

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:

רוני אלנקרי



נעים להכיר – מחלקת חוליה Node

תשובות

עד עתה הגדרנו מחלקות שבהן התכונות הן **מטיפוסים בסיסיים** וכן מחלקות שבהן תכונות הן **עצמים אחרים** (עצמים מורכבים).

1. נתונה המחלקה Date חרוז מיוצג באמצעות התכונות צבע ותאריך ייצור. כתבו את כותרת המחלקה חרוז Bead ותכונותיה.

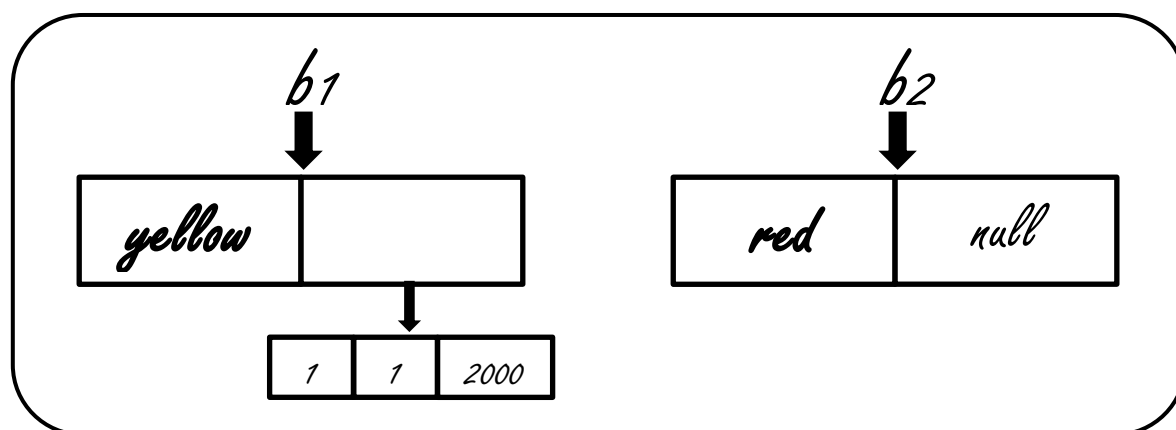
```
public class Bead {
    String color;
    Date date;
}
```

נתון קטע התוכנית הבא :

```
b1 = new Bead("yellow" , new Date(1,1,2000));
```

```
b2 = new Bead("red" , null);
```

2. ציירו את המבנה של החרוזים שנוצרו .



חרוז Bead הוא עצם שמורכב מעצם אחר.

3. עתה נחלף את התכונה תאריך ייצור של חרוז Date date; לתכונה Bead next ;

גם עתה יהיו החרוז עצם מורכב.

מה בכל זאת שונה בהגדרת החרוז החדשה?

יש כאן הוצעה בקורסיבית חרוץ הוא עצם אחת והתכונות שלו הוא עצם מאותה המחלקה.

אנו רואים כי ניתן להגדיר עצם מורכב שאחת התכונות שלו היא עצם מאותה המחלקה.

4. הגדירו במחשב את המחלקה Bead (שיטה בונה לא ריקה, פעולות get , set , toString)

```
public class Bead
{
    private String color;
    private Bead next;

    public Bead(String color, Bead next)
    {
        ...
    }
    public String getColor()
    {
        ...
    }
    public Bead getNext()
    {
        ...
    }
    public void setNext (Bead next)
    {
        this.next = next;
    }

    public String toString()
    {
        ...
    }
}
```

שימו לב כי אין צורך בפעולה setColor.
מדוע?

5. א. הגדירו במחלקת main את החרוזים הבאים :

```
b1 = new Bead("yellow" , null);
b2 = new Bead("green" ,null);
b3 = new Bead("red" , null);
```

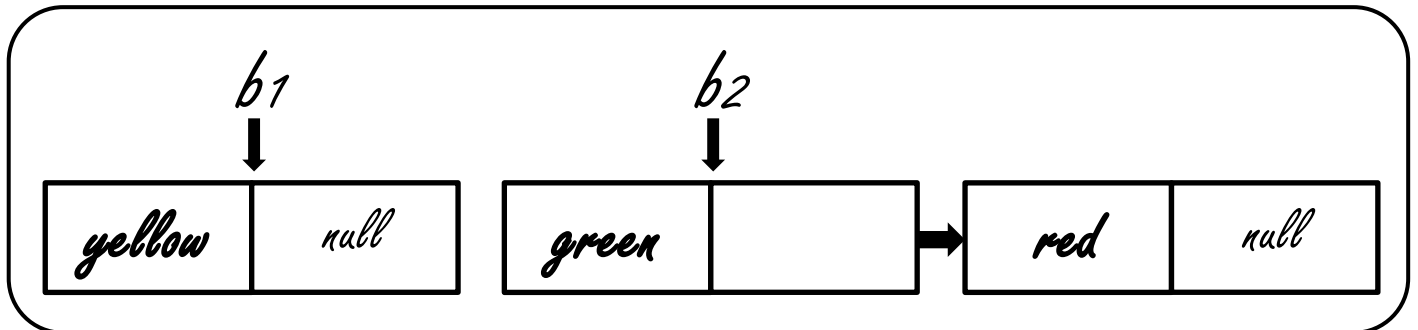
כמה חרוזים נוצרו ? 3

ב. עתה הגדירו את החרוזים באופן הבא (שימו לב לשינוי בהגדרת החרוזים) :

```
b1 = new Bead("yellow" , null);
```

```
b2 = new Bead("green", new Bead("red", null) );
```

ג. ציירו את החרוזים שנוצרו



6.א. הוסיפו לתוכנית פקודות מתאימות להדפסת החרוזים.

```
System.out.println(b1);
System.out.println(b2);
System.out.println(b2.getNext());
```

ב. האם הצלחתם להדפיס את כל החרוזים שנוצרו? _____

ג. מה ההבדל בין ההפניה לחרוז השלישי לעומת ההפניה לחרוז הראשון? לחרוז הפאסון קיימת

הפניה ע"י מעטפה בלבד ואילו לחרוז הפאסי קיימת הפניה מחפוף אחת. החרוז b2

הוסיפו לתוכנית את הפעולות הבאות:

```
b1.setNext(b2);
b2=null;
```

ד. הדפיסו עתה את שלושת החרוזים.

```
System.out.println(b1);
System.out.println(b1.getNext());
System.out.println(b1.getNext().getNext());
```

אז מה למצוא? עזרת?

I. ניתן להשגיף לעצמך תכונה שהיא הפניה לעצמך מאותה המחלקה

II. עצמך ככה אנו מכנים חוליה Node

III. ניתן להשתמש לחוליה אחת ציפף חוליה אחת

7. נתון קטע התוכנית הבא: (הוסיפו אותו לתוכנית שלכם)

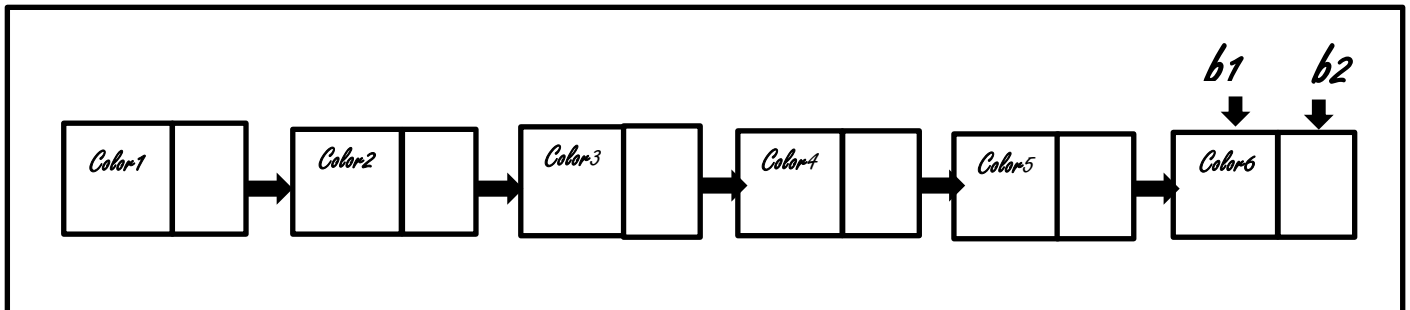
```
System.out.println(" הכנס צבע ");
String c = in.next();
b1 = new Bead(c, null);
for (int i=0 ; i<5; i++)
{
    System.out.println(" צבע הכנס ");
    c = in.next();
    b2 = new Bead(c, null);
    b1.setNext(b2);
}
```



```
b1=b2;
}
```

א. כמה חוליות נוצרו ? 6

ב. ציירו את המבנה שנוצר.



ג. הוסיפו קטע תוכנית שידפיס את כל החוליות שבשרשרת שנוצרה, הריצו ובדקו את הפלט.

ד. האם הצלחתם ? לא חסרה הפנייה לחוליה הראשונה

ה. הוסיפו את הפקודה הבאה בקטע התוכנית שיוצר את שרשרת החוליות.

א. מה מטרת הפקודה לדעתכם? למנוף במקרה first הפנייה לחוליה הראשונה

```
System.out.println(" הכנס צבע ");
String c = in.next();
b1 = new Bead(c , null);
```

הוסיפו את פקודה הבאה

```
Bead first = b1;
```

```
for (int i=0 ; i<5; i++)
{
    System.out.println("צבע הכנס");
    c = in.next();
    b2 = new Bead(c, null);
    b1.setNext(b2);
    b1=b2;
}
```

ב. עתה כתבו קטע תוכנית שידפיס את החוליות שבשרשרת. הריצו ובדקו את הפלט.

ג. תפקידה של ההפניה first : להפנות לחוליה הראשונה בשפת מוצר



אז מה למצוא עז עתה ?

- I. ניתן באמצעות חוליה שיש לה הפניה לחוליה מאותה מחלקה לבנות שפספת של חוליות
- II. בשפספת כזו ניתן לעשות לכל חוליה באמצעות ההפניה אליה קודמת (getNext) לכן אין צורך לשאוף את ההפניה לחוליה על ידי מסתנה נוסף
- III. כיוון שלחוליה הפאסונה אין חוליה קודמת יש לשאוף את ההפניה אליה ע"י מסתנה נוסף **first**
- IV. על מנת לעבור על המבנה שנוצר (שפספת חוליות) נצטרק ל-2 ממציאים first שיכול תמיד את ההפניה חוליה הפאסונה / pos (position) שיצביע בכל פעם על חוליה אחרת בשפספת החוליות איתו נתקדם בשפספת החוליות

בקטע התוכנית הקודם יצרנו שרשרת עם 6 חוליות. כמובן שניתן לייצר שרשרת חוליות בכל גודל .

9. א. האם יש צורך להגדיר מראש את גודל השרשרת (בדומה למערך)? לא
- ב. הוסיפו את הדרוש לקטע התוכנית הבא שמטרתו להדפיס את שרשרת החוליות שייצרנו על ידי לולאת while. וודאו שהודפסו כל החוליות בשרשרת.

```

Bead pos = first ; // הפניה לחוליה הראשונה בשרשרת
while ( pos != null ) // כל עוד מפנה לחוליה קיימת
{
    System.out.println (pos.getInfo()); // הדפסת צבע החרוז
    pos = pos.getNext() // התקדם לחוליה הבא
}
    
```

ג. נתון קטע התוכנית הבא :

```

Bead first = null;

for (int i=0 ; i<6; i++)
{
    System.out.println("צבע הכנס");
    c = in.next();
    b1 = new Bead(c, first);
    first = b1;
}
    
```

עקבו אחר קטע התוכנית וציירו את השרשרת שנוצרה. הריצו על גבי המחשב ובדקו את תשובתכם ע"י הוספת קטע המדפיס את השרשרת שנוצרה.

אז מה למצוא עז עתה ?

- I. ניתן באמצעות חוליה להציג מבנה של מספר חוליות שפספת חוליות
- II. אין צורך להציג מראש את גודל השפספת
- III. אנו נשאוף את ההפניה לחוליה הפאסונה בשפספת



המחלקה חוליה Node<T>

נתונה המחלקה חוליה - Node. העתיקו אותה לפרויקט. המחלקה תשמש לנו לייצר ולעבוד עם שרשרת חוליות מכל טיפוס. לשרשרת כזו נקרא רשימה. בסוף דפי העבודה מצורף ממשק המחלקה.

1. להלן קטע תוכנית לבניית שרשרת חוליות תוך שימוש במחלקה Node. הריצו אותו והדפיסו את השרשרת שנוצרה ע"י לולאת while תוך שימוש בממשק.

```
Node <Integer> first = null;
Node <Integer> pos = null;
for (int i=1 ; i<=10; i++)
{
    if (i==1) // החוליה הראשונה
    {
        pos = new Node<Integer> (i*2)
        first = pos; // החוליה הראשונה
    }
    else
    {
        pos.setNext(new Node<Integer> (i*2));
        pos = pos.getNext();
    }
}
```

2 → 4 → 6 → 8 → 10 → 12 → ... → 20

2. א. מה יבצע הקטע הבא (לאחר יצירת השרשרת בסעיף הקודם) :

תקנוסל החוליה שצפחה 0 להיות החוליה הראשונה

```
pos = new Node<Integer> (0);
pos.setNext(first);
first = pos;
```

ב. הדפיסו את השרשרת החדשה.

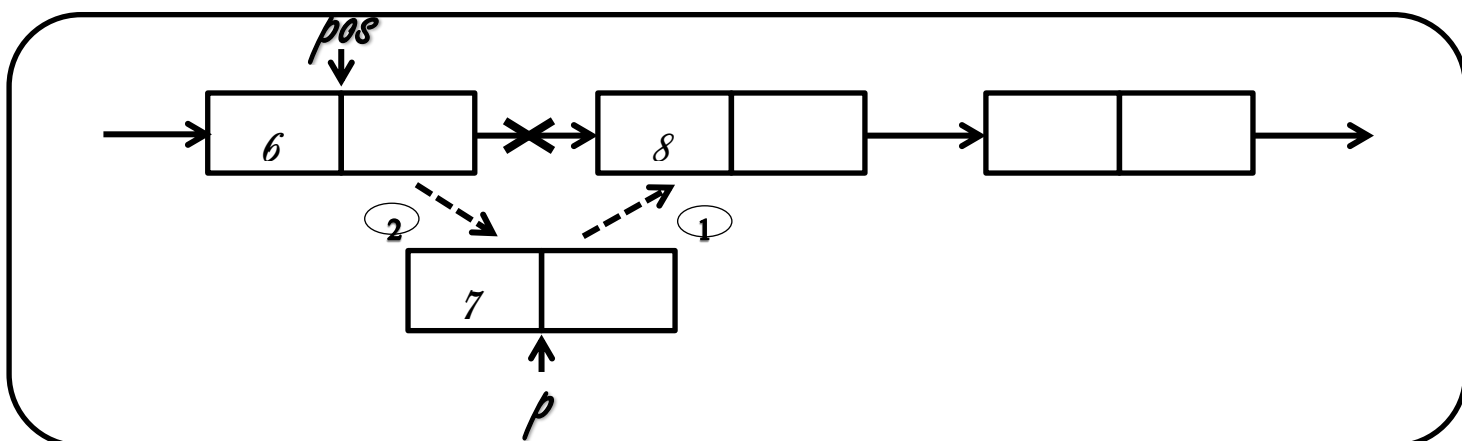
3. מה יבצע הקטע הבא :

```
pos = first.getNext().getNext();
Node <Integer> p = new Node<Integer> (7);
p.setNext(pos.getNext());
pos.setNext(p);
```

תוסף החוליה שערך 7 לאחר החוליה שערך 6

4. להלן שרטוט המדגים את הפעולות בקטע התוכנית האחרון. וודאו שאתם מבינים את השרטוט.

מדוע יש לבצע את הפעולות בסדר הכתוב, ראשית יצירת ההפניה שמסומנת ב-1 ואח"כ ההפניה המסומנת ב-2? אם נשנה קודם את ההפניה החוליה pos "נאבז" את החוליה שערך 8 כך שלא נוכל להפנות את החוליה p אליה.



אז מה למצוא עז עתה ?



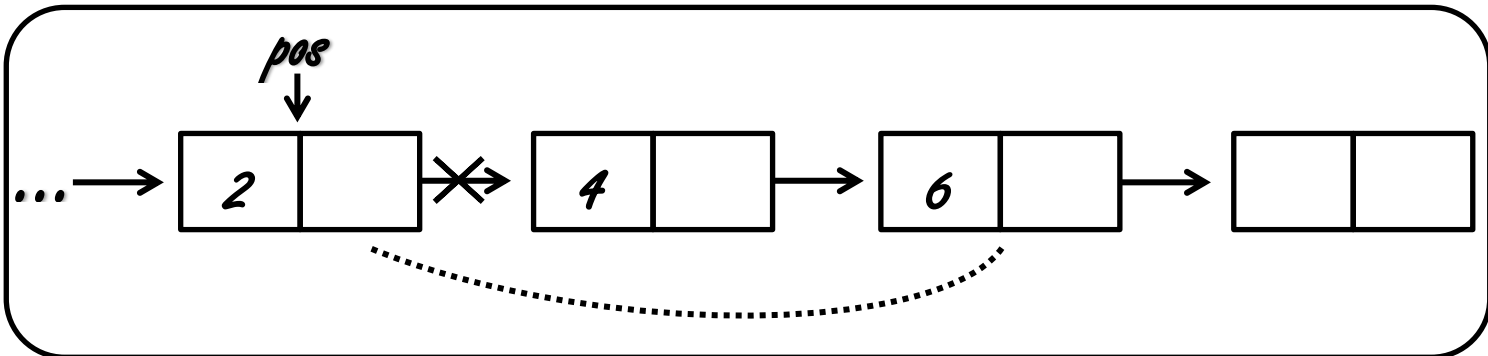
- I. ניתן להוסיף לשפת חוליות חדשות
- II. יש ליצור חוליה חדשה ואז לעצבן את ההפניות בשפת.
- III. אם מכיסיס חוליה במקום הפאסון יש לעצבן את הפניה לפאסון first
- IV. אם מכניסיס חוליה במקום שאינו הפאסון יש לעמוד על המקום שלאחריו תיכנס החוליה החדשה. סגף שינוי ההפניות יהיה : ראשית נשנה את החוליה החדשה לחוליה הבאה אחריה ושנית נשנה ת החוליה הקודמת לה אליה

אם ברצוננו למחוק חוליה מהשרשרת אין צורך למחוק אותה בפועל. עלינו לשנות את ההפניה שמצביעה על החוליה כך שתצביע על החוליה הבאה בתור. כאשר אין אף איבר בשרשרת שיצביע על החוליה היא נמחקת.
("אספן הזבל" של java דואג לפנות אותה בשלב מסוים מהזיכרון)

5. כתבו קטע תוכנית המוחק את החוליה עם הערך 4. שימו לב לאיור המדגים את שלבי הפעולות שעליכם לעשות :

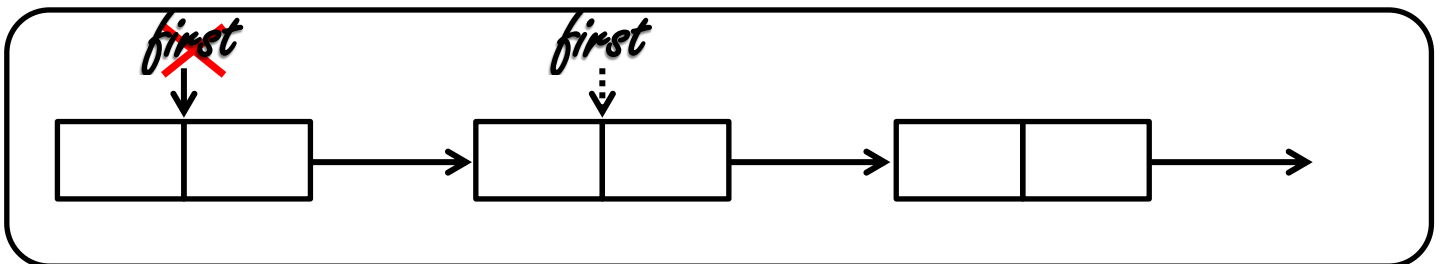
- א. להציב את pos על החוליה שערכה 2
- ב. חוליה זאת הינה חוליה שמיקומה (לפני/אחרי/אותה) לפני החוליה שברצוננו למחוק

ג. לשנות את ההפניה next בחוליה pos כך שתפנה לחוליה שאחרי החוליה שברצוננו למחוק
וההפניה אליה היא : `pos.getNext().getnext()`



6. כתבו את הפעולה המוחקת את החוליה הראשונה. שימו לב לאיור המדגים את הפעולה שעליכם לבצע.

`first = first.getNext();`



אז מה למצוא עז עתה ?



- I. ניתן למחוק חוליות מהשפת.
- II. אם מוחקים חוליה ראשונה יש לשנות את first כך שיפנה לחוליה השנייה.
- III. אם מוחקים חוליה שאינה הראשונה יש לעמוד על החוליה שאחריה ולהפנות אותה לחוליה אחת אחרי החוליה שבוצעו למחוק. (לדאגה על החוליה שבוצעו להבטא.)
- IV. מנסיון "אספן הבל" של java יצרה חוליות שאין להן הפניות ויפנה אותן מהליכיון