

מחלקה Date

עליכם לכתוב מחלקה בשם Date המייצגת תאריך.
למחלקה 3 תכונות, `_day, _month, _year` מסוג מספר שלם.
כתבו בנאי למחלקה המקבל 3 מספרים שלמים המהווים את התאריך.
כמוכן כתבו בנאי העתקה המקבל תאריך ומעתיק את תכונות התאריך
לתאריך חדש.

כתבו את השיטות הבאות:

`getDay, getMonth, getYear, setDay, setMonth, setYear`,
`equals, toString` המקבלת תאריך נוסף ובודקת אם התאריכים
שווים.

כתבו שיטה בשם `before` המקבלת כארגומנט תאריך נוסף ומחזירה
`true` אם התאריך אליה הפעלנו את השיטה באה לפני התאריך
שהתקבל כפרמטר.

כתבו שיטה בשם `after` המקבלת כארגומנט תאריך נוסף ומחזירה
`true` אם התאריך אליה הפעלנו את השיטה באה אחרי התאריך
שהתקבל כפרמטר. יש לכתוב את השיטה בעזרת שימוש בשיטה
`before` בלבד.

כתבו שיטה בשם `lastCentury` שמחזירה אמת אם התאריך הוא
מהמאה הקודמת.
כתבו API למחלקה.

הוסיפו לפרוייקט מחלקה חדשה בשם `DateTester` שתכיל רק שיטת
`main`. קוד ההרצה שתכתבו בשביל לבדוק את המחלקה `Date` יכתב
בשיטה זו.

הגדירו 2 תאריכים `date1, date2` עם תאריכים 3/5/1998 ו-
4/2/2008.

הגדירו תאריך שלישי `date3` בעל אותו תאריך של התאריך השני.

שנו את התאריך הראשון ל- 3/7/1998.

הדפיסו את השנה של התאריך הראשון.

השוו בין התאריך הראשון לשני והדפיסו הודעה מתאימה אם
התאריכים שווים או לא.

השוו בין התאריך השני לשלישי והדפיסו הודעה מתאימה אם
התאריכים שווים או לא.
בדקו אם התאריך הראשון בא לפני התאריך השני.
בדקו אם התאריך הראשון בא אחרי התאריך השני.
בדקו אם התאריך הראשון הוא מהמאה הקודמת.
הדפיסו את שלושת התאריכים.

ניסו לבצע בדיקות של המחלקה בעזרת ספסל האובייקטים של
blue j.

```

/**
 * This class represents a Date Object
 *
 * @author (Judy Isaacs)
 * @version (2011a)
 */

public class Date {
    private int _day;
    private int _month;
    private int _year;

    //constructors:
    /**
     * creates a new Date object
     * @param day the day in the month(1-31)
     * @param month the month in the year(1-12)
     * @param year the year ( 4 digits)
     */

    public Date(int day, int month, int year) {
        _day = day;
        _month = month;
        _year = year;
    }

    /**
     * Copy Constructor
     * @param date to be copied
     */
    public Date(Date other){

        _day = other._day;

```

```

        _month = other._month;
        _year = other._year;
    }

    /** gets the year
     * @return the year
     */
    public int getYear(){
        return _year;
    }

    /** gets the month
     * @return the month
     */

    public int getMonth(){
        return _month;
    }

    /** gets the Day
     * @return the day
     */
    public int getDay(){
        return _day;
    }

    /** sets the year
     * @param yearToSet the value to be set
     */
    public void setYear(int yearToSet){
        _year = yearToSet;
    }

```

```

/** set the month
 * @param monthToSet the value to be set
 */
public void setMonth(int monthToSet){
    _month = monthToSet;
}

/** sets the day
 * @param dayToSet the value to be set
 */
public void setDay(int dayToSet){
    _day = dayToSet;
}

/** check if 2 dates are the same
 * @param other the date to compare this date to
 * @return true if the dates are the same
 */
public boolean equals(Date other){

if(_day==other._day&&_month==other._month&&_year==other._year)
    return true;
    return false;
}
/** check if the date is from the previous century
 *
 * @return true if the date is from the previous
century
 */
public boolean lastCentury(){
    if (_year<2000)

```

```

        return true;
        return false;
    }

/**
 * check if this date is before other date
 * @param other the date to compare this date to
 * @return true if this date is before other date
 */
public boolean before(Date other){

    if (_year<other._year || (_year==other._year &&
    _month<other._month) || (_year==other._year &&
    _month==other._month && _day<other._day))
        return true;
    return false;
}

/**
 * check if this date is after other date
 * @param other the date to compare this date to
 * @return true if this date is after other date
 */

public boolean after(Date other){

    return other.before(this);
}

/**
 * returns a String that represents this date
 *
 * @return String that represents this date
 * in the following format:

```

```
* day/month/year (30/9/1917)
*/
public String toString() {
    return _day + "/" + _month + "/" + _year;
}
}
```

```

public class DateTester
{
    public static void main (String[]args)
    {

        Date d1=new Date(3,5,1998);
        Date d2=new Date(4,2,2007);
        Date d3=new Date(d2);
        d1.setMonth(7);
        System.out.println( d1.getYear());
        if( d1.equals(d2))
            System.out.println("dates 1 and 2 are equal") ;
        else
            System.out.println("dates 1 and 2 are not equal") ;
        if( d2.equals(d3))
            System.out.println("dates 2 and 3 are equal") ;
        else
            System.out.println("dates 2 and 3 are not equal") ;
        if(d1.before(d2))
            System.out.println("date 1 comes before date 2");
        if(d1.after(d2))
            System.out.println("date 1 comes after date 2");
        if(d1.lastCentury())
            System.out.println( "First date from previous century");
        else
            System.out.println( "First date from this century");
        System.out.println( d1);
        System.out.println( d2);
        System.out.println( d3);
    }
}

```