myDelegate1(); //this is step 3 - Invocation
```

```
//Points on instance method
MyDelegate myDelegate2 = new MyDelegate(new Calculations().StamInstance);
```

ניתן להגדיר את המופע של ה delegate כשדה במחלקה, כמאפיין, וניתן גם להגדירו כפרמטר של מתודות. לדוגמה:

```
public static LogDelegate Log { get; set; } //Property of type delegate
```

```
/*delegate parameter exsample*/
/// <summary>
/// Delete student from college, log when done
/// </summary>
/// <param name="student"></param>
public static void Remove(Student student, LogDelegate logParam)
{
    List.Remove(student);
    //Logger.Log("Added " + student.FirstName);
    logParam("Remove " + student.FirstName);
}
```

3. Invocation – הפעלה של הפונקציה

```
myDelegate1(); //this is step 3 - Invocation
```

נכון להקפיד לפני הפעלת המתודות לבדוק שה delegate לא עם הערך null (הוא לא מצביע לשום פונקציה, ואם ננסה להפעילו, נקבל Exception). לדוגמה:

```
if (logParam != null)
{
    logParam("Remove " + student.FirstName);
}
```

אפשרי לשרשר מתודות שה delegate שלנו מצביע עליהם (זה פעולה שנקראת Combine). אנחנו נשתמש כמעט תמיד לביצוע השרשור באופרטור +=.

```
LogDelegate logDelegate= new LogDelegate(new ConsoleLogger().Log);
logDelegate += new FileLogger().Log;
```

בדוגמה הקודמת, כאשר נבצע את ה Invocation שני המתודות יפעלו.

כדאי להסתכל עם dissembler ולראות איך הקוד של delegate מקומפל למחלקה שיורשת מ MulticatDelegate
כדאי לעבור על הפונקציות הסטטיות והאחרות של ה Delegate שיצרנו

שימוש ב delegates די דומה לשימוש ב interfaces ומאפשר לנו פעמים רבות לבצע "היפוך שליטה" –
"Inversion of Control - IoC", ולאפשר שימוש בלוגיקה שיעבירו אלי בלי שהמחלקה שמפעילה את הפונקציה
מתחייבת למימוש קונקרטי, אנחנו רק מתחייבים לחתימה

תרגילי כיתה ובית

1. הגדר Delegate בשם MyFirstDelegate בלי פרמטרים שמחזירה Void.
צור מופע שלו והצבע על פונקציה שמדפיסה למסך "Shalom".
צור מופע נוסף, הצבע על פונקציה Print2 שמדפיסה "Another function called with same delegate".
2. צור Delegate בשם Greet שמקבל string.
צור מופע של Greet, הצבע על מתודה שמקבלת שם ומדפיסה "Shalom {name}".
צור מופע נוסף, הצבע על מתודה שמדפיסה "Welcome {name}".
3. נסה להצביע עם Greet על Print2 (של שאלה 1), האם זה עבד? מדוע?
4. צור מתודה Add שמקבלת 2 מספרים ומחברת אותם. נכתוב גם Sub שמקבלת שני מספרים ומחסירה את השני מהראשון. ניצור 2 כפתורים ב WinForms, הראשון יעשה Add תוך שימוש במופע של delegate, השני יחסיר תוך שימוש במופע אחר של אותו Delegate.
5. צור מחלקה שמנהלת רשימת סטודנטים StudentsList (מייצג List עם מתודות הוספה, מחיקה), כתוב מתודה Filter שתקבל delegate ותחזיר רשימת סטודנטים שעונים על קריטריון מסוים.
6. צור ב StudentsList מתודה Contains שמקבלת delegate. המתודה תאפשר לבדוק אם סטודנט שעונה על קריטריון מסוים נמצא ברשימת הסטודנטים שלנו.
רמז: למתודה יש פרמטר delegate, והמתודה תקבל פונקציה שמקבלת סטודנט ומחזירה bool.
עוברים במתודה Contains על כל הסטודנטים, מפעילים עליהם את הפונקציה שלנו, ומחזירים true למי שעונה על הקריטריון שהוגדר בפונקציה שקיבלנו, כלומר, מי שהמתודה שקיבלנו בארגומנט החזירה עלי true.
7. בהמשך לשאלה הקודמת, כתוב קוד שמשתמש ב Contains לבדוק שיש תלמידים שהציון שלהם מתחת ל 50%.
8. השתמש ב Contains ובדוק אם יש במכלל תלמיד שהשם משפחה שלו והשם הפרטי שלו מתחילים באותה האות.
לסטודנט יש מאפיינים: גיל, ציון, סכום שצריך לשלם, וסכום ששולם.
9. צור ב StudentsList מתודה Sum שמאפשרת להחזיר סיכומים שונים שנבצע על רשימת הסטודנטים (הסיכומים יוחזרו במספרים שלמים - int).
ניחשתם נכון שהמתודה Sum מקבלת delegate (אבל, צריך לחשוב מה החתימה של ה delegate)?
a. באמצעות המתודה Sum והמתודות שנעביר לה כארגומנט, נדפיס את סיכום כל הסכומים ששולמו למכללה.
b. נדפיס גם את כל הסכומים שהסטודנטים עדיין חייבים למכללה (באמצעות מתודה אחרת שעונה לחתימה...)
10. צור ב StudentsList מתודה Average (שמקבלת delegate – עם חתימה שתחשבו עליה) שמאפשרת להחזיר ממוצעים שונים של ממדים שונים שנבצע על רשימת הסטודנטים (הממוצעים יוחזרו במספרים ממשיים - decimal).
a. העבר פונקציה מתאימה והדפס את ממוצע הגיל.
b. העבר פונקציה מתאימה והדפס את ממוצע הציונים.

<https://github.com/efraimt/Hackermes> דוגמאות וכולל דוגמה שקרובה למה שנעשה ב"עולם האמיתי"