

תאריך: 14.12.2025

## מבוא למדעי המחשב – כיתה יא-1

### דף עבודה – מספר #4 בנושא רשימות מקושרות (Linked Lists)

#### מבוא

התרגיל הזה מבוסס על דברים שלמדנו וראינו בתירגול מספר #3:

- שימוש בחוליה מקשרת
- הפרדה בין החוליה המקשרת למחלקה-אובייקט שמכיל את הנתונים
- עקרון הפרדת האחריות
- כמו כן נלמד ונדגיש העתקה רדודה לעומת העתקה עמוקה (למשל כשיוצרים רשימה חדשה על בסיס רשימה קיימת)

לצורך התרגיל נשתמש באותן מחלקות של התרגיל האחרון:

```
public class IntData
{
    public int value;
    public bool special;

    public IntData(int value)
    {
        this.value = value;
        special = false;
    }

    public void Print()
    {
        Utils.PrintIntAndBeep(value, special);
    }

    public void Mul(int n)
    {
        value *= n;
    }
}
```

```
public class IntNode
{
    public IntData data;
    public IntNode next;

    public IntNode(IntData data, IntNode next = null)
    {
        this.data = data;
        this.next = next;
    }
}
```

## תרגילים לכיתה ולבית

- הערה: את התרגילים יש לכתוב בכתב יד במחברת, כמו בבחינה.
- לאחר מכן יש לממש במחשב.
- כאשר ממשים במחשב, לאחר כל פעולה יש להדפיס את הרשימה בכדי לוודא שהיא עובדת היטב (ובכלל, לדאוג את הקוד – כלומר לוודא שכיסינו את המקרים השונים שיכולים להיות).
- לא לשכוח שרשימה תמיד יכולה להיות ריקה, או להכיל רק איבר אחד.
- כתבו לצורך הדיבוג פונקציות שונות שיאפשרו לכם לוודא מה קורה ולדאוג, לדוגמא  
`static void MakeSpecial(int n)` שעוברת על הרשימה ומסמנת כמיוחדים את כל האיברים שהם כפולה של הערך n.
- אם אתם מוצאים צורך – הגדירו פונקציה שתוכלו להשתמש בה בשאלות שונות. לדוגמא כדי לשקול יצור פונקציה `AddAtHead or AddAtTail`

1. כתוב פונקציה סטטית `PrintList` שמקבלת רשימה ומדפיסה אותה.
2. נתונה הפונקציה `public static IntData CreateInt()` – השתמש בפונקציה הזאת וכתוב פונקציה סטטית `IntNode CreateList(int n)` שמייצרת רשימה של n איברים, על-ידי הכנסת כל איבר חדש בראש הרשימה. שימו לב שהפונקציה מחזירה את הרשימה שהיא יצרה.
3. בדומה לשאלה 2, אבל הפעם נכניס כל איבר חדש שנוצר בסוף הרשימה. שם הפונקציה יהיה `CreateListAtTail`. דרישה: הפונקציה הזאת צריכה להיות יעילה מבחינת זמן החישוב. (יעילה מבחינת החישוב: במקום לעבור על הרשימה בכל פעם, עד האיבר האחרון, יהיה לנו משתנה שנכריז עליו למשל: `IntNode tail = null;` בכל פעם שנוסיף איבר לסוף הרשימה, נעדכן את המשתנה הזה כך שיפנה אל האיבר שכרגע הכנסנו. באופן כזה נחסוך הפעולות שדרושות בכדי לסרוק בכל פעם את כל הרשימה. שימו לב שבהכנסת האיבר הראשון אנחנו צריכים לעדכן גם את `head` וגם את `tail`).
4. כתוב פונקציה שמעבירה את האיבר הכי גדול ברשימה לסופה.  
`IntNode MoveBiggestToEnd(IntNode head)`
5. בהתבסס על הפונקציה מהסעיף הקודם כתוב פונקציה שממיינת את הרשימה מהקטן לגדול.
6. נניח כעת שהרשימה כבר ממוינת. כתוב פונקציה שמכניסה איבר חדש (קוראת ל `CreateInt`) למקום המתאים בתור רשימה ממוינת. כלומר, לאחר ההכנסה הרשימה נשארת ממוינת מהקטן אל הגדול.
7. כתוב פונקציה שמקבלת רשימה ואובייקט `IntData`. הפונקציה מחפשת אם האובייקט נמצא ברשימה ומסירה אותו מהרשימה אם הוא נמצא. הפונקציה מחזירה את הרשימה המעודכנת (כלומר `head`). תזכורת, גם כאן, כמו במקרים אחרים קחו בחשבון את המקרים השונים – הרשימה ריקה, האיבר שצריך להוציא הוא הראשון ברשימה, האיבר שצריך להוציא הוא מאמצע הרשימה.
8. כתוב פונקציה כמו בסעיף הקודם, אבל הפעם היא עובדת על רשימה ממוינת. (הנחיה: מאוד דומה למקרה הקודם, אבל אם האיבר לא נמצא ברשימה, אין צורך לעבור על הרשימה עד הסוף. אפשר להפסיק את המעבר כאשר...).

**תרגילים מתקדמים קצת יותר – העתקת רשימה – העתקה רדודה לעומת העתקה עמוקה**

התרגילים הבאים עוסקים ביצירת רשימה חדשה על בסיס רשימה קיימת.

יש שתי אפשרויות לעשות אז זה:

- א. העתקה רדודה. מעתיקים את ההפניות ובעצם, האיברים ברשימה החדשה הם אותם איברים ברשימה הקודמת.
  - ב. העתקה עמוקה – יוצרים רשימה שבה כל איבר הוא יצירה חדשה של האיבר המקורי.  
נרחיב על הנושא הזה בכיתה אחרי חג החנוכה. אני מציע שתתנסו בזה לפני כן בעצמכם.
9. כתוב פונקציה `CreateListOfSpecialShallow` שמקבלת רשימה ומחזירה רשימה שמכילה רק את האיברים שהם `Special`. שימו לב, מתבצעת העתקה רדודה של האיברים.
  10. כתוב פונקציה `CreateTwoListShallow` שמקבלת רשימה ומחזירה מערך שמכיל שתי רשימות. מקום 0 במערך יכיל רשימה של כל האיברים שאינם `special` ואילו מקום 1 במערך יכיל רשימה של כל האיברים שהם `special`. גם כאן מתבצעת העתקה רדודה של האיברים.
  11. הכפילו את הרשימה המקורית ב 2 והדפיסו אותה. הדפיסו את הרשימות שיצרתם בשאלה 11. מה אתם מצפים לראות? האם ההדפסה מייצגת את מה שציפיתם?
  12. חזרו על סעיפים 9 + 10 – הפעם העתקה צריכה להיות עמוקה. כלומר ברשימה החדשה שנוצרת, כל אובייקט `IntData` הוא עותק חדש (נוצר מחדש על ידי `new`), ולא רק הפניה אל האובייקט המקורי. את תשכחו לבצע הכפלה של הרשימה בערך כלשהו, ובדקו על ידי הדפסה את התוצאה.