

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 FullDate מדריך למורה בגרסת סי שרפ

כתיבה ועריכה:

גלית גולד-טולדנו

דני קשת

מחלקת Box עריכה: גלית, גרסת סי שרפ: דני

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

כתיבת מחלקה לטיפול בתאריך מלא (FullDate)

משימה חמישית

בנה פרויקט חדש והכנס לתוכו את מחלקת Date
צרו מחלקה בשם FullDate. מחלקה זו מייצגת תאריך המורכב מיום, חודש ושנה (כמספרים שלמים), וכן שעות ודקות לפני שתיגש לבנות את המחלקה החדשה, אילו שינויים תצטרך לעשות במחלקת Date ?

שימו לב לתוספות ושינויים

(תשובה – מכיוון שהמחלקה החדשה יורשת מהישנה, נרצה להשתמש בתכונות של המחלקה הישנה. בהרשאות הגישה הנוכחיות (private) זה לא ניתן, לכן נשנה לכל התכונות את ההרשאה לprotected
דבר נוסף- אחרי שנכתוב את כל הפעולות המבוקשות, נאתר את הפעולות בהן יש הגדרה מחדש/דריסה/overriding ונוסיף בפעולה הנדרסת את המילה virtual (אלא אם כן כבר כתוב שם overriding כמו בפעולה ToString)

משימה ששית

נרחיב עכשיו את המחלקה Date שכתבתם. נוסיף את הפעולות שאנו זקוקים להן בנוסף לאלה שכבר קיימות ב Date
גם כאן עליכם לממש את כל הפעולות הנוספות ולכתוב מחלקת בדיקה הבונה עצמים מטיפוס FullDate ומפעילה את כל הפעולות שמימשתם יש להקפיד שהתלמידים בונים תוכנית בדיקה המשתמשת בפעולות. במשימות הבאות נבנה גם אנחנו תוכנית בדיקה שמראה אספקטים מעניינים של ירושה ופולימורפיזם

Constuctors	
תיאור ה- C'tor	חתימת ה- C'tor
יוצרת עצם מטיפוס Date על פי הפרמטרים הנתונים.	public Date (int day, int month, int year, int hour, int minute)
יוצרת עצם מטיפוס Date המתחל לתאריך 1 בינואר 1970, חצות	public Date ()
יוצרת עצם מטיפוס Date על פי עצם מטיפוס Date המועבר כפרמטר	public Date (FullDate other)
שיטות נוספות	
(חוץ מ- GetMonth(), SetMonth(int month) וכד' שכבר כתבתם)	
תיאור השיטה	חתימת השיטה
מחזירה true אם התאריך תקין : היום הוא ערך שבין 1 למספר הימים המקסימלי בחודש (ע"פ MAX_DAY). החודש הוא בין 1 ל-12. יש להתחשב בשנים מעוברות. וכן שעות בין 0 ל-23 ודקות בין 0 ל-59	public bool IsValid() הרחבה של הבדיקה לשעות ודקות. כאן יש דריסה לכן יש צורך גם בכתיבת overriding וגם הוספת המילה virtual במחלקת הבסיס
מחזירה true אם התאריך קודם לתאריך שהועבר כפרמטר.	public bool IsPrev(FullDate other) כאן ובפעולה הבאה אנחנו מקבלים כפרמטר תאריך מלא, לכן למרות השם הזה, כיון שיש שוני בטיפוס הפרמטרים- מדובר בהעמסת שיטות ולא בדריסה
מחזירה true אם התאריך זהה לתאריך שהועבר כפרמטר.	public bool IsEqual(Date other) כ"ל
מחזירה מחרוזת של התאריך בפורמט הבא : dd/mm/yyyy_hh:MM יש להקפיד על שתי ספרות ליום ולחודש וכן לשעות ודקות גם כאשר ערכן קטן מ10.	public override string ToString() כמובן שיש כאן דריסה של הפעולה במחלקת הבסיס, שהיא עצמה דרסה את אותה פעולה במחלקת object

יש לכתוב תוכנית בדיקה (כלומר מחלקה שיש בה את השיטה Main() בלבד) בנה מספר "תאריכים" והפעל עליהם את הפעולות הנוספות.

הקוד של מחלקת FullDate

```
public class FullDate : Date
{
    int hour;
    int minute;

    public FullDate(int d, int mo, int y, int h, int m)
        : base(d, mo, y)
    {
        this.hour = h;
        this.minute = m;
    }
    public FullDate(FullDate other)
        : base(other.day, other.minute, other.year)
    {
        this.hour = other.hour;
        this.minute = other.minute;
    }
    public FullDate()
        : base()
    {
        this.hour = 0;
        this.minute = 0;
    }
    public override bool IsValid() //since the name is the same and parameters are the same(none),this is
    overriding
    {
        if (base.IsValid() == false)
            return false;
        if (this.hour < 0 || this.hour > 23 || this.minute < 0 || this.minute > 59)
            return true;
        return true;
    }
    public bool IsPrev(FullDate other) //method overloading,it is not overriding
    {
        if (base.IsPrev(other) == true)
            return true;
        if (IsEqual(other) == true)
        {
            if (this.hour < other.hour)
                return true;
            if (this.hour == other.hour)
                if (this.minute < other.minute)
                    return true;
        }
        return false; //else
    }

    public bool IsEqual(FullDate other)//method overloading
```

חומרי עזר שהוכנו ע"י משתתפי קורס מורים מובילים תשע"א

ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

```
{
    if(!this.IsEqual((Date)other) )
        return false;
    if (this.hour == other.hour && this.minute == other.minute)
        return true;
    return false;
}

public override string ToString()
{
    return base.ToString() + "_" + Hosafat0(this.hour) + ":" + Hosafat0(this.minute);
}

//we will add this method later
public string ToStringDateOnly()
{
    return base.ToString();
}
}
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח

משימה שביעית

בנה את תכנית הבדיקה הבאה

```
static void Main(string[] args)
{
    //-----

    FullDate fd2 = new FullDate(2, 5, 1989, 11, 5);

    Console.WriteLine("fd2 is 02/05/1989 11:05");
    Date d4 = new Date(2, 5, 1989);
    Console.WriteLine("fd4 is 02/05/1989 no hours and minutes");

    FullDate fd5=new FullDate(2, 5, 1989, 8, 12);
    Console.WriteLine("fd5 is 02/05/1989 8:12");

    /*1*/ Console.WriteLine("d4 = fd2 ?" + d4.IsEqual(fd2));
    /*2*/ Console.WriteLine("d4 = fd2 ?" + fd2.IsEqual(d4));

    /*3*/ Console.WriteLine("d5 = fd2 ?ofcourse no " + fd2.IsEqual(fd5));
    /*4*/ Console.WriteLine("d5 = fd2 ?      " + ((Date)fd2).IsEqual(fd5));
    /*5*/ Console.WriteLine("d5 = fd2 ?      " + (fd2).IsEqual((Date)fd5));

    Console.WriteLine("_fd2 " + (Date)fd2);

    Console.WriteLine("_fd2 " + fd2.ToStringDateOnly());

    Console.ReadLine();
}
```

לפני שתריץ כתוב במחברתך מה יהיו תוצאות הפלט – 10 שורות הדפסה
בדוק את תשובותיך – האם הן תואמות את מה שציפית?

זה הפלט

```
Console.WriteLine("fd2 is 02/05/1989 11:05");
Date d4 = new Date(2, 5, 1989);
Console.WriteLine("fd4 is 02/05/1989 no hours and minutes");

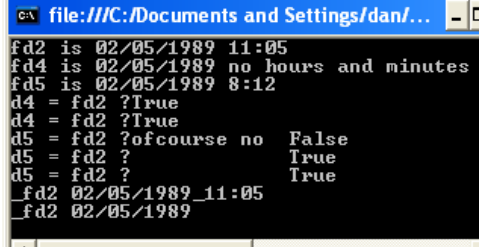
FullDate fd5=new FullDate(2, 5, 1989, 8, 12);
Console.WriteLine("fd5 is 02/05/1989 8:12");

Console.WriteLine("d4 = fd2 ?" + d4.IsEqual(fd2));
Console.WriteLine("d4 = fd2 ?" + fd2.IsEqual(d4));

Console.WriteLine("d5 = fd2 ?ofcourse no " + fd2.IsEqual(fd5));
Console.WriteLine("d5 = fd2 ?      " + ((Date)fd2).IsEqual(fd5));
Console.WriteLine("d5 = fd2 ?      " + (fd2).IsEqual((Date)fd5));

Console.WriteLine("_fd2 " + (Date)fd2);

Console.WriteLine("_fd2 " + fd2.ToStringDateOnly());
```



```
file:///C:/Documents and Settings/dan/...
fd2 is 02/05/1989 11:05
fd4 is 02/05/1989 no hours and minutes
fd5 is 02/05/1989 8:12
d4 = fd2 ?True
d4 = fd2 ?True
d5 = fd2 ?ofcourse no False
d5 = fd2 ?      True
d5 = fd2 ?      True
_fd2 02/05/1989 11:05
_fd2 02/05/1989
```

שתי השורות הראשונות משוות שני אובייקטים מטיפוסים שונים שיש להם יום, חודש ושנה משותפים. אנחנו מקבלים תשובה חיובית לשאלה האם הם זהים. למה?!

`d4.IsEqual(fd2)` הפעולה היא של `d4` ולכן הפעולה היא של תאריך "קצר". אמנם הפרמטר הוא תאריך מלא, אבל לפי הפעולה הוא עובר המרה לא מפורשת כלפי מעלה ואנחנו משווים רק לפי תאריך קצר

`d2.IsEqual(fd4)` הפעולה היא של `d2` ולכן הפעולה היא לכאורה של תאריך "מלא". אבל הפרמטר הוא תאריך

קצר, לכן לפי העיקרון של העמסת שיטות, הקומפילר מבין שמבין 2 השיטות יש לבחור בפעולה שבמקור ב `Date`

השורה השלישית משווה בין שני תאריכים מלאים לא שווים לכן ברור שהתשובה תהיה "לא שווים"

השורה הרביעית והחמישית, אנחנו מנסים "לעבוד" על הקומפילר ומבקשים להמיר את אחד התאריכים ל-`Date` אנחנו מצליחים כיוון שכמו שראינו בדוגמה הראשונה והשנייה – בשורה רביעית עשינו המרה מפורשת וביקשנו שבזמן הקומפילציה, `fd2` ייראה כאילו הוא מטיפוס `Date`. זה גרם לבחור את הפעולה ממחלקת הבסיס ולמרות שבעצם יש כאן שני תאריכים מלאים-ההשוואה הייתה בין התכונות של תאריך קצר בשורה החמישית "הכרזנו" שהפרמטר הוא תאריך קצר ולכן לא יכלה להתבצע פעולה של תאריך מלא

דבר מעניין קרה בשורה

```
Console.WriteLine("_fd2 " + (Date)fd2);
```

לפי הדיון שפיתחנו, הדבר הקל ביותר, אם רוצים להציג לתאריך מלא רק את החלק הקצר שלו, זה פשוט להמיר אותו באופן מפורש

אבל המציאות שונה !!! הוא נשאר בשלו

הסיבה – אחרי שהעצם מומר לתאריך קצר, אנחנו עוברים לשלב הריצה. כאן מסתכלים איזה שיטה עומדת להתבצע.

יש לבצע `ToString`. בשלב זה, בזמן הריצה, שואלים אם לפעולה זו יש פעולה שדורסת אותה ("ירידה במורד

האובייקט") וכיוון שהתשובה חיובית, מתבצעת הפעולה `ToString` שבמחלקת `FullDate`. זהו מנגנון הפולימורפיזם המבוסס על דריסת שיטות.

אז למה זה לא קרה מקודם? – כי מקודם לא דיברנו על דריסה אלא על העמסת שיטות!!!

→ המשך

משימה שמינית

בנה את תכנית הבדיקה הבאה. שים לב שאחרי כל תת נושא יש פקודת `Console.ReadLine()` שעוצרת את ההרצה. ביכולתך לעקוב אחרי כל תת נושא. לפני ההרצה נסה להבין מה מנסים להראות בכל קטע ונסה לחזות מה יהיה הפלט?

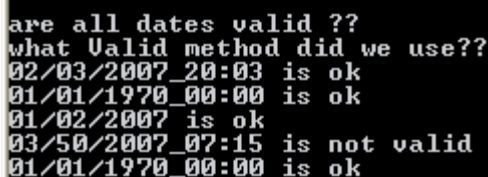
```
static void Main1(string[] args)
{
    Date[] dArr = new Date[5];
    dArr[0] = new FullDate(2, 3, 2007,20,3);
    dArr[1] = new FullDate();
    dArr[2] = new Date(1,2,2007);
    dArr[3] = new FullDate(3, 50, dArr[0].GetYear(),7,15);
    dArr[4] = dArr[1];
    Console.WriteLine("displaying the dates.what ToString method did we use??");
    for (int i = 0; i < dArr.Length; i++)
    {
        Console.WriteLine(dArr[i]);
    }
    Console.ReadLine();
    Console.WriteLine();
}
```



המערך הוא אמנם של תאריכים קצרים, גם אם יש בו תאריכים מלאים (פולימורפיזם, המרה כלפי מעלה, בלתי מפורשת), אבל בזמן הריצה מתבצעת דריסה בכל מקום אפשרי, לכן אוביקט שהוא במקורו תאריך קצר יודפס ככזה, ואילו תאריך מלא, גם אם הוא יושב במערך כאילו הוּא קצר, ברגע האמת, בזמן ריצה, מתבצעת דריסה ומופעלת מתודת `ToString` שכתובה במחלקה הנגזרת

```
//recognizing the type of each object
Console.WriteLine("are all dates valid ??");
Console.WriteLine("what Valid method did we use??");

for (int i = 0; i < dArr.Length; i++)
{
    if(dArr[i].IsValid() )
        Console.WriteLine(dArr[i]+" is ok");
    else
        Console.WriteLine(dArr[i]+" is not valid");
}
Console.ReadLine();
```



ההסבר למה שקרה כאן זהה לגמרי להסבר הקודם כי גם כאן התבצעה דריסה איפה שצריך

חומרי עזר שהוכנו ע"י משתתפי קורס מורים מובילים תשע"א
ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

```
Console.WriteLine("\n\nrecognizing the type of each object - version 1.");
```

```
for (int i = 0; i < dArr.Length; i++)
{
    if(dArr[i] is FullDate )
        Console.WriteLine(dArr[i]+" is FullDate");
    else
        Console.WriteLine(dArr[i]+" is Date");
}
Console.ReadLine();
```

```
recognizing the type of each object - version 1.
02/03/2007_20:03 is FullDate
01/01/1970_00:00 is FullDate
01/02/2007 is Date
03/50/2007_07:15 is FullDate
01/01/1970_00:00 is FullDate
```

לכאורה הכל ברור. אנחנו משתמשים באופרטור `is` שמתפקידו להגיד לנו באיזה טיפוס מדובר. אנחנו שואלים אותו אם מדובר ב `FullDate`. אם כן מדפיסים ואם לא מדפיסים שזה תאריך קצר. מצויין!!

```
Console.WriteLine();
Console.WriteLine("recognizing the type of each object - version 2.it seems the same.is it?");
for (int i = 0; i < dArr.Length; i++)
{
    if(dArr[i] is Date )
        Console.WriteLine(dArr[i]+" is Date");
    else
        Console.WriteLine(dArr[i]+" is FullDate");
}
Console.ReadLine();
```

לפי אותו הגיון, אפשר לכתוב אותו דבר ולשאל אם זה `Date` ולקבל בדיוק את אותו תשובות.
האמנם?

```
recognizing the type of each object - version 2.
02/03/2007_20:03 is Date
01/01/1970_00:00 is Date
01/02/2007 is Date
03/50/2007_07:15 is Date
01/01/1970_00:00 is Date
```

הפתעה-לא קיבלנו מה שרצינו. מכיוון שכל העצמים מסוג `Date` (מה שנכון נכון) קיבלנו מה שקיבלנו

נחזור, מה בעצם קרה כאן?! בדיקה זו אינה מבחינה בין עצמים מטיפוס `Date` לעצמים מטיפוס `FullDate` כיוון שמתקיים יחס ירושה בין שתי המחלקות כך ש: `Date IS FullDate`.
היחס אינו פועל בכיוון ה"הפוך" כלומר `Date IS NOT FullDate`.

מכאן, שעבור כל העצמים יודפס `is Date`.
מסקנה, האופרטור `IS` אינו שימושי עבור מחלקות שבראש ההיררכיה של עץ הירושה.

→ המשך

חומרי עזר שהוכנו ע"י משתתפי קורס מורים מובילים תשע"א
ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

//showing only short dates

```
Console.WriteLine("\n\nwe want to show only short date(no hour and minute");
Console.WriteLine("\nversion 1 - 'change all objects to Date' what happens?");
for (int i = 0; i < dArr.Length; i++)
{
    Console.WriteLine((Date)dArr[i] );
}
Console.ReadLine();
```

```
version 1 - 'change all objects to Date' what happens?
02/03/2007_20:03
01/01/1970_00:00
01/02/2007
03/50/2007_07:15
01/01/1970_00:00
```

כמובן שלא נקבל מה שרצינו, כיוון שכבר המרנו בצורה לא מפורשת, ועכשיו רק כתבנו את בצורה מפורשת, הדריסה כמובן שהתבצעה (בזמן ריצה)

```
Console.WriteLine("\nversion 2 - use 'is' and use special methos built in fullDate .is it right?why?");
```

```
for (int i = 0; i < dArr.Length; i++)
{
    if (dArr[i] is Date)
        Console.WriteLine(dArr[i] + " (originally Date)");
    else
    {
        Console.WriteLine( ((FullDate)dArr[i]).ToStringDateOnly() +
" (originally FullDate)");
    }
}
```

כאן בנינו פעולה מיוחדת בתוך מחלקת התאריך המלא. הסיבה שזה לא יכול לעבוד היא שכמו בדוגמה קודמת, אנחנו שואלים האם העצם הוא Date ומיד מקבלים תשובה חיובית לגבי כולם. לא עובד!

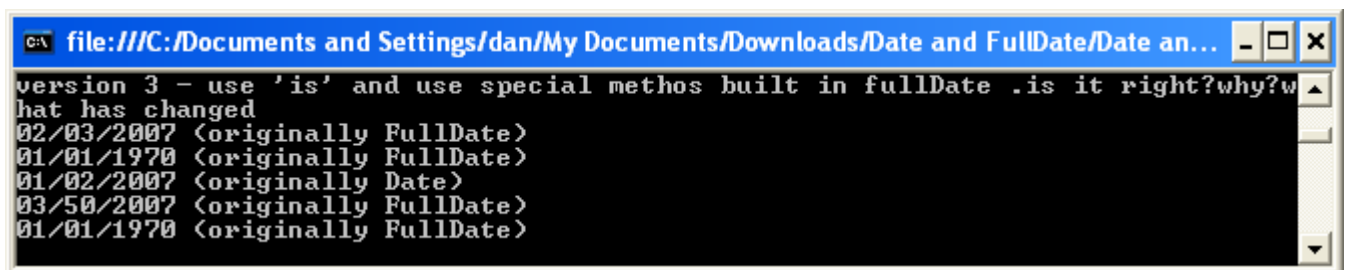
```
Console.ReadLine();
```

→המשך

חומרי עזר שהוכנו ע"י משתתפי קורס מורים מובילים תשע"א
ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

```
Console.WriteLine("\nversion 3 - use 'is' and use special methos built in fullDate .is it  
right?why?what has changed");
```

```
for (int i = 0; i < dArr.Length; i++)  
{  
    if (dArr[i] is FullDate)  
        Console.WriteLine( ((FullDate)dArr[i]).ToStringDateOnly() +  
" (originally FullDate)");  
    else  
    {  
        Console.WriteLine(dArr[i] + " (originally Date)");  
    }  
}
```



```
file:///C:/Documents and Settings/dan/My Documents/Downloads/Date and FullDate/Date an...  
version 3 - use 'is' and use special methos built in fullDate .is it right?why?w  
hat has changed  
02/03/2007 <originally FullDate>  
01/01/1970 <originally FullDate>  
01/02/2007 <originally Date>  
03/50/2007 <originally FullDate>  
01/01/1970 <originally FullDate>
```

סוף סוף!! כאן עשינו הפרדה נכונה בין העצמים. רק עצם אחד לא היה Fulldate. היתר שכן היו, החזירו תוצאה לפי
פעולה שבנינו במיוחד כדי לענות על הבעיה. כמובן שאין מה לדבר כאן על דריסה כי השם
שונה לגמרי מפעולות אחרות

```
Console.ReadLine();  
}
```