

זהו דף שהכנתי לתלמידים ללימוד עצמי של עצמים ומחלקות לאחר שהוצגו מספר דוגמאות לכיתה ובכל זאת התלמידים לא הרגישו בטוחים עם הנושא, אפשר בקלות לגזור מכאן גם דפי עבודה למעבדה.

מתאים לאחר שהתלמידים יודעים את רוב החומר של יסודות 1+2.

עקרונות לתכנות מונחה עצמים

מהו עצם?

בהקשר של מדעי המחשב עצם יהיה כל דבר הניתן לתאר אותו באמצעות **תכונות** ובאמצעות **פעולות**.

דוגמא – **תלמיד** הוא עצם המאופיין ע"י -

התכונות: שם, ת.ז., כיתה, רשימה של מקצועות הלימוד. כמובן שניתן להוסיף עוד ועוד תכונות. (שימו ♥ - כל תכונה היא למעשה משתנה מטיפוס כלשהו, שם הוא string, רשימת המקצועות יכולה להיות מערך של string, כיתה היא מספר שלם וכך הלאה).

הפעולות של העצם **תלמיד** יהיו אוסף פעולות שניתן **להפעיל** על עצם מטיפוס תלמיד, לדוגמא – פעולה **הבונה** עצם מטיפוס תלמיד, פעולה זו קובעת בד"כ את הערכים של התכונות של העצם, פעולה **המתארת** את העצם, פעולה **המחזירה** ערך של תכונה מסוימת (למשל: החזר-שם-תלמיד), פעולה המשנה ערך של תכונה מסוימת (למשל: הוסף-מקצוע-לימוד).

כיצד מגדירים עצם בשפה?

כדי להגדיר עצם יש לכתוב **מחלקה** – זהו למעשה קובץ הכולל בתוכו את ההגדרה של התכונות והפעולות.

את המלקה נהוג לייצג באמצעות טבלה המתארת את התכונות והפעולות. בהמשך נראה כיצד לתאר מחלקה שאנו כותבים ב-C# - במקרה הזה נשתמש בשמות הפעולות והתכונות כפי שיופיעו בתוכנית עצמה.

ייצוג עצם בתרשים:

העצם	תלמיד
תכונות	שם ת.ז. כיתה מקצועות לימוד
פעולות	בנה-תלמיד שנה-שם הוסף-מקצוע האם-לומד-מחשבים באיזה-כיתה

לכל תכונה אנו צרכים לקבוע מה **הטיפוס** שלה (int, string וכדומה) וגם קובעים הרשאת גישה – כלומר מי יכול לגשת לתכונה. בד"כ התכונות הן בהרשאת פרטית (private) – כלומר אין אפשרות גישה ישירה לתכונה עצמה (אלא באמצעות פעולה של המחלקה), ביחידה זו עוסקים בתכונות בהרשאת גישה פומבית הסבר בהמשך ©

לכל פעולה אנו צרכים לקבוע מה הפעולה מקבלת (הפרמטרים הרשומים בסוגרים) האם הפעולה תחזיר ערך או שתהיה פעולת void.

כיצד משתמשים בעצם - דוגמא עם המחלקה Cat

המחלקה לבדה מגדירה את הטיפוס אך לא מבצעת שום דבר כל עוד לא **מזהירים בתוכנית הראשית** על משתנים שהם מטיפוס של המחלקה. אחרי שהצהרנו על המשתנים האלו יצרנו **מופע של העצם**. עם המופע הזה אפשר להשתמש עם הפעולות של המחלקה.

הגדרת עצם

דוגמא – למשל, אם כתבנו מחלקה Cat.cs, נניח כי התכונות של החתול הם: שם (string), גובה (int), סוג (string).

תרשים המחלקה:

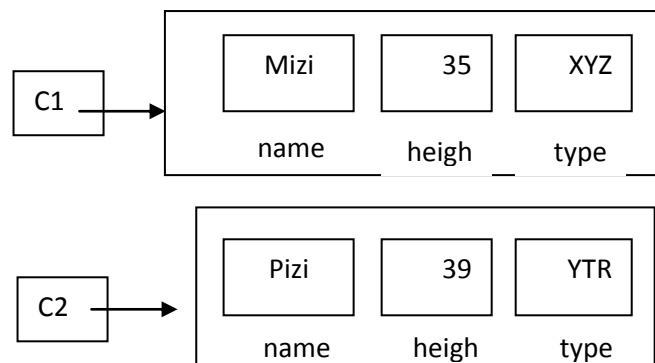
העצם	Cat
תכונות	שם גובה סוג
פעולות	בנה-חתול אחזר-שם אחזר-גובה אחזר-סוג שנה-שם שנה-גובה שנה-סוג האם-החתול-גבוה

בתוכנית הראשית נצהיר על חתולים.. ההצהרה כוללת גם קביעה של ערכי התכונות של החתול.

```
Cat c1, c2 = new Cat ("Mizi", 35, "XYZ");
```

```
Cat c1, c2 = new Cat ("Pizi", 39, "YTR");
```

נוכל לתאר את העצמים כך:



כמובן שאפשר לייצג את התכונות ע"י משתנים בתוכנית ולכתוב כך:

```
int h = .... // קלט או בכל דרך אחרת
```

```
string name=..... , type=.....
```

```
Cat c3 = new Cat (name, h, type);
```

פניה לתכונות של העצם

לתכונות של העצם נפנה באמצעות העצם שהם שייכות לו – ובאמצעות הפעולות המיועדות לכך.

אנחנו עוסקים בתכונות שהן פרטיות (privat) ולכן הגישה אליהן היא ע"י פעולות המוגדרות מראש במחלקה. ככלל לכל תכונה נבנה פעולה המאחזרת את המידע שלה (פעולות Get) – שם הפעולה יהיה המילה Get + שם התכונה, למשל GetType(). נסתכל בדוגמא:

```
int h = c1.GetHigh();
```

```
string catName = c2.GetName();
```

באופן כללי, בהוראות הללו – המבנה הוא **<שם התכונה>.Get** **<שם המשתנה>**

באופן דומה יהיה לנו גם פעולות **קובעות** (הקובעות ערכים חדשים לתכונות) אלו הן פעולות ה- Set והן יקבלו כפרמטר ערך חדש לתכונה. למשל:

```
c1.SetType("QWE");
```

```
c3.SetName("Luci");
```

הפעלת פעולות של העצם

הפעולות של העצם הן כל אותן הפעולות שכתבנו במחלקה (למשל בקובץ Cat.cs) – לכל הפעולות קבענו הרשאה: public – כך שנוכל להשתמש בפעולות מתוך התוכנית הראשית. הפעלת הפעולות נעשית באופן דומה לפעולות של המחלקה Random ו-String.

המבנה הכללי הוא, **(רשימת הפרמטרים) <שם התכונה>. <שם המשתנה>** למשל:

```
if (c3.IsTall()) Console.WriteLine("Yes");
```

נראה דוגמאות נוספות בהמשך.

בקובץ זה מוצגות דוגמאות ל-4 מחלקות, עבור כל אחת מהן הצגנו את המחלקה עצמה (הקובץ cs ששמו כשם העצם של המחלקה) וגם את השימוש במחלקה מתוך המחלקה הראשית.

בחלק מהדוגמאות יש גם שאלות (חלקן פתורות) לדוגמא.

אנא עיינו ביסודיות בדוגמאות ונסו מזלכם בפתרון התרגילים.

דוגמא מס' 1 – המחלקה Time

לפניך המחלקה Time המגדירה עצם המאופיין ע"י התכונות- דקות ושעות

hour ו- minute הן **תכונות** של המחלקה

הן SetHour, SetMinutes, GetHore, GetMinutes, MinuteToMid, Msg **פעולות** של המחלקה

כעת נראה גם את תרשימים המחלקה:

Time		העצם
int hour int minutes		תכונות
Time (int h, int m)	פעולה בונה עצם מטיפוס Time	פעולות
int GetHour()	פעולה המאחזרת את ערך התכונה hour	
int GetMinutes()	פעולה המאחזרת את ערך התכונה minutes	
void SetHour(int h)	פעולה הקובעת (משנה) את ערך התכונה hour	
void SetMinutes (int m)	פעולה הקובעת (משנה) את ערך התכונה minutes	
string ToString ()	פעולה המחזירה מחרוזת שהיא תאור של העצם	
int MinutsToMid ()	פעולה המחזירה כמה דקות נשארו עד חצות לילה	
string Msg ()	פעולה המחזירה הודעה בהתאם לשעה ביום (בוקר טוב/ לילה טוב וכו'....)	

```
public class Time
{
    public int hour = 0; // שעות
    public int minutes = 0; // דקות
    // פעולה בונה
    public Time(int h, int m)
    {
        this.hour = h;
        this.minutes = m;
    }
    // פעולות קובעות
    public void SetHour(int h)
    {
        this.hour = h;
    }
    public void SrtMinutes(int m)
    {
        this.minutes = m;
    }
    // פעולות מאחזרות
    public int GetHour()
    {
        return this.hour;
    }
    public int GetMinutes()
    {
        return this.minutes;
    }
    // תאור של העצם
    public override string ToString()
    {
        return hour + ":" + minutes;
    }

    // הצות ועד מסויימת משעה נותרו דקות כמה שתחזיר פעולה
    public int MinuteToMid()
    {
        return (23 - hour) * 60 + (60 - minutes);
    }
    // טוב יום/טוב לילה/טוב בוקר הודעה המחזירה פעולה
    public string Msg()
    {
        if (hour > 6 && hour < 12) return "Good
Morning";
    }
}
```

```
        if (hour > 17 && hour < 22) return  
        "Good Evening";  
        return "Have a nice day";  
    }  
}  
}
```

אחרי שכתבנו מחלקה אפשר להשתמש בה כדי להגדיר **עצמים** מטיפוס Time - אלו הם משתנים בתוכנית הראשית, כל משתנה שומר מידע עבור השעות והדקות של זמן נתון.

לפניך קטע בתוכנית הראשית המגדיר שני משתנים מטיפוס Time קובע את ערכי התכונות שלהם מדפיס את תאור הזמן וידפיס הודעה מי הזמן המאוחר יותר

```
int h,m;  
Time time1, time2;  
h = int.Parse(Console.ReadLine());  
m = int.Parse(Console.ReadLine());  
time1 = new Time(h,m);  
h = int.Parse(Console.ReadLine());  
m = int.Parse(Console.ReadLine());  
time2 = new Time(h,m);  
Console.WriteLine("Time 1 = " + time1);  
Console.WriteLine("Time 2 = " + time2);  
  
int mid1 = time1.MinuteToMid();  
int mid2 = time2.MinuteToMid();  
  
if (mid1 > mid2)  
    Console.WriteLine("Time1 is earlier than Time2");  
else  
    Console.WriteLine("Time2 is earlier than Time1");
```

עברו על קטע התוכנית ושימו לב כיצד נעשה השימוש בפעולות של המחלקה Time.

שאלות אפשריות המתייחסות למחלקה Time:

1. כתוב במחלקה Time פעולה המקבלת מספר שלם שהוא שעה ביום והפעולה תחזיר true אם השעה היא אותה שעה כמו השעה של העצם Time אחרת תחזיר הפעולה false
2. נתונים 3 משתנים מטיפוס Time, לכל אחד הדפס את תיאור הזמן ואת ההודעה המתאימה
3. נתונים 3 משתנים מטיפוס Time, הצג כפלט כמה מבניהם מייצגים שעה המתרחשת בלילה (בין 22:00 ל- 5:00)

תשובות:

1.

```
// פעולה המשווה בין השעה המתקבלת כפרמטר לשעה שהיא התכונה של העצם
public bool IfSameHour(int ho)
{
    if (this.hour == ho) return true;
    else return false;
}
```

2.

```
Console.WriteLine(time1 + " " + time1.Msg());
Console.WriteLine(time2 + " " + time2.Msg());
Console.WriteLine(time3 + " " + time3.Msg());
```

3.

```
int kama = 0;
if (time1.Grthour() >= 21 || time1.Grthour() < 5) kama++;
if (time2.Grthour() >= 21 || time2.Grthour() < 5) kama++;
if (time3.Grthour() >= 21 || time3.Grthour() < 5) kama++;
```


דוגמא מס' 2 – המחלקה Point

Point הוא עצם המגדיר נקודה במערכת הצירים – לנקודה יש שתי תכונות x, y המגדירות את הנקודה במערכת.

תרשים המחלקה Point

Point	העצם
double x double y	תכונות
Point (double x, double y)	פעולות
void SetX(double x)	
void SetY(double y)	
double GetX()	
double GetY()	
string override string ToString ()	
double Dest ()	
int WhichQu ()	

נתבונן במחלקה Point – הקובץ Point.cs – (שימו לב גם להערות)

```

class Point
{
    privat double x;
    privat double y;
    // פעולה הבונה את העצם וקובעת לו ערכים התחלתיים
    // שימו לב - הפעולה מקבלת את השם של המחלקה וגם אין לה ערך מוחזר
    // המבנה הזה ייחודי לפעולה הבונה
    public Point(double x, double y)
    {
        // היא מילה שמורה this //
        השימוש שלה הוא פניה לעצם שהזמין את הפעולה //
        - התכונה איקס של העצם שהזמין את הפעולה x //
        this.x = x;
        this.y = y;
    }
    // פעולות קובעות - פעולות Set
    public void SetX(double x)
    {
        this.x = x;
    }

    public void SetY(double y)
    {
        this.y = y;
    }
    // פעולות מאחזרות - פעולות Get
    public double GetX()
    {
        return this.x;
    }
    public double GetY()
    {
        return this.Y;
    }

    public override string ToString()
    {
        string str;
        str = "(" + this.x + " , " + this.y + ")";
        return str;
    }
    public double Dest()
    {
        double d;
        // פיתגורס במשפט שימוש
        d = Math.Sqrt(x * x + y * y);
        return d;
    }
    public int WhichQu()
    {
        if (x >= 0 && y >= 0) return 1;
        if (x >= 0 && y < 0) return 2;
        if (x < 0 && y >= 0) return 3;
        return 4;
    }
}
    
```

שאלות המתייחסות למחלקה Point:

1. כתוב קטע תוכנית היוצר 2 נקודות המקבלות את ערכיהן מהקלט
2. הדפס את המרחק של כל נקודה מראשית הצירים
3. כתוב פעולה המקבלת (בתוכנית הראשית) שתי נקודות ומחזירה True אם המרחק בין שתי הנקודות מהראשית שווה ואחרת הפעולה תחזיר False
4. כתוב פעולה במחלקה Point המחזירה true אם בנקודה הערך של x שווה לערך של y

תשובות:

1+2

```
double x, y;
Point pointA, pointB;
x = double.Parse(Console.ReadLine());
y = double.Parse(Console.ReadLine());
pointA = new Point(x, y); //
x = double.Parse(Console.ReadLine());
y = double.Parse(Console.ReadLine());
pointB = new Point(x, y); //
```

```
Console.WriteLine(pointA ());
Console.WriteLine(pointB ());
```

```
double destA, destB;
destA = pointA.Dest();
destB = pointB.Dest();
Console.WriteLine(destA);
Console.WriteLine(destB);
```

3.

```
static bool IfSameDest(Point p1, Point p2)
{
    double d1, d2;
    d1 = p1.Dest();
    d2 = p1.Dest();
    if (d1 == d2) return true;
    else return false;
}
```

הפעולה כתובה כמו כל פעולה שאתם מכירים, הפעולה מקבלת שתי נקודות, כלומר שני פרמטרים **מטיפוס** Point.

4. הפעולה במחלקה Point (כלומר כתובה בתוך הקובץ Point.cs)

```
public bool IfSame ()
{
    return x == y;
}
```

דוגמא מס' 3 – המחלקה Targil

המחלקה Targil מתארת תרגיל חשבוני בין שני מספרים שלמים. התכונות של המחלקה יהיו – שני המספרים והפעולה המתבצעת בניהם (חיבור, חיסור, ...)

Targil	העצם
int n1 char op int n2	תכונות
Targil (int num1, int num2, char op)	פעולה הבונה עצם מטיפוס Targil וקובעת את ערכי התכונות שלו
void SetN1(int n1)	פעולה המשנה את ערך התכונה num1
void SetN2(int n2)	פעולה המשנה את ערך התכונה num2
void SetOp(char op)	פעולה המשנה את האופרטור של התרגיל (אופרטור – /, %, *, -, +)
int GetN1()	פעולה מחזירה את ערך המספר הראשון
char GetOp()	פעולה מחזירה את ערך התו שהוא הפעולה של התרגיל
int GetN2 ()	פעולה מחזירה את ערך המספר השני
string override string ToString ()	פעולה המתארת את העצם Targil
int Result ()	פעולה המחזירה את תוצאת החישוב של התרגיל (הפתרון)

שימוש במחלקה Targil בתוכנית הראשית
לפניך קטע תוכנית –

- a. מה יהיה הפלט של הקטע ?
b. תאר את העצמים t1,t2,t3 בתום הקטע.

```
Targil t1, t2, t3;
t1 = new Targil (4, 6, '+');
t2 = new Targil (10, 3, '%');
t3 = new Targil (5, 7, '*');
Console.WriteLine (t1);
Console.WriteLine (t2);
Console.WriteLine (t3);
int sum = t1.Result () + t2.Result ();

Console.WriteLine (sum);
t1.setN1 (6);
t2.SetOp ('-');
t3.SetN1 (10);
t3.SetN2 (12);
Console.WriteLine (t1);
Console.WriteLine (t2);
```

```
Console.WriteLine(t3);
```

הפלט של התוכנית:

4+6=10

10%3=1

5*7=35

11 זה הערך של המשתנה sum שהצבנו בו את הסכום של תוצאת שני התרגילים

6+6=12

10-3=7

10*12=120

תארו העצמים יעשה באמצעות תרשים המלבנים, בדומה לדוגמא של העצם .cat

נציג כאן גם את המימוש של המחלקה הזו למרות שאין הכרח לראות את המחלקה עצמה כדי להשתמש בה.

```
class Targil
{
    public int num1;
    public char op;
    public int num2;

    public Targil(int n1, int n2, char ch)
    {
        this.num1 = n1;
        this.num2 = n2;
        this.op = ch;
    }

    public void SetN1(int n1) { this.num1 = n1; }
    public void SetN2(int n2) { this.num1 = n2; }
    public void SetOp(char ch) { this.op = ch; }

    public int GetN1() { return this.n1; }
    public int GetN2() { return this.n2; }
    public char GetOp() { return this.op; }

    public int Result ()
    {
        // נעשה שימוש בהוראה switch - לא למדנו אותה ואין חובה
        // להשתמש בה, אפשר להחליף את הקטע הזה עם הוראת .if
        int res=0;
        switch (op)
        {
            case '+': res = this.num1 + this.num2; break;
            case '-': res = this.num1 - this.num2; break;
            case '*': res = this.num1 * this.num2; break;
            case '/': res = this.num1 / this.num2; break;
            case '%': res = this.num1 % this.num2; break;
        }
        return res;
    }

    public override string ToString()
    {
        return this.num1 + " " + this.op + " " + this.num2 +
            " = " + this.Result();
    }
}
```

דוגמא מס' 4 – המחלקה Malben

לטיפוס מלבן יש שתי תכונות – אורך ורוחב. כאשר יוצרים מלבן חדש אנו נרצה לקבוע את המידות של המלבן. הפעולות שנרצה לעשות עם המלבן יהיו – חישוב שטח, חישוב היקף, שינוי רוחב המלבן, שינוי אורך המלבן וחישוב אלכסון של המלבן

מטעמי פשטות ונוחות קבענו את הרוחב והאורך להיות מספרים שלמים.

Malben		העצם
int a		תכונות
int b		
Malben (int a, int b)	פעולה הבונה את העצם	פעולות
string GetMalben()	פעולה המחזירה תאור של המלבן	
void SetA (int a)	פעולה המשנה את היקף המלבן	
void SetB (int b)	פעולה המשנה את רוחב המלבן	
int Perimeter ()	פעולה המחזירה את היקף המלבן	
int Area ()	פעולה המחזירה את שטח המלבן	
double Diagonal()	פעולה מחזירה אורך האלכסון של המלבן	
bool Equal (Malben m)	פעולה תחזיר true אם שני מלבנים שווים ואחרת תחזיר false	

לפניך המחלקה, השלם את ההוראות החסרות:

```
class Malben
{
    public int a;
    public int b;

    public Malben(int a, int b)
    {
        _____
        _____
    }

    public string GetMalben ()
    {
        _____
        _____
        _____
    }
}
```

```
    }

    public void SetA(int a)
    {
        this.a = a;
    }

    public void SetB(_____)
    {
        _____
    }

    public int Perimeter()
    {
        return 2 * a * b;
    }

    public int Area()
    {
        _____
    }

    public double Diagonal()
    {
        return Math.Sqrt(a * a + b * b);
    }

    public bool Equal(Malben m)
    {
        if (this.a == m.a && this.b == m.b) return true;
        else return false;
    }
}
```

הסבר על הפעולה Equal:

הפעולה הזו מקבלת כפרמטר משתנה מטיפוס Malben וכך היא יכולה להשוות בין שני מלבנים, המלבן הראשון הוא זה שהועבר כפרמטר והמלבן השני הוא המלבן שמפעיל את הפעולה.

נראה שימוש של המחלקה של הפעולה בתוכנית הראשית:

נניח כי יש שני עצמים מטיפוס מלבן mal1 ו-mal2, נכתוב קטע תוכנית שידפיס YES אם שני המלבנים שווים ואחרת ידפיס NO:

```
if (mal1.Equal(mal2)) Console.WriteLine("YES");
    else Console.WriteLine("NO");
```

שימו לב לאופן הזימון של הפעולה Equal - mal1 הוא העצם המפעיל את הפעולה ו-mal2 הוא העצם המועבר לפעולה כפרמטר.

סיכום

הצגנו את נושא העצמים והמחלקות – הסברנו מהו למעשה עצם – מהן התכונות של העצם והפעולות שניתן לבצע עם העצמים.
הדגמנו בצורה מפורטת את כתיבת מחלקה המגדירה עצם בשפה וכן את השימוש בעצמים מתוך התוכנית הראשית.

ראיתם 4 דוגמאות של עצמים שונים: Point, Malben, Targil, Time

מתוך הדוגמאות הצגנו מספר שאלות, שאלות כאלו ודומות תצטרכו לדעת לפתור.

דף עבודה

Child הוא טיפוס המתאר ילד ע"י שתי תכונות: שם ושנת לידה. הפעולות שניתן לבצע על העצם Child הן:

- בניית עצם מטיפוס Child וקביעת הערכים של התכונות
- שינוי שם
- שינוי שנת לידה
- תיאור העצם Child
- האם שני ילדים נולדו באותו השנה
- באיזה שנה הילד יחגוג/חגג יום הולדת 16

1. השלם את התרשים

Child	העצם
	תכונות
	פעולות

2. כתוב את המחלקה Child (הקובץ Child.cs)
3. הגדירו בתוכנית הראשית מערך של ילדים yeldim בגודל 10.
4. בנו את המערך (קצת עבודה אבל לא נורא...) וענו על השאלות הבאות:
 - a. כתוב קטע תוכנית שהפלט שלו יהיה כמה ילדים במערך yeldim נולדו אחרי שנת 2000,
 - b. כתוב קטע תוכנית המדפיס את שמות הילדים שנולדו בשנת 1999 או קודם.
 - c. מתברר כי יש טעות בקובץ ולכל הילדים יש שגיאה בשנת הלידה, שנת הלידה הנכונה היא שנה הקודמת לשנה הקיימת. למשל אם רשום לילד ששנת הלידה שלו היא 2003 אז השנה הנכונה היא 2002. כתוב פעולה המקבלת את מערך ומעדכנת אותו
 - d. כתוב פעולה המקבלת את מערך הילדים ושנת לידה, והפעולה תדפיס את שמות הילדים שנולדו בשנה הזו.
 - e. כתוב פעולה המקבלת את המערך ואת ומחזירה כמה ילדים השם שלהם התחיל באות הזו
 - f. כתוב פעולה המקבלת את מערך הילדים ומחזירה
 - g.
 - h. כמה ילדים יש שם המתחיל באות של הילד הראשון במערך