

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 ג'אווה

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 וייבא אליו את מחלקת Date
צור 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, נעדיף להשתמש בתכונות 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.

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.	<code>public Date (int day, int month, int year, int hour, int minute)</code>
יוצרת עצם מטיפוס Date המתחל לתאריך 1 בינואר 1970, חצות	<code>public Date ()</code>
יוצרת עצם מטיפוס Date על פי עצם מטיפוס Date המועבר כפרמטר	<code>public Date (FullDate other)</code>
שיטות נוספות	
(חוץ מ- <code>getMonth()</code> , <code>setMonth(int month)</code> 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	<code>public boolean isValid()</code> 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.	<code>public boolean isPrev(FullDate other)</code> 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.	<code>public boolean isEqual(Date other)</code> כנ"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.	<code>public String toString()</code> 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 extends Date
{
    protected int hour;
    protected int minute;

    public FullDate(int d, int mo, int y, int h, int m)
    {
        super(d, mo, y);
        this.hour = h;
        this.minute = m;
    }
    public FullDate(FullDate other)
    {
        super(other.getDay(), other.getMonth(), other.getYear());
        this.hour = other.hour;
        this.minute = other.minute;
    }
    public FullDate()
    {
        super();
        this.hour = 0;
        this.minute = 0;
    }
    public boolean isValid() //since the name is the same and parameters
                             //are the same(none),this is overriding
    {
        if (super.isValid() == false)
            return false;
        if (this.hour < 0 || this.hour > 23 || this.minute < 0 || this.minute > 59)
            return true;
        return true;
    }
    public boolean isPrev(FullDate other) //method overloading,it is not overriding
    {
        if (super.isPrev(other) == true)
            return true;
        if (this.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
    }
}
```

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

```
public boolean isEqual(FullDate other)//method overloading
{
    if(!this.isEqual((Date)other) )
        return false;
    if (this.hour == other.hour && this.minute == other.minute)
        return true;
    return false;
}

public String toString()
{
    return super.toString() + " _ " + hosafat0(this.hour) + ":" + hosafat0(this.minute);
}

//we will add this method later
public String toStringDateOnly()
{
    return super.toString();
}
}
```

משימה שביעית

בנה את תכנית הבדיקה הבאה

```
public static void main(String[] args)
{
    //-----

    FullDate fd2 = new FullDate(2, 5, 1989, 11, 5);
    System.out.println("fd2 is 02/05/1989 11:05");

    Date d4 = new Date(2, 5, 1989);
    System.out.println("fd4 is 02/05/1989 no hours and minutes");

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

    /*1*/ System.out.println("d4 = fd2 ?" + d4.isEqual(fd2));
    /*2*/ System.out.println("d4 = fd2 ?" + fd2.isEqual(d4));

    /*3*/ System.out.println("d5 = fd2 ?ofcourse no " + fd2.isEqual(fd5));
    /*4*/ System.out.println("d5 = fd2 ?      " + ((Date)fd2).isEqual(fd5));
    /*5*/ System.out.println("d5 = fd2 ?      " + (fd2).isEqual((Date)fd5));

    System.out.println("_fd2 " + (Date)fd2);

    System.out.println("_fd2 " + fd2.toStringDateOnly());

    System.out.println ();
}
```

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

זה הפלט

```
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??

בשורה הראשונה `d4.isEqual(fd2)` הפעולה היא של `d4`, עצם שהוקצה כ-`Date` ולכן הפעולה היא של תאריך "קצר", למרות העובדה שהפרמטר `fd2` הוא תאריך מלא.

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

בשורה השנייה הפעולה היא של d2 ולכן הפעולה היא לכאורה של תאריך "מלא". אבל הפרמטר הוא תאריך קצר, לכן לפי העיקרון של העמסת שיטות, הקומפיילר 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, מבצעים המרה כלפי מעלה של העצם fd2 כך שהמתודה equals אשר ב- Date מופעלת. באופן זה מושווה רק התאריך ה"קצר" והפעולה פולטת true.
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 האובריידינג") 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?

```
public static void main(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];
    System.out.println("displaying the dates.what ToString method did we use??");
    for (int i = 0; i < dArr.length; i++)
    {
        System.out.println(dArr[i]);
    }
    System.out.println();
}
```

```
displaying the dates.what ToString method did we
use??
02/03/2007_20:03
01/01/1970_00:00
01/02/2007
03/50/2007_07:15
01/01/1970_00:00
```

המערך `dArr` מוגדר מטיפוס "תאריך קצר" – `Date`.
כשמתבצעת השמה של תאריך ארוך `FullDate` במערך `dArr`, מתרחשת המרה (לא מפורשת) כלפי מעלה.
המתודה `toString()` הוגדרה מחדש (נדרסה) במחלקה `Date` לאחר שהתקבלה בירושה מהמחלקה `Object`.
באופן דומה המתודה הוגדרה מחדש (נדרסה) גם במחלקה היורשת `FullDate`.
בלולאה בה מודפסים פרטי "אברי המערך" (אובייקטים מעץ הירושה `Date`), מופעל מנגנון הפולימורפיזם אשר מזהה את הטיפוס ממנו הוקצה האובייקט בתחילה. בהתאם לטיפוס, מופעלת מתודה `toString` המתאימה.

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

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

for (int i = 0; i < dArr.length; i++)
{
    if(dArr[i].isValid() )
        System.out.println(dArr[i]+" is ok");
    else
        System.out.println(dArr[i]+" is not valid");
}
System.out.println();
```

```
are all dates valid ??
what Valid method did we use??
02/03/2007_20:03 is ok
01/01/1970_00:00 is ok
01/02/2007 is ok
03/50/2007_07:15 is not valid
01/01/1970_00:00 is ok
```

דוגמא נוספת להדגמת מנגנון הפולימורפיזם.

```
System.out.println();
System.out.println("\n\nrecognizing the type of each object - version 1.");

for (int i = 0; i < dArr.length; i++)
{
    if(dArr[i] instanceof FullDate )
        System.out.println(dArr[i]+" is FullDate");
    else
        System.out.println(dArr[i]+" is Date");
}
System.out.println();
```

```
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
```

שימוש באופרטור instanceof (זיהוי המחלקה ממנה הוקצה העצם).
קטע הקוד בודק האם העצם במקום i במערך dArr הוקצה כ- FullDate .


```
System.out.println();
System.out.println("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] instanceof Date )
        System.out.println(dArr[i]+" is Date");
    else
        System.out.println(dArr[i]+" is FullDate");
}
System.out.println();
```

```
recognizing the type of each object - version 2.it seems
the same.is it?
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
```

שימוש באופרטור instanceof (זיהוי המחלקה ממנה הוקצה העצם).
הפעם, קטע הקוד בודק האם העצם במקום i במערך dArr הוקצה כ- Date .

בדיקה זו אינה מבחינה בין עצמים מטיפוס Date לעצמים מטיפוס FullDate כיוון שמתקיים יחס ירושה בין שתי המחלקות כך ש: FullDate IS Date .
היחס אינו פועל בכיוון ה"הפוך" כלומר FullDate IS NOT Date .

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

//showing only short dates

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

```
we want to show only short date(no hour and minute

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
```

הדגמה נוספת של מנגנון הפולימורפיזם.
למרות הנסיון להמרה כלפי מעלה, פעולה ה- `toString` שתבצע תלויה בטיפוס ממנו הוקצה האובייקט מלכתחילה.

```
System.out.println("\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] instanceof Date)  
    System.out.println(dArr[i] + " (originally Date)");  
else  
{  
    System.out.println( ((FullDate)dArr[i]).toStringDateOnly() + " (originally FullDate)");  
}  
}
```

```
version 2 - use 'is' and use special methos built in fullDate .is it right?why?  
02/03/2007_20:03 (originally Date)  
01/01/1970_00:00 (originally Date)  
01/02/2007 (originally Date)  
03/50/2007_07:15 (originally Date)  
01/01/1970_00:00 (originally Date)
```

מטרת הקטע היא להדגים הפעלת מתודות "יחודיות" למחלקה (בדוגמא זו: `toStringDateOnly()`).
שוב, הקטע מדגים בחירה שגויה של שימוש באופרטור `instanceof` והמחלקה `Date`.

```
System.out.println();  
System.out.println("\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] instanceof FullDate)  
        System.out.println( ((FullDate)dArr[i]).toStringDateOnly() + " (originally FullDate)");  
else  
{  
    System.out.println(dArr[i] + " (originally Date)");  
}  
}
```

```
version 3 - use 'is' and use special methos built in  
fullDate .is it right?why?  
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)
```

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

בוצעה הבחנה בין העצמים.

המתודה ה"ייחודית" ToStringDateOnly הופעלה על העצמים שזוהו כ- FullDate, באמצעות כלפי מטה (המרה מפורשת).

כדאי לדעת: מנגנון הפולימורפיזם מאפשר גמישות רבה יותר בפיתוח, לעומת שימוש ב- instanceof. שימוש במנגנון זה מאפשר כתיבת תכנה המשתמשת בעץ הירושה. ניתן להרחיב את עץ הירושה (מחלקות נוספות) מבלי לשנות את התכניות שנכתבו (המשתמשות בעץ הירושה).
שימוש ב- instanceof מאלץ את כותב התכנה לשנות את הקוד הקיים כאשר מתרחב עץ הירושה.

```
System.out.println();  
}
```