חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"א

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

תכנות מונחה עצמים: משימה מתרחבת

מחלקת Box מדריך למורה

חלק א': מימוש ושימוש – המחלקה Box

חלק ב': מערכי עצמים – מערכי קופסאות

חלק ג': תכונות ומתודות מחלקתיות (סטטיות) – המחלקה Box

חלק ד': ירושה – קופסא צבעונית ColoredBox

כתיבה ועריכה:

גלית גולד-טולדנו

דני קשת

מחלקת Box עריכה: גלית, גרסת סי שרפ: דני

מחלקת Date עריכה: דני, גרסת ג'אווה: גלית

המחלקה Box – מדריך למורה

התרגול מבוסס על ממשק המחלקה **Box** המופיע בספר הלימוד "עיצוב תכנה מבוסס עצמים", עמוד 12.

חלק א' –מימוש ושימוש - המחלקה Box

המחלקה **Box** נבחרה כהדגמה ראשונה כיון שנוח לדמות את המחלקה ל"מכונה המייצרת קופסאות".
השימוש ברעיון "מוחשי" עשוי לסייע להבנת מהות המחלקה.
המורה יפתח את נושא העצמים בדוגמא **Box** כעצם, בהתאם לתכנים וההנחיות שבספר הלימוד, עמודים 9-16. עמודים אלו עוסקים במהות העצם ואופן בו השימוש בו.
ניתן להשתמש במערך שיעור זה כשיעור שני או שלישי בתכנות מונחה עצמים.
לאחר שהתלמידים קיבלו "תחושה", מהו עצם וכיצד משתמשים בו, ניתן לעבור לאופן שבו ממשים עצם.

הכנה לקראת דף העבודה - חלק א'

המורה יממש בכיתה את המחלקה **Box**, באופן חלקי בהתאם לממשק המופיע בעמוד 12.
יש להדגיש כי לכל מחלקה יש "חברי מחלקה" – תכונות ומתודות.
תכונות: יש לפרט את התכונות במלואן.
שיטות: שיטה בונה, שיטות מאחזרות, שיטות קובעות ו- **ToString()**.
המורה יממש את המחלקה באופן חלקי כאשר ידגים כל "סוג" מתודה (למשל, ידגים את אופן הכתיבה של **GetLength()** ולא יממש פעולות מאחזרות נוספות).

1. פתח פרויקט בשם **BoxProject**.
2. כתב את המחלקה **Box** בהתאם לממשק שבעמוד 12.
3. כתב מחלקת בדיקה בשם **BoxText** ובה מתודה ראשית.
   3.1. התכנית תקלוט את מימדיהן של שלוש קופסאות ותייצר אותן (box1, box2 , box3) .
   3.2. **התכנית תדפיס:**
   - אורכה של הקופסה הרחבה ביותר.
   - האם כל הקופסאות זהות במימד האורך.
   - האם כל הקופסאות שונות זו מזו במימד הגובה.
   - האם קיימת **קובייה** בין שלש הקופסאות ?

   חישבו על דרך יעילה לביצוע המשימה. ניתן להוסיף פעולות למחלקה Box.

דרך יעילה תחשב לבניית שיטה במחלקה Box המחזירה 'אמת' את הקופסא היא קובייה, אחרת תחזיר 'שקר'.

```
public bool IsCube()
{
    return (this.length==this.width && this.width==this.height);
}
```

4. הוסיפו לתכנית שתי קופסאות בעלות מימדים זהים box4, box5.
בדקו האם שתי הקופסאות זהות והציגו הודעה מתאימה.
האם ההודעה שהתקבלה נכונה ? _____
מצופה שתלמיד יבדוק שוויון בין הקופסאות באמצעות הפקודה: if (box4==box5)…..
מכיוון שבהוראה זו בודקים שוויון בין ההפניות ולא בין התכונות, התלמיד יקבל הודעה כי הקופסאות אינן זהות.
יש להניח כי בזמן המעבדה יעלו שאלות. בשלב זה, רצוי לא לספק הסברים מלבד העובדה שהתשובה אינה שגויה. הסברים יינתנו בסיום התרגיל.
סעיף זה חושף את התלמיד כבר משלבי הלימוד הראשונים למושג **הפניות והשוואת עצמים**.
כמו כן, הסעיף מתרגל **שיטה המקבלת עצם מאותו הטיפוס** ומשווה אותו עם ה"עצם הנוכחי".

5. חזור למחלקה **Box** והוסף מתודה נוספת בשם **()SetHeight** המשנה את גובה הקופסה כך שיהיה שווה לאורכה.
חזור המחלקה **BoxTest** והפעל את המתודה על אחת מהקופסאות.
הדפס את מימדי הקופסה לאחר השינוי.
האם לצורך הפעולה היה עליך למחוק את המתודה **SetHeight (double height)**? _____
פעולה זו עוסקת בנושא **העמסת שיטות**.

6. חזור למחלקה **Box** והוסף בנאי (פעולה בונה, קונסטרקטור) בשם **()Box** הבונה קובייה בעלת אורך צלע 1 ס"מ.
חזור המחלקה **BoxTest** וצור באמצעותה את הקופסה box6.
הדפס את מימדי הקופסה החדשה שיצרת.
פעולה זו עוסקת בנושא **העמסת שיטות בונות**.
הנח כי בבית המלאכה החליטו לצבוע את הקופסאות. שנה את המחלקה (תכונות ומתודות) בהתאם.
הערה! לצורך שינוי זה, ניתן להניח כי קופסאות שלא ניתן עבורן צבע, צבען יהיה לבן.
צור במחלקה BoxText מספר קופסאות צבעוניות והדפס את תכונותיהן.
מה צבען של הקופסאות box1-box6? בדוק.

דיון

המורה יערוך דיון בכיתה באמצעות בדיקת פתרונות שהציעו התלמידים לסעיפים.
במהלך בדיקת הפתרונות, מומלץ להציג בפני התלמידים פתרון מלא של המחלקה **Box** ו- **BoxTest**, על-מנת שברשות התלמיד תהיה דוגמא בסיסית לאופן בו מממשים ומשתמשים במחלקה.
נושאים ומושגים שילמדו במשימה זו:
מחלקה, חברי מחלקה, שיטה בונה, שיטה בונה מועמסת, שיטות מאחזרות, שיטות קובעות, שיטות מועמסות, השוואת עצמים, הפניות, שיטה המשווה עצם נוכחי עם עצם מטיפוס המחלקה, המתודה equals והמתודה ToString() – חשיפה לחשיבות הופעתן בכל מחלקה.

קוד הפתרון:

```
public class Box
{
	private double length;
	private double width;
	private double height;

	public Box(double length, double width, double height)
	{
		this.length=length;
		this.width=width;
		this.height=height;
	}

	public Box()
	{
		this.length=1;
		this.width=1;
		this.height=1;
	}

	public double GetLength()
	{
		return this.length ;
	}

	public double GetWidth()
	{
		return  this.width;
	}

	public double GetHeight()
	{
		return this.height ;
	}

	public void SetLength(double  newLength)
	{
		this.length=newLength ;
	}

	public void SetWidth(double newWidth)
	{
```

```
            this.width=newWidth ;
        }

        public void SetHeight(double  newHeight)
        {
            this.height= newHeight ;
        }

        public void SetHeight()
        {
            this.height= this.length ;
        }

        public double GetVolume()
        {
            return this.length*this.width*this.height;
        }

        public override string ToString()
        {
            string s="";
            s="Box{"+this.length+","+this.width+","+this.height+"}";
            return s;
        }

        public bool IsCube()
        {
          return (this.length==this.width && this.width==this.height);
        }

        public bool Equals(Box other)
        {
             return (this.length==other.length &&
                     this.width==other.width  &&
                     this.height==other.height);
        }
}

public class BoxTest
{
        public static void Main(string[] args)
        {

            double length,width, height;
```

```
//3.1 סעיף
Console.WriteLine("Enter box1 dimensions:");
length=double.Parse(Console.ReadLine());
width=double.Parse(Console.ReadLine());
height=double.Parse(Console.ReadLine());
Box box1=new Box(length,width,height);
Console.WriteLine("Enter box2 dimensions:");
length=double.Parse(Console.ReadLine());
width=double.Parse(Console.ReadLine());
height=double.Parse(Console.ReadLine());
Box box2=new Box(length,width,height);
Console.WriteLine("Enter box3 dimensions:");
length=double.Parse(Console.ReadLine());
width=double.Parse(Console.ReadLine());
height=double.Parse(Console.ReadLine());
Box box3=new Box(length,width,height);

// 3.2 סעיף
Box maxWidth=box1;
if (maxWidth.GetWidth()<box2.GetWidth())
        maxWidth=box2;
if (maxWidth.GetWidth()<box3.GetWidth())
        maxWidth=box3;
Console.WriteLine(maxWidth.GetLength());
//
if(box1.GetLength()==box2.GetLength() &&
  box2.GetLength()==box3.GetLength())
        Console.WriteLine("Same length.");
else
        Console.WriteLine("Not the same length.");
if(box1.GetHeight()!=box2.GetHeight() &&
  box2.GetHeight()!=box3.GetHeight() &&
  box3.GetHeight()!=box1.GetHeight())
        Console.WriteLine("Height: All different.");
else
        Console.WriteLine("Height: Not all different.");
if(box1.IsCube()  || box2.IsCube() || box3.IsCube())
        Console.WriteLine("At least one cube.");
else
        Console.WriteLine("No cube.");
Console.WriteLine(box1.ToString());
Console.WriteLine(box2.ToString());
Console.WriteLine(box2.ToString());
//Console.WriteLine(box2); it's ok as well
```

```
            // 4 סעיף
            Box box4=new Box(4,6,8);
            Box box5=new Box(4,6,8);
            if(box4.Equals(box5))
                  Console.WriteLine("Same dimensions.");
            else
                  Console.WriteLine("Not the same dimensions.");

            //5 סעיף
            box4.SetHeight();
            Console.WriteLine(box4.ToString());

            //6 סעיף
            Box box6=new Box();
            Console.WriteLine(box6.ToString());
      }
}
```

דוגמת הרצה (פלט):

```
Enter box1 dimensions:
1
2
3
Enter box2 dimensions:
4
5
6
Enter box3 dimensions:
3
3
3
4.0
Not the same length.
Height: Not all different.
At least one cube.
Box{1.0,2.0,3.0}
Box{4.0,5.0,6.0}
Box{4.0,5.0,6.0}
Same dimensions.
Box{4.0,6.0,4.0}
Box{1.0,1.0,1.0}
```

חלק ב' –מערכי עצמים – מערכי קופסאות

מטרת תרגול זה הינה לאפשר לתלמיד לתרגל מערכי עצמים תוך כדי חזרה על תבניות שנלמדו ביסודות.

פתח את הפרויקט **BoxProject** ובצע את המשימות הבאות:

1. הוסף למחלקה **BoxText** מערך קופסאות a באורך 4.

   הדגשה: תאי המערך a בשלב זה הוא null .

2. קלוט את נתוני הקופסאות.

   בשלב הדיון יש להדגיש את הקצאות מקומות המערך והקצאת הקופסאות (עצמים) ו"השמתם" במערך. יש להדגיש את נושא ה"הפנייה" לקופסא (עצם).

3. בצע את הבדיקות הבאות על המערך a:

   3.1. בדוק האם כל הקופסאות הן "קוביות" והצג הודעה מתאימה.

   תרגול שימוש בפעולות המחלקה (פעולות פנימיות) באמצעות המתודה IsCube().

   3.2. בדוק האם קיימות לפחות שתי קופסאות בעלות מימדים זהים.

   תרגול שימוש בפעולות המחלקה (פעולות פנימיות).

   בדיקת השוויון תעשה באמצעות `Equals(Box other)`. מתודה זו נחשבת לקשה יותר להבנה מהמתודה IsCube() (מהסעיף הקודם), בשל העובדה שלמתודה מועבר פרמטר אחד ולמרות זאת נערכת השוואה בין 2 עצמים.

   הסעיף מתרגל פעם נוספת את רעיון השוואה העצמים.

   3.3. הוסף מתודה המקבלת שתי קופסאות b1,b2 ומחזירה 'אמת' אם נפח הקופסא b1 גדול מנפח הקופסא b2. אחרת, תחזיר 'שקר'.

   כתיבת מתודה חיצונית המבצעת השוואה בין שתי קופסאות באמצעות שימוש במתודה הפנימית GetVolume().

   הסעיף מדגיש את ההבדל בכתיבת מתודה פנימית (בתוך המחלקה Box) כמו GetVolume() הפועלת על העצם this, לעומת מתודה חיצונית הפועלת על הפרמטרים המועברים בחתימה.

   3.4. בדוק, באמצעות המתודה מהסעיף הקודם, האם מערך הקופסאות ממוין בסדר עולה על-פי נפח. הצג הודעה מתאימה.

   הסעיף מתרגל זימון של מתודה חיצונית, בניגוד לזימונים קודמים בתרגיל, בהם זומנו מתודות באמצעות העצם (מונחה עצמים) , מתודות פנימיות.

4. ממש את המתודה הבאה:

   // הפעולה מדפיסה את הקופסאות שבמערך a החל ממקום 0 ועד למקום pos

   // pos>=0

   public static void Print(Box[] a, int pos)

5. הוסף מערך קופסאות נוסף b באורך 4.

   יש להוסיף למערך b רק את הקופסאות שהן "קוביות".

   בסעיף זה יש לצפות ל-2 פתרונות אפשריים:

   פתרון 1 – הפתרון השכיח: העתקת הפניות בלבד.

פתרון 2 – העתקת העצם.

יש לדון בכיתה על שני הפתרונות האפשריים ולשרטט את מצב העצמים בשתי הדרכים.

יש להסביר ולהדגיש מצב של "הפנייה כפולה" וסכנותיה.

הסעיפים הבאים נועדו לעורר שאלות בקרב תלמידים שפתרו באמצעות העתקת הפניות.

* יש להדגיש כי תאים בהם "אין קופסאות" נחשבים לתאים ריקים null שלא ניתן לבצע עליהם פעולות המחלקה (למשל ToString).

6. הדפס את נתוני הקופסאות שבמערך a באמצעות המתודה Print.
7. הדפס את נתוני הקוביות שבמערך b באמצעות המתודה Print.
8. הגדל ב-1 ס"מ את מימדי הקוביות שבמערך b.
9. הדפס את נתוני הקופסאות שבמערך a באמצעות המתודה Print.
10. הדפס את נתוני הקוביות שבמערך b באמצעות המתודה Print.

עיין בפעולות שביצעת בסעיפים 6-10.

האם המערך a עבר שינוי ? לא

האם המערך b עבר שינוי ? כן

האם תוצאות ההדפסה של המערך a בסעיף 6 **שונות** מתוצאות ההדפסה שהתקבלו בסעיף 9 ? כן / לא

אם התקבלו תוצאות שונות, ציין **מהו השינוי** ונסה להסביר **מדוע** :

תוצאות שונות יתקבלו כאשר התלמיד העתיק הפניות ולא עצמים.

אם התקבלו הדפסות זהות, נסה להסביר באיזה מצב תתקבלנה הדפסות שונות ?

הדפסות שונות יתקבלו במקרה שתתבצע העתקת הפניות. במקרה זה תתרחש "הפנייה כפולה" שתעדכן את העצם ה"יחיד" דרך מערך b.

* כפתרון לסעיף יש להעדיף את הפתרון השני בו יועתקו עצמים.

לצורך כך, בשלב ראשון יש ללמד העתקת עצם באמצעות השיטה הבונה הקיימת.

לאחר שהתלמידים הבינו, יש לשכלל את הפתרון באמצעות הוספת שיטה בונה מעתיקה למחלקה **Box** :

```
public Box(Box other)
    {
        this.length= other.length;
        this.width=  other.width;
        this.height= other.height;
    }
```

נושאים ומושגים שילמדו במשימה זו :

שיטה בונה מעתיקה, הפניות, הפנייה כפולה, מתודה חיצונית מול מתודה פנימית, null, השוואת עצמים, this.

חלק ג' –תכונות ומתודות מחלקתיות (סטאטיות) – המחלקה Box

מטרת התרגול היא לאפשר לתלמיד להתנסות תרגול תכונות ומתודות סטאטיות.

התרגיל מאפשר לתלמידים להתנסות בתרגיל ולהסיק מסקנות . אין הכרח ללמד את הנושא לפני ביצוע התרגיל.

ניתן להיעזר או לערוך חשיפה קצרה לנושא באמצעות בספר הלימוד "עיצוב תכנה מבוסס עצמים" – פרק 3, עמ' 57-61.

בתרגיל זה נלמד כיצד לספור את הקופסאות שיצרנו וכיצד נוכל להנפיק עבורן מספר זיהוי ללא חזרות.

לשם כך, פתח את הפרויקט **BoxProject** ובצע את המשימות המפורטות.

1. הוסף למחלקה **Box**:

- תכונה **מחלקתית** (סטאטית) **count**, המונה את מספר הקופסאות (המופעים/העצמים) שנוצרו באמצעות המחלקה.
- תכונה של **מופע** – תכונה המזהה כל קופסא (עצם) על פי מספר זיהוי **id**.

בסעיף זה ניצור את התכונות הדרושות למניית הקופסאות וזיהוי באמצעות "מספר זהות".

כך ייראו תכונות המחלקה:

```
public class Box
{
        private double length;
        private double width;
        private double height;
        private static int count=0;  // מונה הקופסאות מאותחל ל-0
        private int id;
```

2. טיפול **בשיטות בונות**:

יש **למנות** את הקופסאות בעת יצירתן. לפיכך, נמנה את הקופסא שנוצרה בשיטה הבונה.

מכיוון שקיימות שתי דרכים ליצור קופסאות (שתי שיטות שונות), נבצע מנייה בשתיהן.

בנוסף, בכל פעם שאנו מונים, "נשמור" את הערך החדש כ"**מספר זיהוי**" של המופע.

כך ייראו השיטות הבונות:

```
public Box(double length, double width, double height)
{
        this.length=length;
        this.width=width;
        this.height=height;
        Box.count++;
        this.id=Box.count;
}
public Box()
{
        this.length=1;
        this.width=1;
        this.height=1;
        Box.count++;
        this.id=Box.count;
}
```

3. שיטות המאחזרות מידע:

- נוסיף שיטה מחלקתית (סטאטית), המחזירה את מספר הקופסאות שנוצרו(מופעי המחלקה).

הצורך בהוספת מתודה מאחזרת נובע מהגדרת הרשאת גישה private לתכונה count.

```
public static int GetCount()
{
        return Box.count;
}
```

4.

- נוסיף שיטה המחזירה את מספר הזיהוי של העצם.

```
public int GetId ()
{
        return this.id;
}
```

הסמיכות בכתיבת 2 מתודות "שונות באופיין", מדגיש לתלמיד את טכניקת העבודה עם static. בנוסף, קיים קשר רעיוני בין 2 המתודות הנובע מאופן מימוש השיטה הבונה.

5. בדוק את אופן פעולת השיטות שהגדרת במחלקה **BoxText באופן הבא:**

א) שלב במתודה הראשית את שורת הקוד שלפניך ב-3 מקומות שונים (בהתחלה, באמצע ובסוף).

```
Console.WriteLine("מספר הקופסאות שיוצרו עד כה: " + Box.GetCount());
```

ב) המתודה **GetCount** הא **מתודה מחלקתית**.

הסבר מה היא מבצעת ומדוע היא מוגדרת כ"**מתודה מחלקתית**" ?

המתודה מחזירה את מספר המופעים שנוצרו באמצעות המחלקה Box.

המתודה מוגדרת כ"מחלקתית" משם שהיא מתייחסת לתכונה השייכת לכלל המופעים ולא לתכונה השייכת למופע יחיד .

6. ג) הוסף למתודה הראשית את הפעולות הבאות (בסוף):

```
Console.WriteLine ("box1 id"+box1.GetId() );
Console.WriteLine ("box3 id"+box3.GetId() );
```

ד) הדפס את מספרי הזיהוי של הקופסאות המאוחסנות במערך a.

ה) המתודה **GetId** אינה מתודה מחלקתית.

הסבר מה היא מבצעת ומדוע אינה מוגדרת כ"**מתודה מחלקתית**" ?

המתודה מחזירה את מספר הזיהוי של הקופסא/מופע "עליו" היא מופעלת.

המתודה אינה מוגדרת כ"מחלקתית" משם שהיא מתייחסת לתכונה השייכת למופע יחיד ולא לכלל מופעי המחלקה .

ו) **הנח** כי מוסיפים (במתודה הראשית) קופסא נוספת: **Box myBox=new Box(5,10,15);**

רשום ביטוי המציג את **מספר הזיהוי** של הקופסא: _____

רשום ביטוי המציג את **מספר הקופסאות** שיוצרו באמצעות המחלקה Box עד כה:

_____

7. הוסף למחלקה Box מנגנון (תכונה/ות ומתודה/ות) המאפשר לבדוק האם אי-פעם נוצרה קופסא שהיא קובייה.

הערה! יש להתעלם מקופסאות **שלא** נוצרו כקובייה אך במשך הזמן "השתנו" ל"קובייה".

סעיף זה מאפשר לבדוק את מידת הבנתו של התלמיד.

ניתן לדון עם התלמידים לפני ביצוע המשימה או לפתור בפורום כיתתי.

דיון

static הינו נושא מופשט ולכן מעורר קושי. הקושי נובע מהבנת נושא ה"עצמים".

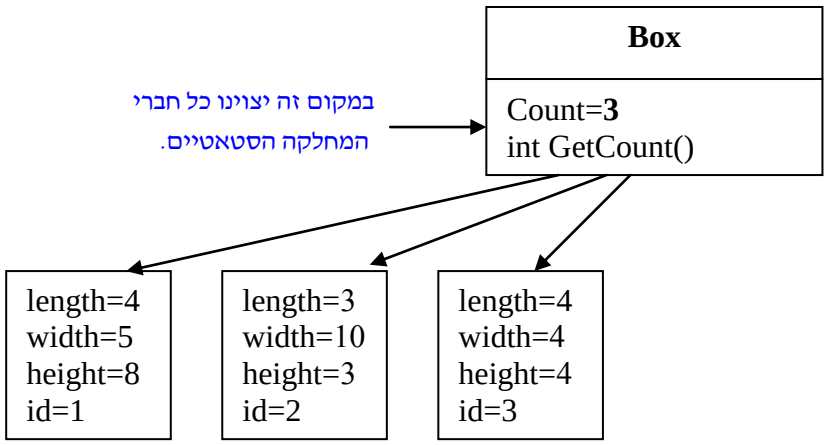
התלמידים שזה עתה נחשפו לנושא ה"עצמים", מתקשים להבין כיצד ניתן להתייחס לתכונה ש"אין לה עצם...".

פעמים רבות תלמידים כותבים נכון ("הבנה טכנית") אך לא מבינים את המהות.

* שימוש במושג "מתודות חיצוניות" כ- "מתודות סטאטיות".

בנקודה זו המורה צריך לזכור לאורך לימוד הנושא כי **המושג "מתודה חיצונית" אינו זהה למושג "מתודה סטאטית"** כיוון שמתודה סטאטית יכולה להיות גם "בתוך המחלקה".

לצורך הבנת המושג "תכונה/מתודה מחלקתית", רצוי לערוך שרטוט של העצמים המתקבלים באופן הבא:



שרטוט זה יכול להמחיש לתלמידים כי חברים סטאטיים יכולים להתקיים גם ללא מופעים.

חלק ד' – ירושה – קופסא צבעונית ColoredBox

התרגיל עוסק בירושה, לאחר לימוד פרק 6 בספר הלימוד "תכנות מונחה עצמים" בשפת ג'אווה, המרכז להוראת המדעים, האוניברסיטה העברית.

הספר יצא לאור רק במהדורת ג'אווה. למרות זאת,הספר הנ"ל מתאים גם ללומדי סישארפ,כיוון שההבדלים אינם רבים,במיוחד בפרקים 1-5.

החל מפרק 6,יש לשים לב באופן מיוחד. מומלץ מאוד "להמיר" את המצגת המצורפת לספר לשפת סי שארפ,כיוון שקיים שוני ניכר,הן בתחביר והן באופן ההתייחסות לירושה בשתי השפות.

התרגיל עוסק מנגנון ירושה, המרות, שימוש חוזר בקוד, העמסת שיטות, דריסת שיטות ופולימורפיזם.

בית המלאכה החליט לצבוע את הקופסאות אותן הוא מייצר.

פתח את הפרויקט **BoxProject** ובצע את המשימות המפורטות.

1. צור מחלקה בשם **ColoredBox** היורשת מהמחלקה **Box**.

   תכונות המחלקה: אורך, רוחב, גובה וצבע.

   ממש את ממשק המחלקה הבא.

הפעולה בונה קופסא צבעונית בעלת אורך length , רוחב width, גובה height וצבע color.	public ColoredBox(double length, double width, double height, string color)
הפעולה בונה קובייה צבעונית בעלת אורך צלע 1 וצבע color.	public ColoredBox(string color)
הפעולה מחזירה 'אמת' אם הקופסה הצבעונית הנוכחית זהה לקופסא הצבעונית הנוספת other. אחרת, תחזיר 'שקר'.	public bool Equals (ColoredBox other)
הפעולה מחזירה את נתוני הקופסא הציבעונית.	public override string ToString()

ענה על השאלות הבאות:

2. השיטה Equals היא מתודה נדרסת או מועמסת ? הסבר.

   Equals היא מתודה מועמסת כיוון שהיא שונה בפרמטרים שהיא מקבלת מהמתודה Equals שהתקבלה בירושה מ-Box.

3. האם עקרון "שימוש חוזר בקוד" בא לידי ביטוי במתודה Equals ?

   ניתן ומומלץ לממש את המתודה Equals אשר במחלקה הנוכחית באמצעות שימוש ב-base, המאפשר להשתמש מחדש בקוד הקיים ב-Equals שבמחלקת הבסיס Box.

```
public bool Equals(ColoredBox other)
{
return base.Equals(other)&& this.color.Equals(other.color);
}
```

4. הנח כי הוגדרה קופסא (לא צבעונית) **Box box=new Box(3,4,5);**

כמו כן הוגדרה הקופסא הצבעונית

ColoredBox clBbox=new ColoredBox(3,4,5,"Red");

בדוק את הפקודות הבאות וציין עבור כל אחת אם היא נכונה או שגויה והסבר את קביעתך:

זימון שיטה	נכון / שגוי	הסבר
clBbox.Equals(box)	נכון	מאחר והפרמטר שמועבר הוא מטיפוס Box, תופעל המתודה equals מתוך Box.
box.Equals(clBbox.)	נכון	מאחר והפעולה equals פועלת על עצם מטיפוס Box, תופעל המתודה מתוך המחלקה Box ותבצע השוואה בין שתי קופסאות (לא צבעוניות).
clBbox.Equals((**ColoredBox**)box)	שגוי	ניסיון שגוי לבצע המרה כלפי מטה מעצם Box ל- ColoredBox.

5. ציין אילו עקרונות תכנות מונחה עצמים ממומשים בשיטה ToString()?

דריסה, שימוש חוזר בקוד (base) ופולימורפיזם (בזימון המתודה).

6. הפעולה הבאה מחזירה 'אמת' אם הקופסה (הצבעונית) הנוכחית זהה לקופסא הנוספת other. אחרת, תחזיר 'שקר'.

```
public bool Equals(Box other)
{
        if(other is coloredBox)
                return this.Equals((ColoredBox)other);//***
         return false;
}
```

7. **הנח** כי הפעולה נוספה למחלקה.

ענה על השאלות הבאות:

- השיטה `Equals` היא מתודה נדרסת או מועמסת? הסבר.

מתודה דורסת כיוון שחתימתה זהה לחתימת המתודה במחלקה Box.

- הסבר את **האופן** בו נבדקת זהות הקופסאות:

המתודה בודקת האם העצם other היא קופסא צבעונית, אם כן עורכת המרה כלפי מטה (ניתן לוותר על ההמרה...) ומזמנת את המתודה Equals המקבלת כפרמטר קופסא צבעונית.

אם העצם other אינו קופסא צבעונית, יוחזר false.

- האם המתודה נחוצה לביצוע המשימה? הסבר.

אין צורך במתודה. המתודה המועמסת Equals (ב- Box וב- coloredBox) מספיק.

- עיין בשורה המסומנת ב- *** בשורה הבאה return this.Equals(other);

האם השורות שקולות ? לא

הסבר: שלא כמו מנגנון הפולימורפיזם הפועל באמצעות זימון שיטה "דרך" עצם, כאשר מועבר העצם other מטיפוס Box כפרמטר לשיטה, השיטה מתייחסת אליו כטיפוס Box ולא מבצעת כל המרה.

8. הוסף למחלקה **BoxText** את העצמים הבאים:

```
ColoredBox clBox1=new ColoredBox(1,2,3,"Red");
ColoredBox clBox2=new ColoredBox(1,2,3,"Red");
ColoredBox clBox3=new ColoredBox(1,2,3,"Blue");
Box box=new Box(1,2,3);
```

הוסף את הפעולות הבאות:

```
1) Console.WriteLine (clBox1.Equals(clBox2));
2) Console.WriteLine (clBox1.Equals(box));
3) Console.WriteLine (box.Equals(clBox1));
4) Box boxTemp=clBox1;
5) Console.WriteLine (boxTemp.Equals(clBox3));
```

הרץ את ההוראות, צפה בפלט והסק מסקנות.

9. מלא את הטבלה שלפניך באופן הבא:

ציין את שם המחלקה שמתודת ה-equals שלה פעלה והסבר מדוע.

מספר שורה	Box / ColoredBox	הסבר
1	**ColoredBox**	`clBox1, clBox2` – עצמים מטיפוס **ColoredBox**.
2	**Box**	`box` **(פרמטר) – עצם מטיפוס** Box.
3	**Box**	`box` **(עליו מזומנת השיטה) – עצם מטיפוס** Box.
5	**ColoredBox**	`clBox3` עצם מטיפוס **ColoredBox**. `boxTemp` – מנגנון הפולימורפיזם מאפשר לזמן את פעולת ה-equals המתאימה לטיפוס ממנו הוקצה העצם המזומן.

מימוש המחלקה ColoredBox

```
public class ColoredBox : Box
{
        private string color;

        public ColoredBox(double length, double width, double height,
                                                                string color)
                :       base( length,  width,  height);
    {
                this.color=color;
        }

        public ColoredBox( string color)  : base();
        {
            this.color=color;
        }

        public bool Equals(ColoredBox other)
        {
                Console.WriteLine("Inside ColoredBox with ColoredBox:");
                 return base.Equals(other)&&
                                                this.color.Equals(other.color);
        }


        public override string ToString()
        {
                return base.ToString() + "color:"+this.color;
        }
}
```