

## **חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"ה**

ניתן להשתמש בחומרים לצורך הוראה בלבד.

**לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי**

ללא קבלת אישור מראש מצוות הפיתוח

**כתיבה ועריכה:**

**אביטל גרינולד, דורית כהן, משה שטיינר**

## מחלקת שירות הכוללת אוסף פעולות שימושיות המטפלות ברשימה מקושרת של מספרים שלמים גירסת C#

מטרת המשימה: בניית מחלקת שירות המכילה אוסף פעולות שימושיות על רשימת חוליות מקושרת של מספרים שלמים.

צור פרויקט **MyList**.

בנה את המחלקות: **UtilListInt**

**Program**.

המחלקה **UtilListInt** תכיל את הפעולות המפורטות להלן, תוכנית הבדיקה במחלקה **Program** תזמן אותן ותציג פלטים ברורים.

**רשימה מקושרת** או בקיצור **רשימה** הוא מבנה דינמי הבנוי מאפס חוליות או יותר.

רשימה ריקה היא רשימה ללא חוליות, כלומר ייצוג שלה יהיה: `Node<T> list = null;`  
**חוליה**, `Node<T>`, הוא טיפוס המכיל ערך והפנייה לחוליה הבאה או `null`, המסמן את סוף הרשימה.

להלן ממשק המחלקה `Node<T>`:

יעילות הפעולה	כותרת הפעולה	תיאור הפעולה
O(1)	<code>public Node(T x)</code>	יוצרת חוליה בעלת ערך x והפנייה ל null
O(1)	<code>public Node(T x, Node&lt;T&gt; next)</code>	יוצרת חוליה בעלת ערך x והפנייה לחוליה next
O(1)	<code>public T GetValue()</code>	מחזירה את ערך החוליה
O(1)	<code>public Node&lt;T&gt; GetNext()</code>	מחזירה את ההפניה לחוליה העוקבת או null
O(1)	<code>public void SetValue(T x)</code>	משנה את ערך החוליה ל x
O(1)	<code>public void SetNext(Node&lt;T&gt; next)</code>	משנה את ההפניה לחוליה הבאה ל next
O(1)	<code>public bool HasNext()</code>	מחזירה 'אמת' אם יש חוליה עוקבת, 'שקר'-אחרת.
O(1)	<code>public String ToString()</code>	מחזירה מחרוזת המתארת את ערך החוליה

הנחיות והערות:

- (1) ביצירת המחלקות יש להשתמש בטיפוס `Node<T>`. לשם כך הוסף את ספריית `unit4.dll` לרשימת ה-`reference`. בנוסף, יש להוסיף התייחסות לספרייה בראש הקובץ:  
`using Unit4.CollectionsLib ;`  
 זיכרו למחוק את כל משפטי ה-`using` האחרים (למעט `using System;`)
- (2) הוסף בכל פעם מימוש של פעולה אחת ובדוק אותה. לאחר שבדקת שהיא עובדת - עבור לפעולה הבאה וכך הלאה.
- (3) ממש את הפעולות שמסומנות ב (\*) הן באופן איטרטיבי (בלולאה) והן באופן רקורסיבי. לפעולות במימוש רקורסיבי הוסף את האות R לשם הפעולה. למשל פעולה בשם `Sum` תקבל את השם `SumR` בצורתה הרקורסיבית.
- (4) לכל אחת מהפעולות, רשום את היעילות. נמק תשובתך.
- (5) באומרנו "רשימה", הכוונה להפניה לחוליה ראשונה ברשימה, או `null` אם היא ריקה.
- (6) הקפד לתת שמות שמרמזים על תפקיד ההפניה לעצם מטיפוס `Node<T>`.
  - אם משמעותו רשימה, קבע שם כגון: `lst list` וכדומה.
  - אם משמעותו חוליה בודדת, קבע שם כגון `node`.
  - אם משמעותו מקום ברשימה, קבע שם כגון: `position, pos, p` וכדומה.

<b>( א ) פעולות בנייה</b>		
<code>public static Node&lt;int&gt;Create1_N (int n)</code>	פעולה המקבלת מספר טבעי n ומחזירה רשימה של מספרים טבעיים בסדר עולה מ 1 עד n כולל.	<b>1</b>
<code>public static void Print(Node&lt;int&gt; list)</code>	פעולה המקבלת הפנייה לחוליה ראשונה ברשימה של שלמים ומדפיסה אותה בצורה $n2 \rightarrow \dots \rightarrow null \rightarrow n1$	<b>2</b>
<code>public static Node&lt;int&gt;BuildRandom (int n, int low, int high)</code>	פעולה היוצרת רשימה עם n מספרים אקראיים בין low ל-high כולל. n מספר טבעי. הפעולה מחזירה הפנייה לחוליה ראשונה ברשימה.	<b>3</b>
<code>public static Node&lt;int&gt;Create()</code>	פעולה הבונה רשימה עם מספרים שנקלטו מהמשתמש. סוף קלט 99. הפעולה מחזירה הפנייה לחוליה ראשונה ברשימה, null אם היא ריקה.	<b>4</b>

<b>( ב ) פעולות החזרת ערך מספרי</b>		
<code>public static int Count(Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את אורכה, כלומר את מספר האיברים ברשימה. (*)	<b>5</b>
<code>public static int Sum(Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את סכום ערכיה. 0 - אם הרשימה ריקה. (*)	<b>6</b>
<code>public static double Average (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את ממוצע ערכיה. 0 - אם הרשימה ריקה.	<b>7</b>
<code>public static int Product (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את מכפלת ערכיה. 0 - אם הרשימה ריקה. (*)	<b>8</b>
<code>public static int Max(Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את הערך המקסימלי. הנח שהרשימה אינה ריקה. (*)	<b>9</b>
<code>public static int Min(Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את הערך המינימלי. הנח שהרשימה אינה ריקה. (*)	<b>10</b>
<code>public static int SumEven (Node&lt;int&gt; list)</code>	פעולה רקורסיבית המקבלת רשימה של מספרים שלמים ומחזירה את סכום הערכים הזוגיים. (*)	<b>11</b>
<code>public static int SumOddPlaces (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה את סכום הערכים במקומות האי-זוגיים. החוליה הראשונה במקום 1.	<b>12</b>
<code>public static int CountX (Node&lt;int&gt; list, int X)</code>	פעולה המקבלת רשימה של מספרים שלמים ומספר שלם נוסף x, ומחזירה את מספר הפעמים שהמספר x מופיע ברשימה. (*)	<b>13</b>

<b>( ג ) פעולות המחזירות ערך בוליאני</b>		
<code>public static bool IsUp (Node&lt;int&gt; list)</code>	הפעולה מקבלת רשימה של מספרים שלמים ומחזירה 'אמת' אם היא ממוינת מסודרת בסדר עולה, 'שקר' – אחרת. עבור רשימה ריקה תחזיר 'אמת'. (*)	<b>14</b>
<code>public static bool IsExist (Node&lt;int&gt; list, int x)</code>	הפעולה מקבלת רשימה של מספרים שלמים ומספר שלם נוסף x, ומחזירה 'אמת' אם המספר x נמצא ברשימה, 'שקר' – אחרת. (*)	<b>15</b>
<code>public static bool IsArithmetic (Node&lt;int&gt; list)</code>	הפעולה מקבלת רשימה של מספרים שלמים ומחזירה 'אמת' אם היא מהווה סדרה חשבונית, 'שקר' – אחרת. עבור רשימה ריקה תחזיר 'אמת'. (*) <b>סדרה חשבונית</b> היא סדרת מספרים שההפרש בין כל שני ערכים סמוכים זהה.	<b>16</b>

<b>( ד ) פעולות המחזירות הפנייה למקום ברשימה</b>		
<code>public static Node&lt;int&gt; Goto (Node&lt;int&gt; list, int step)</code>	פעולה המקבלת רשימה של מספרים שלמים לא ריקה ומספר צעדים step ומחזירה את ההפניה לחוליה במקום step. אם step גדול ממספר החוליות, תחזיר null. (*)	<b>17</b>
<code>public static Node&lt;int&gt; GotoLast (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים לא ריקה ומחזירה את ההפניה לחוליה אחרונה, או null אם הרשימה ריקה. (*)	<b>18</b>
<code>public static Node&lt;int&gt; FirstPosX (Node&lt;int&gt; list, int x)</code>	פעולה המקבלת רשימה של מספרים שלמים ומספר שלם נוסף, x, ומחזירה הפניה למקום ההופעה הראשון של x מתחילת הרשימה, או null – אם לא נמצא. (*)	<b>19</b>
<code>public static Node&lt;int&gt; PosMin (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים שלמים ומחזירה הפנייה למיקום הערך הקטן ביותר ברשימה. הנחה: הרשימה אינה ריקה, וכל הערכים שונים זה מזה.	<b>20</b>

- הערות: (1) בכל הפעולות הבאות נחזיר הפנייה לחוליה ראשונה ברשימה חדשה אשר נוצרה מרשימה קיימת או null. באומרנו: מחזיר רשימה, הכוונה מחזיר הפניה לחוליה הראשונה ברשימה או null.
- (2) בכל הפעולות הבאות, הנח pos או prev הם מקומות ברשימה או null.

<b>(ה) שינוי רשימה קיימת, בפועל מחזיר רשימה חדשה</b>		
<code>public static Node&lt;int&gt; Insert (Node&lt;int&gt; list, Node&lt;int&gt; pos, int x)</code>	פעולה המקבלת רשימה, מקום ברשימה pos וערך x ומחזירה רשימה חדשה כאשר הערך x נוסף מקום אחד אחרי pos. אם pos=null, תוחזר רשימה בה x התווסף לפני חוליה ראשונה. יעילות הפעולה היא $O(1)$ .	21
<code>public static Node&lt;int&gt; Insert (Node&lt;int&gt; list, int step, int x)</code>	פעולה המקבלת רשימה, מספר צעדים step וערך x ומחזירה רשימה חדשה בה הערך x נוסף אחרי step צעדים מתחילת הרשימה. אם $0=x, step$ , יתווסף לפני חוליה ראשונה. אם step גדול מאורך הרשימה, x יתווסף בסוף הרשימה.	22
<code>public static Node&lt;int&gt; Delete (Node&lt;int&gt; list, Node&lt;int&gt; prev)</code>	פעולה המקבלת רשימה ומקום ברשימה prev ומחזירה רשימה חדשה ללא החוליה העוקבת ל prev. אם prev=null, תוחזר רשימה ללא החוליה הראשונה. יעילות הפעולה היא $O(1)$ .	23
<code>public static Node&lt;int&gt; Delete (Node&lt;int&gt; list, int step)</code>	פעולה המקבלת רשימה ומספר צעדים step מתחילת הרשימה ומחזירה רשימה חדשה ללא הערך הנמצא במיקום step.	24

<b>(ו) פעולות על רשימות ממוינות</b>		
<code>public static Node&lt;int&gt; InsertSort (Node&lt;int&gt; list, int x)</code>	פעולה המקבלת רשימה ממוינת בסדר עולה ומספר שלם x ומחזירה רשימה חדשה הכוללת את הערך x כך שהיא שומרת על המיון שלה.	25
<code>public static Node&lt;int&gt; Merge (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות ממוינות בסדר עולה. בכל רשימה הערכים שונים זה מזה. הפעולה מחזירה רשימה חדשה שהיא מיזוג של שתי הרשימות. כל ערך מופיע רק פעם אחת לכל היותר. הרשימה החדשה מכילה את הערכים משתי הרשימות בסדר עולה. מותר לשנות את הרשימות המקוריות. המיזוג יתבצע ביעילות ליניארית.	26
<code>public static Node&lt;int&gt; Cut (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות ממוינות בסדר עולה. בכל רשימה הערכים שונים זה מזה. הפעולה מחזירה רשימה חדשה שהיא חיתוך של שתי הרשימות. כלומר: הרשימה החדשה תכיל את הערכים הנמצאים בשתי הרשימות והיא בסדר עולה. מותר לשנות את הרשימות המקוריות. החיתוך יתבצע ביעילות ליניארית.	27

<b>( ז ) פעולות על שתי רשימות</b>		
<code>public static bool IsAllExist (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות list1, list2, ומחזירה 'אמת' אם כל איברי list2 נמצאים בסדר כלשהו ב-list1, 'שקר'-אחרת.	28
<code>public static bool IsSameValues (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות list1, list2, ומחזירה 'אמת' אם שתי הרשימות מכילות אותן ערכים ללא השיבות לסדר, 'שקר'-אחרת.	29
<code>public static bool IsAllDifferent (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות list1, list2, ומחזירה 'אמת' אם לשתי הרשימות אין ערכים משותפים, 'שקר'-אחרת.	30
<code>public static bool IsBeginWith (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות list1, list2, ומחזירה 'אמת' אם רשימה list2 מוכלת בשלמותה ברשימה list1 החל מראשית הרשימה list1, 'שקר'-אחרת.	31
<code>public static bool IsInclude (Node&lt;int&gt; list1, Node&lt;int&gt; list2)</code>	פעולה המקבלת שתי רשימות list1, list2 ומחזירה 'אמת' אם רשימה list2 מוכלת בשלמותה ברשימה list1, 'שקר'-אחרת.	32

<b>( ח ) פעולות מיון רשימות</b>		
<code>public static Node&lt;int&gt; CreateSorted()</code>	פעולה הבונה רשימה ממוינת בסדר עולה ממספרים שנקלטו מהמשתמש. סוף קלט 99. הפעולה מחזירה הפנייה לחוליה ראשונה ברשימה, null אם היא ריקה.	33
<code>public static Node&lt;int&gt; InsertionSort (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים בסדר כלשהו ומחזירה רשימה ממוינת בסדר עולה בשיטת <b>מיון-הכנסה insertion-sort</b>	34
<code>public static Node&lt;int&gt; SelectionSort (Node&lt;int&gt; list)</code>	פעולה המקבלת רשימה של מספרים בסדר כלשהו וממיינת את הרשימה בסדר עולה בשיטת <b>מיון-בחירה selection-sort</b>	35

**הנחיה והסבר:**

במיון-הכנסה יוצרים רשימה ריקה ומכניסים אליה את הערכים מהרשימה תוך שמירה על סדר עולה. פעולה זו נעשה שימוש בפעולה **הכנס-לרשימה-ממוינת**.  
במיון-בחירה עוברים על הרשימה המקורית ומוצאים את הערך הקטן ביותר ומחליפים אותו עם הערך הנוכחי.