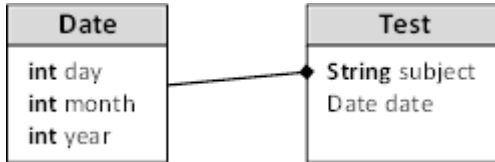


לוח מבחנים

מבוסס על דף עבודה ממקור לא ידוע

1. המחלקה מבחן Test מגדירה מבחן לפי תרשים uml הבא :



להלן ממשק המחלקה Test

כותרת הפעולה	תיאור הפעולה
Test (string subject, Date date)	פעולה בונה היוצרת מבחן במקצוע subject ובתאריך date
string GetSubject ()	פעולה המחזירה מקצוע המבחן
Date GetDate ()	פעולה המחזירה את תאריך המבחן
void SetSubject (string subject)	פעולה המשנה את מקצוע המבחן ל- subject
void SetDate (Date date)	פעולה המשנה את תאריך המבחן ל- date
bool SameDate (Test t)	פעולה המחזירה אמת אם המבחן הנוכחי מתקיים באותו יום כמו המבחן שהתקבל כפרמטר ושקר אחרת
bool Equals (Test t)	פעולה המחזירה אמת אם המבחן הנוכחי זהה למבחן שהתקבל כפרמטר ושקר אחרת
string ToString ()	פעולה המחזירה מחרוזת המתארת את מצב המבחן

כתבו את המחלקות Date ו- Test.

2. המחלקה לוח מבחנים TestTable מגדירה אוסף של מבחנים.

המבחנים שמורים במערך ויש בו לכל היותר 20 מבחנים.

א. כתוב את כותרת המחלקה ואת תכונותיה.

תכונות המחלקה: מערך של מבחנים, קבוע המציין את גודל המערך, lastTest המציין מהו מספר

המבחנים שקיימים במערך בפועל.

להלן ממשק המחלקה TestTable:

כותרת הפעולה	תיאור הפעולה
TestTable ()	פעולה בונה היוצרת לוח מבחנים (ריק).
void Add (Test t)	פעולה המוסיפה מבחן t ללוח המבחנים. הנחה: יש מספיק מקום בלוח המבחנים ואין מבחן באותו תאריך.
bool IsEmpty ()	פעולה המחזירה אמת אם לוח המבחנים ריק ושקר אחרת

Test GetTest (string subject)	פעולה המחזירה הפנייה למבחן הראשון בלוח המבחנים במקצוע subject
Test GetTest (Date date)	פעולה המחזירה הפנייה למבחן הראשון בלוח המבחנים בתאריך date
Test Remove (string subject)	פעולה המוחקת את המבחן שראשון בלוח המבחנים, במקצוע subject. הפעולה מחזירה הפנייה למבחן שנמחק מהלוח.
Test Remove (Date date)	פעולה המוחקת את המבחן שראשון בלוח המבחנים, בתאריך date. הפעולה מחזירה הפנייה למבחן שנמחק מהלוח.
string ToString ()	פעולה המחזירה מחרוזת המתארת את מצב לוח המבחנים

3. כתוב תכנית היוצרת לוח מבחנים.

- הוסף ללוח 5-7 מבחנים.
- הצג את לוח המבחנים.
- הוחלט להחליף בין התאריכים של המבחנים במתמטיקה ובספרות. החלף את המבחנים בלוח והצג את הלוח החדש.
- קלוט תאריך ובדוק איזה מבחן מתקיים בתאריך זה. (הצג את המבחן - מקצוע + תאריך).

4. הוסף למחלקה **לוח מבחנים** פעולה בשם: הוסף מבחן בצורה ממויינת InsertSorted(Test t) המקבלת מבחן t ומוסיפה אותו ללוח המבחנים הממוין בצורה ממויינת לפי תאריך הבחינה.

הערות:

- שים לב שמעבר לפעולות הממשק המובאות, יתכן ותצטרך להוסיף למחלקות השונות פעולות נוספות, כמו למשל פעולה בוליאנית: Equals(date) המשווה שני תאריכים או Equals(test) לשני מבחנים.
- להוספת מבחן בצורה ממויינת נוסף את הפעולה הבוליאנית Before(date) למחלקה תאריך, שתחזיר אמת אם התאריך הנוכחי מוקדם לתאריך שהתקבל ושקר אחרת.

עבודה נעימה

המחלקה Date

```
/**
 * Date - המחלקה תאריך
 */
public class Date
{
    //--- תכונות המחלקה ---
    private int day;    // יום
    private int month; // חודש
    private int year;  // שנה

    //--- תכונה של המחלקה ---
    public static int count = 0; // מונה מספר העצמים מסוג תאריך

    //--- פעולות בונות ---
    // --- בנאי מחדל ---
    public Date()
    {
        this.day = 1;
        this.month = 1;
        this.year = 1980;
        count++;
    }

    // --- בנאי המקבל יום חודש ושנה ---
    public Date(int day, int month, int year)
    {
        this.day = day;
        this.month = month;
        this.year = year;
        Date.count++;
    }

    // --- פעולה בונה מעתיקה / בנאי מעתיק ---
    public Date(Date d)
    {
        this.day = d.day;
        this.month = d.month;
        this.year = d.year;
        Date.count++;
    }

    // --- פעולות מאחזרות לכל תכונה ---
    // --- פעולה המחזירה את היום הנוכחי ---
    public int GetDay()
    {
        return this.day;
    }

    //--- פעולה המחזירה את החודש הנוכחי ---
    public int GetMonth()
    {
        return this.month;
    }
}
```

```
//--- פעולה המחזירה את השנה הנוכחית ---
public int GetYear()
{
    return this.year;
}

//--- פעולת עדכון לכל תכונה ---
//--- day -ל- היום הנוכחי ---
public void SetDay(int day)
{
    this.day = day;
}

//--- month -ל- החודש הנוכחי ---
public void SetMonth(int month)
{
    this.month = month;
}

//--- year -ל- השנה הנוכחית ---
public void SetYear(int year)
{
    this.year = year;
}

// --- את פעולה המחזירה את מצב התאריך הנוכחי כמחרוזת ---
public string ToString()
{
    return this.day + "-" + this.month + "-" + this.year;
}

//--- פעולה המחזירה אמת אם התאריכים זהים ושקר אחרת ---
public bool Equals (Date d)
{
    return this.day == d.day &&
           this.month == d.month &&
           this.year == d.year;
}

//---פעולה המחזירה אמת אם התאריך הנוכחי ---
//--- בא לפני התאריך שהתקבל ושקר אחרת ---
public bool Before (Date d)
{
    if (this.year < d.year) return true;
    if (this.year > d.year) return false;

    if (this.month < d.month) return true;
    if (this.month > d.month) return false;

    if (this.day < d.day) return true;
    return false;
}
}
```

המחלקה Test

```

/**
 *   המחלקה מבחן Test
 */
public class Test
{
    //--- תכונות המחלקה ---
    private string subject;    // מקצוע הבחינה
    private Date date;        // תאריך הבחינה

    //--- פעולה בונה ---
    public Test(string subject, Date date)
    {
        this.subject = subject;
        this.date = new Date(date);    // Date של המעתיק של
    }

    public string GetSubject()
    {
        return this.subject;
    }

    public Date GetDate()
    {
        return this.date;
    }

    public void SetDate(Date date)
    {
        this.date = date;
    }

    public void SetSubject(string subject)
    {
        this.subject = subject;
    }

    //--- האם שני המבחנים מתקיימים באותו יום ? ---
    public bool sameDate(Test t)
    {
        return this.date.Equals(t.date);
    }

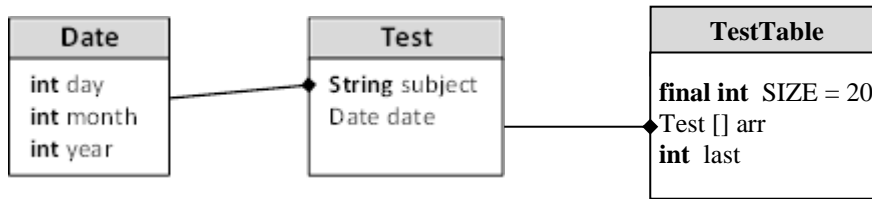
    //--- פעולה המחזירה את מצב המבחן הנוכחי ---
    public string ToString()
    {
        string str = "subject: " + this.subject;
        str += "\t date: " + this.date.ToString();
        return str;
    }
}

```

פעולות מאחזרות
לכל תכונה

פעולות מעדכנות
לכל תכונה

לוח המבחנים באיחוס אצרה



המחלקות בפרויקט:

```

/*
 *      TestTable - לוח מבחנים
 */
public class TestTable
{
    private Test [] arr;           // מערך המבחנים
    public const int SIZE = 20;    // קבוע המגדיר את מספר המבחנים המרבי
    private int last;             // מספר המבחנים הקיימים בפועל

    //--- פעולה בונה היוצרת לוח מבחנים ריק ---
    public TestTable()
    {
        this.arr = new Test[SIZE];
        this.last = 0;
    }

    //--- הוספת מבחן ללוח המבחנים ---
    //--- הנחה: יש מספיק מקום בלוח המבחנים ---
    public void Add (Test t)
    {
        this.arr[this.last] = t;
        this.last ++;
    }

    //--- הוספת מבחן ללוח המבחנים ממוין לפי תאריך הבחינה ---
    public void InsertSorted (Test t)
    {
        //--- שלב 1 - מציאת המקום המתאים ---
        int i = 0;
        while (i < this.last && this.arr[i].GetDate().Before(t.GetDate()))
            i++;

        //--- שלב 2 - הזזת המבחנים ופינוי מקום למבחן החדש ---
        for (int j = this.last ; j > i ; j--)
            this.arr[j] = this.arr[j-1];

        //--- שלב 3 - הכנסת המבחן למקום שהתפנה והגדלת last ---
        this.arr[i] = t;
        this.last ++;
    }

    //--- פעולה המחזירה אמת אם לוח המבחנים ריק ושקר אחרת ---
    public bool IsEmpty()
    {
        return this.last == 0;
    }
}
  
```

```
//--- פעולה המחזירה הפנייה למבחן לפי שם המבחן ---
public Test GetTest (string subject)
{
    int i = 0;
    while (i < this.last && ! this.arr[i].GetSubject().Equals(subject))
        i++;
    return this.arr[i];    // אם המבחן לא נמצא יוחזר null
}

//--- פעולה המחזירה הפנייה למבחן לפי תאריך המבחן ---
public Test GetTest (Date d)
{
    int i = 0;
    while (i < this.last && ! this.arr[i].GetDate().Equals(d))
        i++;
    return this.arr[i];    // אם המבחן לא נמצא יוחזר null
}

//--- פעולה המוציאה ומחזירה את המבחן הראשון במקצוע הנתון ---
public Test Remove (string subject)
{
    //--- שלב 1 - איתור המבחן שיש להסיר ---
    int i = 0;
    while (i < this.last && ! this.arr[i].GetSubject().Equals(subject))
        i++;

    if (i == this.last)    // המבחן לא נמצא בלוח
        return null ;

    //--- שלב 2 - השמת הפנייה למבחן ---
    Test t = this.arr[i];

    //--- שלב 3 - הסרת המבחן מהלוח ---
    this.last -- ;    // last יפנה למבחן האחרון
    this.arr[i] = this.arr[this.last];    // העברת המבחן האחרון למקום שהתפנה
    this.arr[this.last] = null;    // השמת null בתא האחרון שהתפנה

    //--- שלב 4 - החזרת הפנייה למבחן שהוסר מלוח המבחנים ---
    return t;
}

//--- פעולה המוציאה ומחזירה את המבחן הראשון בתאריך הנתון ---
public Test Remove (Date d)
{
    int i = 0;
    while (i < this.last && ! this.arr[i].GetDate().Equals(d))
        i++;

    if (i == this.last)    // המבחן לא נמצא בלוח
        return null ;

    Test t = this.arr[i];    // השמת הפנייה למבחן

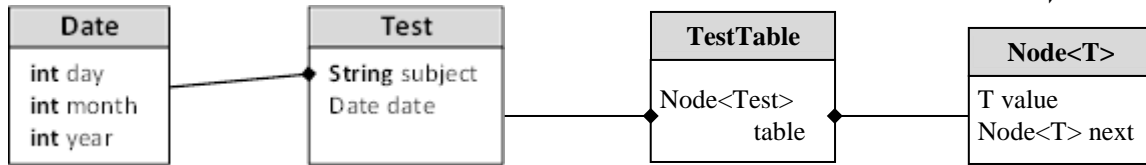
    this.last -- ;    // last האחרון למבחן יפנה
    this.arr[i] = this.arr[this.last];    // העברת המבחן האחרון למקום שהתפנה
    this.arr[this.last] = null;    // השמת null בתא האחרון שהתפנה

    return t;
}
```

```
--- פעולה המחזירה מחרוזת המתארת את מצב לוח המבצנים ---  
public string ToString()  
{  
    string str = "";  
    for (int i = 0 ; i < this.last ; i++)  
        str += this.arr[i].ToString() + "\n";  
    return str;  
}  
}
```


לוח המבחנים באימוש שרשרת חוליות

המחלקות בפרויקט:



```

/*
 * Node <T> הגנרית
 */
public class Node<T>
{
    private T value;
    private Node<T> next;

    public Node(T value)
    {
        this.value = value;
        this.next = null;
    }

    public Node(T value, Node<T> next)
    {
        this.value = value;
        this.next = next;
    }

    public T GetValue()
    {
        return value;
    }

    public Node<T> GetNext()
    {
        return next;
    }

    public bool HasNext ()
    {
        return this.next != null;
    }

    public void SetValue(T value)
    {
        this.value = value;
    }

    public void SetNext(Node<T> next)
    {
        this.next = next;
    }

    public string ToString()
    {
        return this.value.ToString();
    }
}
  
```

```
/*
 *      TestTable - לוח מבחנים
 */
public class TestTable
{
    private Node <Test> table; // הפנייה למבחן הראשון בלוח המבחנים

    //--- פעולה בונה היוצרת לוח מבחנים ריק ---
    public TestTable()
    {
        this.table = null;
    }

    //--- הוספת מבחן ללוח המבחנים ---
    public void Add (Test t)
    {
        this.table = new Node<Test>(t, this.table);
    }

    //--- הוספת מבחן ללוח המבחנים ממוין לפי תאריך הבחינה ---
    public void InsertSorted (Test t)
    {
        //--- מציאת המקום המתאים ושמירת המקום הקודם לו ב-prev ---
        Node<Test> pos = this.table;
        Node<Test> prev = null;

        while (pos != null && pos.GetValue().GetDate().Before(t.GetDate()))
        {
            prev = pos;
            pos = pos.GetNext() ;
        }

        //--- יצירת חוליה למבחן החדש ---
        Node<Test>temp = new Node<Test> (t, pos);

        //--- האם יש להכניס את המבחן החדש בהתחלה ---
        //--- (נכון גם למצב שבו לוח המבחנים ריק וזהו המבחן הראשון) ---
        if (prev == null)
            this.table = temp;
        else
            prev.SetNext(temp); //--- הכנסת המבחן באמצע או בסוף ---
    }

    //--- פעולה המחזירה אם לוח המבחנים ריק ושקר אחרת ---
    public bool IsEmpty()
    {
        return this.table == null;
    }

    //--- פעולה המחזירה הפנייה למבחן לפי שם המבחן ---
    public Test GetTest (string subject)
    {
        Node<Test> p = this.table;
        while (p != null && ! p.GetValue().GetSubject().Equals(subject))
            p = p.GetNext();
        if (p != null)
            return p.GetValue(); // אם המבחן לא נמצא יוחזר null
        return null;
    }
}
```

```
//--- פעולה המחזירה הפנייה למבוק לפי תאריך המבוק ---
public Test GetTest (Date d)
{
    Node<Test> p = this.table;
    while (p != null && ! p.GetValue().GetDate().Equals(d))
        p = p.GetNext();
    if (p != null)
        return p.GetValue(); // אם המבוק לא נמצא יוחזר null
    return null;
}

//--- פעולה המוציאה ומחזירה את המבוק הראשון במקצוע הנתון ---
public Test Remove (string subject)
{
    Node<Test> p = this.table;
    Node <Test> prev = null;
    while (p != null && ! p.GetValue().GetSubject().Equals(subject))
    {
        prev = p;
        p = p.GetNext();
    }

    //--- המבוק לא נמצא ---
    if (p == null)
        return null;

    //--- המבוק ראשון בשרשרת ---
    if (prev == null)
        this.table = p;
    else
        prev.SetNext(p); // עדכון הקודם למבוק

    p.SetNext(null); // ניתוק מהבוק מהשרשרת
    return p.GetValue();
}

//--- פעולה המוציאה ומחזירה את המבוק הראשון בתאריך הנתון ---
public Test Remove (Date d)
{
    Node<Test> p = this.table;
    Node <Test> prev = null;
    while (p != null && ! p.GetValue().GetDate().Equals(d))
    {
        prev = p;
        p = p.GetNext();
    }

    //--- המבוק לא נמצא ---
    if (p == null)
        return null;

    //--- המבוק ראשון בשרשרת ---
    if (prev == null)
        this.table = p;
    else
        prev.SetNext(p); // עדכון הקודם למבוק

    p.SetNext(null); // ניתוק מהבוק מהשרשרת
    return p.GetValue();
}
```

```
//--- פעולה המחזירה מחרוזת המתארת את מצב לוח המבצנים ---  
public string ToString()  
{  
    string str = "";  
    int i = 1;  
    Node<Test> p = this.table;  
    while (p != null)  
    {  
        str += "test" + i + ": " + p.GetValue().ToString() + "\n";  
        i++;  
        p = p.GetNext();  
    }  
    return str;  
}  
}
```

התכנית

שימו לב! התכנית אינה משתנה כשמשתנה האוסף ממערך לשרשרת חוליות

```
public class ChkTestTable
{
    public static void Main(string [] args)
    {
        TestTable tt1 = new TestTable(); // הלווח הלא ממויין
        TestTable tt2 = new TestTable(); // הלווח הממויין

        for (int i = 0 ; i < 7 ; i++)
        {
            Test t = GetTest();
            tt1.Add(t);
            tt2.InsertSorted(t);
        }

        Console.WriteLine("Test Table");
        Console.WriteLine ("~~~~~");
        Console.WriteLine (tt1.ToString());
        Console.WriteLine ();

        Console.WriteLine ("Sorted Test Table");
        Console.WriteLine ("~~~~~");
        Console.WriteLine (tt2.ToString());
        Console.WriteLine ();
    }

    //--- פעולה המחזירה מבחן ---
    static Random rnd = new Random();
    static string [] subArr = {"Math", "Litriture", "CS", "History",
                               "Physics", "English"};

    public static Test GetTest()
    {
        Date date = GetDate();
        int sb = rnd.Next (subArr.Length);

        return new Test (subArr[sb], date);
    }

    //--- תאריך המחזירה פעולה ---
    static int [] yearArr = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    public static Date GetDate()
    {
        int m = rnd.Next (12); // הגרלת חודש
        while (m > 4 && m < 8) // אם החודש הוא יוני, יולי או אוגוסט
            m = rnd.Next (12); // (שבו אין מבחנים), הגרל חודש חדש

        int d = rnd.Next (yearArr[m])+1; // הגרלת יום בחודש
        int y;
        if (m < 8) // if (m < 5) ואפשר גם
            y = 2016;
        else
            y = 2015;

        return new Date (d, m+1, y);
    }
}
```

פלט התכנית

Test Table

~~~~~

|                           |                  |
|---------------------------|------------------|
| test1: subject: Math      | date: 4-1-2016   |
| test2: subject: English   | date: 10-11-2015 |
| test3: subject: Physics   | date: 3-12-2015  |
| test4: subject: Litriture | date: 8-9-2015   |
| test5: subject: CS        | date: 17-12-2015 |
| test6: subject: English   | date: 22-2-2016  |
| test7: subject: CS        | date: 19-2-2016  |

Sorted Test Table

~~~~~

test1: subject: Litriture	date: 8-9-2015
test2: subject: English	date: 10-11-2015
test3: subject: Physics	date: 3-12-2015
test4: subject: CS	date: 17-12-2015
test5: subject: Math	date: 4-1-2016
test6: subject: CS	date: 19-2-2016
test7: subject: English	date: 22-2-2016