

מאגר מעבדות לשפות התכנות החדשות

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מד"ר תמר פז.

המעבדה בקובץ זה מיועדת לתלמידים הלומדים מדעי המחשב בשפת התכנות ג'אווה והיא מותאמת לסביבת אקליפס.

המעבדה מיועדת לשיעורי המעבדה והיא מבוססת על שיטת ההוראה לפיה הלימוד של כל נושא חדש יפתח בהתנסות אישית במעבדה. לאחריה, יבוא דיון כיתתי, שבעקבותיו ייפתרו משימות שונות.

המעבדה מתרכזת ב בלימוד נושא מרכזי אחד: רשימות מקושרות.

עצם מורכב שאחת מתכונותיו היא מטיפוס עצמו. עמוד 1

חוליה ורשימה מקושרת עמוד 5

המחלקה הגנרית עמוד 9

עצם מורכב שאחת מתכונותיו היא מטיפוס עצמו

המחלקה *Bead* לקוחה מפרק 6 בספר "עיצוב תכנה מבוסס עצמים", תשס"ח 2007, המרכז להוראת המדעים, האוניברסיטה העברית, ירושלים

משימה 1 – חלק א'

עצם מורכב הוא עצם שבו עצם אחר משמש כתכונה, ובפרט העצם שמשמש כתכונה יכול גם להיות מטיפוס המחלקה של העצם המורכב. במילים אחרות, עצם מורכב יכול גם להיות עצם שאחת מתכונותיו היא הפניה לעצם מאותו הטיפוס! למשל, תרשים UML הבא מתאר את המחלקה *Bead* (חרוז).

Bead
String color Bead nextBead
Bead (String color) Bead (String color, Bead nextBead) String getColor() Bead getNextBead() void setNextBead (Bead nextBead)

- לעצם מטיפוס המחלקה *Bead* יש שתי תכונות: *color* צבע החרוז מטיפוס _____, ו-*nextBead* הפניה לחרוז המושחל אחריו בשרשרת. ההפניה היא מטיפוס *כתובת fe*

```
public Bead (String color)
{
    this.color = color;
    this.nextBead = null;
}
```

- למחלקה *Bead* יש שתי פעולות בונות. פעולה בונה אחת היא הפעולה הבאה שמקבלת רק את הצבע של החרוז, ומציבה בתכונה *nextBead* את הערך *null*.

null הוא הפניה ריקה שאינה מובילה לשום עצם. לכן, הפעולה הבונה הנוכחית דואגת שבתכונה *nextBead* לא תהיה הפניה לשום עצם.

- הפעולה הבונה השניה מקבלת צבע של חרוז והפניה לעצם מטיפוס *Bead* (חרוז).

בעמוד הבא, מופיע שלד של המחלקה *Bead*.

- לפניכם שלד של המחלקה `Bead`. השלימו אותו.

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

    public Bead (String color)
    {
        this.color = _____;
        this.nextBead = null;
    }

    public Bead (String color, Bead nextBead)
    {
        this.color = color;
        this.nextBead = nextBead;
    }

    public String getColor()
    {
        return this.color;
    }

    public Bead getNextBead()
    {
        return _____;
    }

    public void setNextBead (Bead nextBead)
    {
        this.nextBead = _____;
    }
}
```

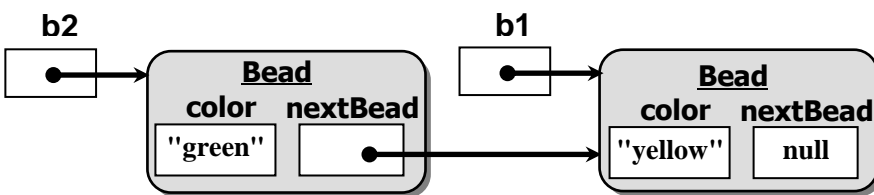
בתכונה `nextBead` של העצם הנוכחי אין שום הפניה

- פיתחו מחלקה בשם `Bead` וממשו בה את המחלקה `Bead`.

```
public static void main (String[] args)
{
    Bead b1 = new _____ ("yellow");
    _____ b2 = new _____ ("green", b1);
}
```

משימה 1 – חלק ב'

הפעולה הראשית הבאה יוצרת שני חרוזים (צהוב וירוק) ויוצרת מהם שרשרת בהתאם לאיור הבא.



- פיתחו מחלקה חדשה בתוך הפרויקט המכיל את המחלקה `Bead`.

- השלימו את הפעולה הראשית וכתבו אותה במחלקה החדשה.

משימה 1 – חלק ג'

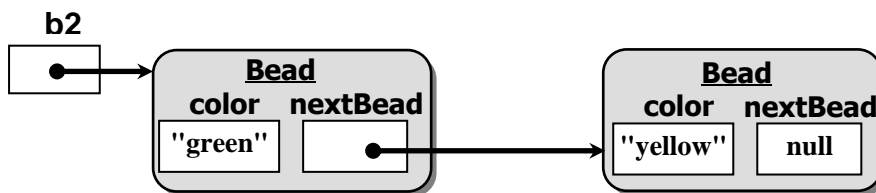
```
private static void printChain(Bead first)
{
    System.out.print (first.getColor()+ "--");
    Bead second = first.getNextBead();
    System.out.print (second.getColor()+ "--");
    System.out.println ("end");
}
```

כדי לבדוק שאכן יצרתם שרשרת הוסיפו למחלקה הראשית את הפעולה printChain הבאה שמדפיסה את השרשרת. בפעולה הראשית, לאחר יצירת השרשרת, זמנו את הפעולה printChain להדפסת השרשרת.

הזימון של הפעולה printChain יראה כך: printChain(b2)

משימה 1 – חלק ד'

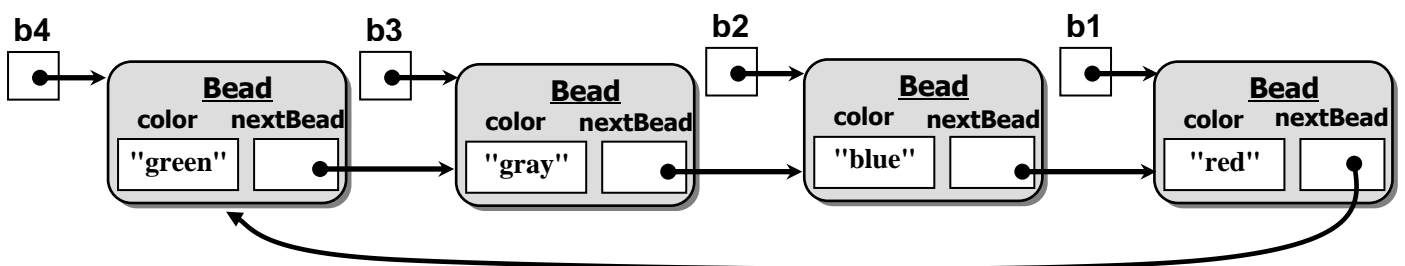
שנו את הפעולה הראשית כך שהשרשרת הנוצרת תראה כך:



גם הפעם זמנו את הפעולה printChain להדפסת השרשרת.

משימה 2

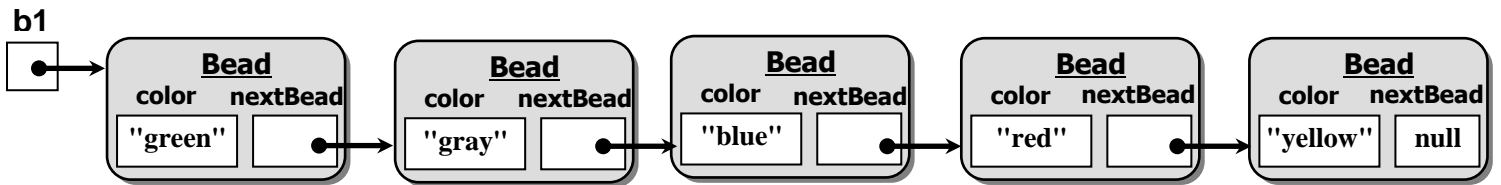
צרו שרשרת מעגלית שתכיל 4 חרוזים. למשל כך,



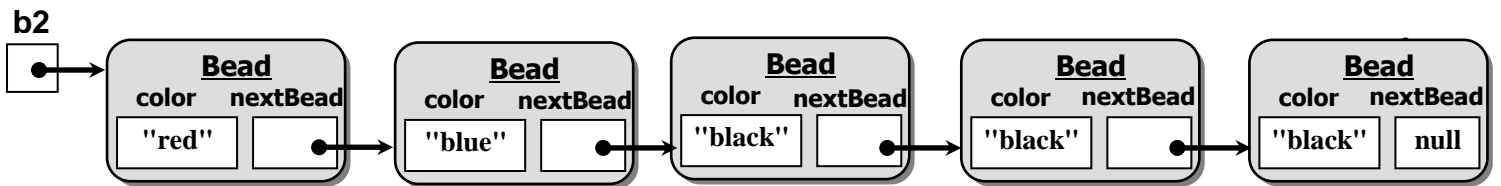
- כדי לבדוק את השרשרת שיצרתם שנו את הפעולה printChain כך שתתאים להדפסת שרשרת באורך 5 (החרוז הראשון יודפס בהתחלה ובסוף). הנחיה: עיברו על השרשרת באמצעות לולאה.

משימה 3 – חלק א'

צרו שתי שרשראות שונות. השרשרת הראשונה תהיה מורכבת מחמישה חרוזים: ירוק, אפור, כחול, אדום, כחול, שחור, שחור, שחור. המשתנה b1 יכיל הפניה לחרוז הראשון של שרשרת זו.



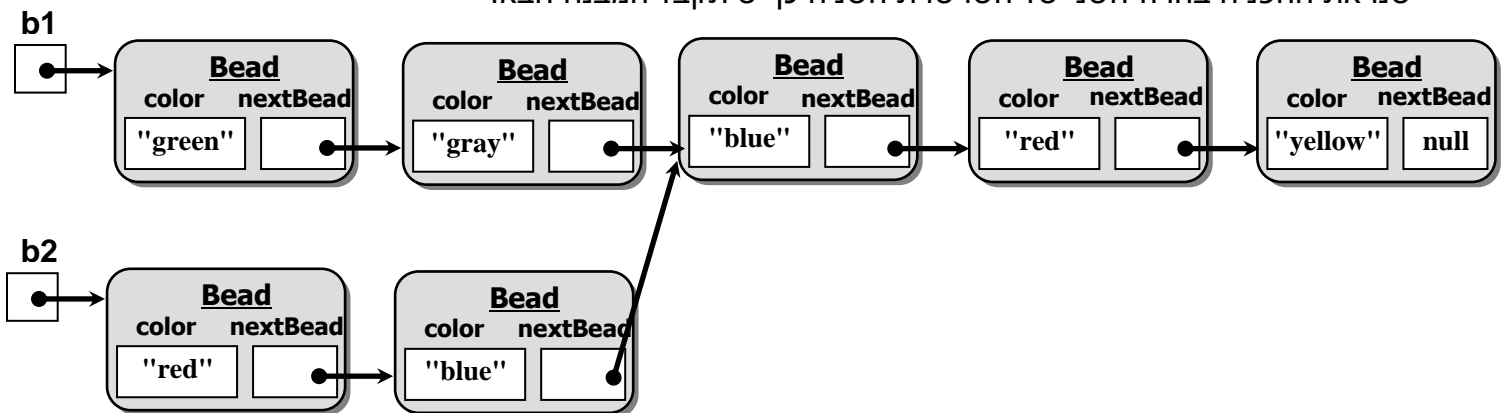
השרשרת השנייה תהיה מורכבת גם היא מחמישה חרוזים: אדום, כחול, שחור, שחור, שחור. המשתנה b2 יכיל הפניה לחרוז הראשון של שרשרת זו.



- הדפיסו את שתי השרשראות (זמנו את הפעולה printChain עבור כל אחת מהשרשראות).

משימה 3 – חלק ב'

שנו את ההפניה בחרוז השני של השרשרת השנייה כך שיתקבל המבנה הבא:



- כדי לבדוק את השינוי שעשיתם, צריך לזמן את הפעולה printChain כאשר המשתנה _____ משמש לה כפרמטר.
- זמנו את הפעולה printChain וודאו כי חיברתם את השרשראות כנדרש.

משימה 4

שנו את הפעולה printChain כך שתתאים לשרשרת לינארית (לא מעגלית) באורך לא ידוע מראש. רמז: יש לעבור על השרשרת עד שמגיעים לחרוז שערך nextBead שלו הוא _____.

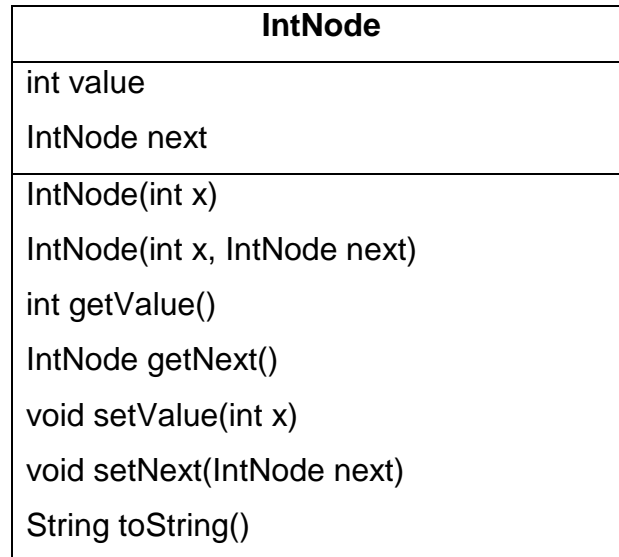
- צרו שרשראות שונות באורכים שונים ובדקו באמצעותם את הפעולה שכתבתם.

חוליה ורשימה מקושרת (שרשרת חוליות)

משימות 5-6 מבוססת על דפי עבודה שנכתבה על-ידי:
מאצון אבן חיים, דורית ליקרמן ודורית כהן

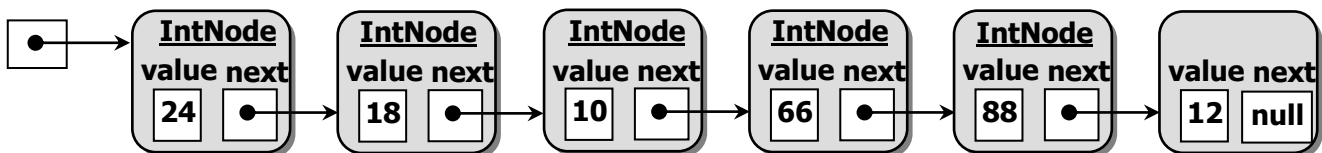
משימה 5 – חלק א'

תרשים UML הבא מתאר את המחלקה IntNode (חוליה מטיפוס int).



- לעצם מטיפוס המחלקה IntNode (או: למופע של המחלקה IntNode) יש שתי תכונות: ערך כלשהו מטיפוס _____, והפניה למופע נוסף של המחלקה IntNode.

עצם מטיפוס המחלקה IntNode נקרא **חוליה**. כיוון שאפשר לבנות מספר עצמים שבתכונה next של כל אחד מהם תהיה הפניה לעצם נוסף מטיפוס IntNode וליצור בכך **שרשרת חוליות** (כמו שחיברנו מחרוזת של חרוזים). למשל כך:



הערך השמור בתכונה Value הוא **הערך של החוליה**, והערך השמור בתכונה next הוא הפניה לחוליה **העוקבת**, או null במידה ואין חוליה עוקבת.

בעמוד הבא מופיע ממשק המחלקה IntNode. היעזרו בו כדי לממש את המחלקה

- לפניכם ממשק המחלקה IntNode.

IntNode (int x)	הפעולה בונה חוליה. הערך של החוליה הוא x, ואין לה חוליה עוקבת.
IntNode (int x, IntNode next)	הפעולה בונה חוליה. הערך של החוליה הוא x, והחוליה העוקבת לה היא next. ערכו של next יכול להיות null.
int getValue ()	הפעולה מחזירה את הערך השמור בחוליה.
IntNode getNext ()	הפעולה מחזירה הפניה לחוליה העוקבת. אם אין חוליה עוקבת הפעולה מחזירה null.
void setValue (int x)	הפעולה משנה את הערך השמור בחוליה ל-x.
void setNext (IntNode next)	הפעולה משנה את החוליה העוקבת להיות next. ערכו של next יכול להיות null.
String toString ()	הפעולה מחזירה מחרוזת המתארת את החוליה.

- פיתחו פרויקט חדש, פיתחו בו מחלקה IntNode וממשו בה את המחלקה IntNode. רמזים:

- העזרו במחלקה Bead
- הפעולה toString :

```
public String toString ()
{
    return " "+ this.value;
}
```

```

public class IntChain
{
    public static void main (String[] args)
    {
        int num = (int)(Math.random()*100);
        IntNode first = new IntNode (num);
        buildChain (first);
        printChain (first);
    }

    private static void buildChain (IntNode pos)
    {
        int num;
        for (int i= _____; i<5 ; _____)
        {
            num = _____;
            pos.setNext (new IntNode(num));
            pos = pos.getNext();
        }
    }

    private static void printChain (IntNode pos)
    {
        while (pos !=null)
        {
            System.out.print(pos.toString());

            pos = _____;
        }
        System.out.println(" end");
    }
}

```

משימה 5 – חלק ב'

לפניכם שלד של מחלקה ראשית IntChain.

- הפעולה הראשית: 1. מגרילה מספר אקראי, 2. בונה חוליה ומציבה במשתנה _____ את ההפניה לחוליה זו.
- 3. מזמנת את הפעולה buildChain.
- 4. מזמנת את הפעולה printChain.
- ההוראה:

IntNode first = new IntNode (num);

- יוצרת עצם מטיפוס IntNode שבתכונה value שלו מוצב המספר _____, ובתכונה next שלו מוצב הערך _____.
- הפעולה buildChain מקבלת הפניה לחוליה. הפעולה יוצרת רשימה של 6 חוליות שבתכונה value של כל אחת מהן יש מספר אקראי בתחום 0-100. החוליה הראשונה ברשימה היא החוליה שההפניה אליה התקבלה כפרמטר.

▪ ההוראה:

pos.setNext (new IntNode(num));

- יוצרת עצם מטיפוס IntNode ומציבה את ההפניה אליו בתכונה next של העצם שההפניה אליו היא במשתנה _____.

- הפעולה printChain מקבלת הפניה לחוליה ראשונה ברשימה ומדפיסה את כל הרשימה.

- בפעולה printChain, המשתנה pos מקבל הפניה לחוליה הראשונה. בכל "סיבוב" של הלולאה הוא מקבל הפניה לחוליה אחרת. הדבר אינו משפיע על המשתנה first ששומר את ההפניה לחוליה הראשונה ברשימה. מדוע? _____

- השלימו את המחלקה.

- בתוך הפרויקט המכיל את המחלקה IntNode, פיתחו מחלקה חדשה IntChain והקלידו בה את המחלקה IntChain.

- הריצו את המחלקה IntChain.

משימה 5 – חלק ג'

הוסיפו למחלקה הראשית IntChain את שתי הפעולות הבאות:

- פעולה שמקבלת הפניה לחוליה ראשונה ברשימה ומחזירה את המספר הגדול ביותר ברשימה.
 - פעולה שמקבלת מספר שלם והפניה לחוליה ראשונה ברשימה. הפעולה מחזירה true אם המספר מופיע כערך באחת מהחוליות של הרשימה, ומחזירה false אחרת.
- הוסיפו לפעולה הראשית (main) זימון לשתי הפעולות החדשות שכתבתם, ובדקו שהן מבצעות את הנדרש.

אל תמחקו את המחלקה IntChain, נזדקק לה בהמשך!

משימה 6 – חלק א'

- פיתחו בפרויקט הנוכחי מחלקה חדשה CharNode וממשו בה מחלקה דומה למחלקה IntNode. במחלקה זו הערך של התכונה value (הערך של החוליה) יהיה מטיפוס char.

את משימות 6-ב, 8, פיתרו בלי לספור את מספר החוליות ברשימה!

משימה 6 – חלק ב'

- פיתחו בפרויקט הנוכחי מחלקה ראשית חדשה CharChain וכתבו בה את הפעולות הבאות:
 - פעולה buildChain שמקבלת הפניה לחוליה ראשונה של רשימה מקושרת ויוצרת רשימה שכל איבר בה הוא תו בתחום a-j. הפעולה תיצור תווים אקראיים בתחום a-k, תיצור חוליה עבור כל תו שיוגרל ותוסיף אותה לרשימה. התהליך ייפסק כאשר יוגרל התו k (שלא יתווסף לרשימה). אפשר ליצור תווים באופן אקראי כך:

```
char c= (char)((int)(Math.random()*11)+97);
```

- הסבר: יוצרים מספר אקראי בתחום 97-107 (הקוד האסקי של האותיות a-k) והופכים את המספר לתו.
- פעולה שמקבלת הפניה לחוליה ראשונה ברשימה ומדפיסה את כל הרשימה.
 - פעולה שמקבלת הפניה לחוליה ראשונה ברשימה ומוסיפה, לאחר החוליה השלישית, חוליה חדשה ובה התו X. למשל, אם הרשימה היא: t-q-q-q-r-r-s-t-a-t, אזי לאחר הפעולה הרשימה תהיה t-q-q-X-q-r-r-s-t-a-t. הנחה: הרשימה מורכבת מ-3 חוליות לפחות.
 - פעולה ראשית שתבדוק את כל הפעולות שכתבתם.

משימה 7

- הוסיפו למחלקה CharList פעולה שתקבל הפניה לחוליה ראשונה ברשימה ותחזיר את אורך הרצף הגדול ביותר של תווים זהים ברשימה. למשל, עבור הרשימה t-q-q-q-r-r-s-t-a-t, הפעולה תחזיר את המספר 3.
- הוסיפו לפעולה הראשית זימון לפעולה החדשה ולאחריה זימון לפעולה להדפסת הרשימה.

משימה 8

- הוסיפו למחלקה CharChain פעולה שתקבל הפניה לחוליה ראשונה ברשימה. הפעולה תוציא מהרשימה את כל החוליות שהערך של החוליה שלפניהן הוא a. למשל, אם הרשימה היא: t-a-m-p-a-z, אזי לאחר הפעולה הרשימה תהיה t-a-p-a.
- הנחה: הרשימה לא מכילה רצף של a-ים. כלומר לא יופיעו ברצף שתי חוליות או יותר שערךן a.
- הוסיפו לפעולה הראשית זימון לפעולה החדשה ולאחריה זימון לפעולה להדפסת הרשימה.

המחלקה הגנרית

משימה 9 – חלק א'

- המחלקות IntNode ו- CharNode שונות זו מזו רק בטיפוס של התכונה _____ . באופן דומה נוכל לבנות מחלקות נוספות שיגדירו חוליות מטיפוסים שונים. למשל המחלקה DoubleNode תגדיר חוליה מטיפוס _____ .
- כיוון שהשוני בין המחלקות הוא רק בטיפוס של התכונה value, ג'אווה מאפשרת להגדיר מחלקה שבה הטיפוס של התכונה value יקבע רק בעת יצירת מופע של המחלקה.
- תרשים UML הבא מתאר **מחלקה גנרית** Node <T> שבה הערך של הטיפוס יקבע רק בעת השימוש במחלקה. כיוון שהטיפוס עדיין לא נקבע הוא מסומן באות T.

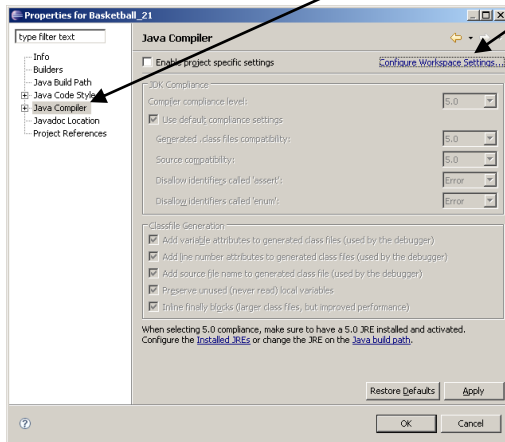
Node<T>
T value
Node<T> next
Node (T x)
Node (T x, Node<T> next)
T getValue()
Node<T> getNext()
void setValue (T x)
void setNext (Node<T> next)
String toString()

במקום טיפוס הנתונים רשמים T, אך בכל מקום שרוצים לציין את השם של המחלקה רשמים **Node<T>**. מסיבה זו, התכונה value היא מטיפוס T, ואילו התכונה next, שערכה יהיה הפניה לעצם (חוליה) מטיפוס המחלקה, היא מטיפוס Node<T>.

- פתחו מחלקה חדשה בשם Node וממשו בה את המחלקה Node<T>

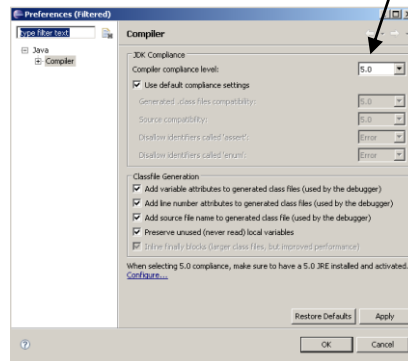
שימו לב, למרות שהשם של המחלקה הוא Node, הכותרת של המחלקה תהיה: **public class Node<T>** אם לאחר הקלדת המחלקה Node מופיעות שגיאות, ראו הנחיות בעמוד הבא.

אם לאחר הקלדת המחלקה Node מופיעות שגיאות, צריך לעדכן את התכנה כך שתשתמש במנגנון הגרניות. לשם כך, בצעו את הפעולות הבאות: בשורת התפריטים העליונה בחרו ב Project. בחלון שנפתח קעת בחרו ב Properties. בחלון שנפתח קעת, לחצו על Java Compiler



Configure Workspace Setting: מצד ימין למעלה יופיע הכיתוב: לחצו עליו.

בחלון שנפתח קעת בחרו ב 5.0, לחצו על Apply, ואשר פעמיים (ok)



משימה 9 – חלק ב'

- מחקו את המחלקה IntNode מהפרויקט.
- שנו את המחלקה IntList כך שבכל מקום שמופיע בה IntNode, ירשם <Integer>Node (שימו לב, Integer עם | גדולה ולא int !).
- הריצו קעת את המחלקה IntList ובדקו כי היא מבצעת את הנדרש לאחר השינוי.
- באופן דומה מחקו את המחלקה CharNode ושנו את המחלקה CharList בהתאם.

במחלקה גנרית, ערך של חוליה (ערך של התכונה value) מוכרח להיות הפניה לעצם (למופע של מחלקה) ולא טיפוס פשוט! כלומר, הוא אינו יכול להיות char, double, int,

לכן, בתוכנת ג'אווה לכל טיפוס פשוט מוגדרת מחלקה מקבילה:

Integer עבור int

Double עבור double

Character עבור char, וכן הלאה

מחלקות אלה נקראות: מחלקות עוטפות (type wrapper classes)

ההמרות מהטיפוסים הפשוטים לעצמים עוטפים, וחזרה, נעשות אוטומטית על ידי התוכנה.

משימה 10

כתבו תכנית שתקלוט עבור כל אחת מהקבוצות הרשומות באיגוד הכדורסל את שמה, את מספר הניצחונות שצברה בעונה האחרונה ואת מספר ההפסדים שספגה. קליטת הנתונים תפסק עם קליטת השם "aaa" (ונדרשנו שלא לקלוט נתונים נוספים עבור שם זה). התכנית תציג כפלט:

א. את השם של הקבוצה שהיחס בין מספר הניצחונות שלה לבין סה"כ המשחקים ששיחקה הוא הגדול ביותר (אם למספר קבוצות יש את היחס הגדול ביותר, התכנית הציגה כפלט את השמות של כולן).
 ב. את השמות של הקבוצות שהיחס בין מספר הניצחונות שלהן לבין סה"כ המשחקים ששיחקו הוא מתחת לממוצע.

השמות של הקבוצות משמשים כמזהים (אין שתי קבוצות או יותר בעלות אותו שם. ואי אפשר לשנות את השמות של הקבוצות).

עד כה, כדי לפתור את המשימה, כתבנו שלוש מחלקות. מחלקה Team שמייצגת קבוצה אחת, מחלקה Union שמייצגת את האיגוד ומחלקה ראשית.

ממשק המחלקה Team

Team(String name)	פעולה בונה. מקבלת שם של קבוצה. קולטת מהמשתמש מספר ניצחונות ומספר הפסדים.
String getName()	מחזירה את השם של הקבוצה.
double getRatio()	מחזירה את היחס בין מספר הניצחונות של הקבוצה לבין סה"כ המשחקים בהם השתתפה.

למחלקה Team הגדרנו שתי תכונות: שם הקבוצה, והיחס בין מספר הניצחונות של הקבוצה לבין סה"כ המשחקים בהם השתתפה.

ממשק המחלקה Union

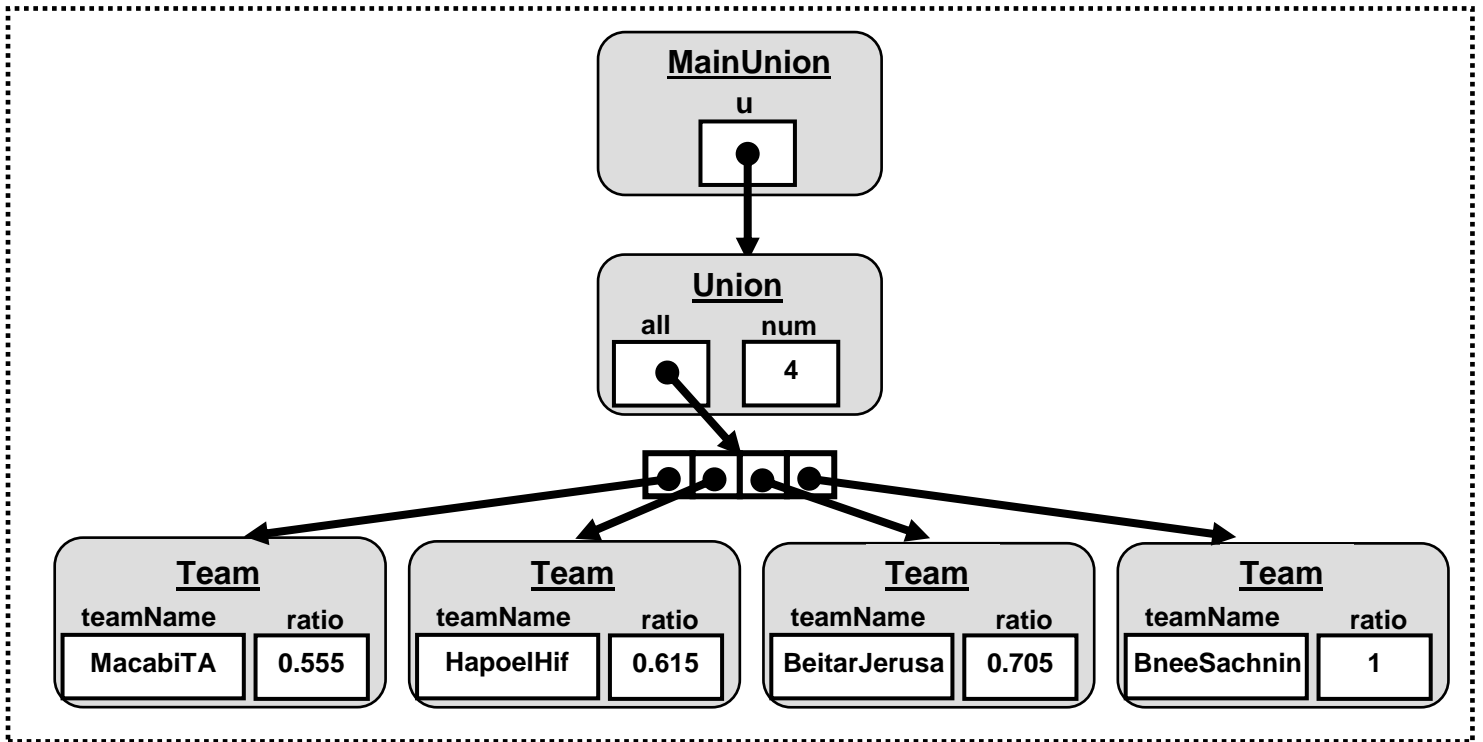
Union ()	פעולה בונה.
void printBiggest ()	פולטת את השמות של הקבוצות שהשיגו את היחס הגדול ביותר בין מספר הניצחונות לבין מספר המשחקים.
void underAverage ()	פולטת את השמות של הקבוצות שהיחס בין מספר הניצחונות שלהן לבין מספר המשחקים הוא מתחת ליחס לממוצע.

למחלקה Union הגדרנו שתי תכונות: מערך שבכל איבר בו תהיה הפניה לעצם מטיפוס Team, ומספר שלם שיציין את מספר הקבוצות באיגוד.

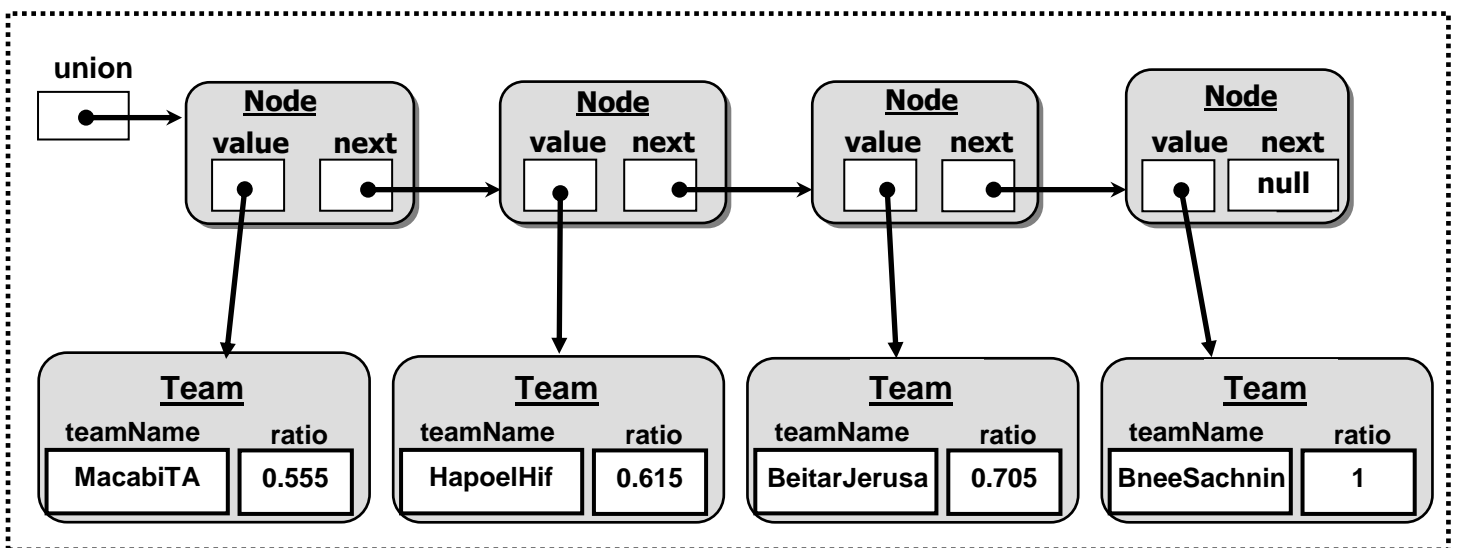
הפעולה הראשית:

```
public class MainUnion
{
    public static void main(String[] args)
    {
        Union u=new Union();
        u.printBiggest();
        u.underAverage();
    }
}
```

הקשר בין המחלקות:



כאמור, לפתרון המשימה, שמרנו במחלקה Union מערך עם הפניות לעצמים מטיפוס Team. המחלקה UnionNode הבאה שומרת אינפורמציה זהה לזו שנשמרה במחלקה Union אך במקום המערך, יש בה רשימה מקושרת (שרשרת חוליות) שהערך של כל חוליה בה הוא הפניה לעצם מטיפוס Team. כלומר, בתכונה value של כל חוליה נשמרת הפניה לעצם מטיפוס Team, ובתכונה next נשמרת הפניה לחוליה הבאה.



שימו לב, השינוי הוא במחלקה Union בלבד ואינו משפיע על המחלקה הראשית ועל המחלקה Team.

```

import java.util.Scanner;
public class UnionNode
{
    private Scanner input = new Scanner(System.in);
    private Node <Team> union=null;

    public UnionNode () // פעולה בונה
    {
        Node <Team> t ;
        System.out.println ("enter first team name");
        String name = input.next();
        this.union = new Node <Team> (new Team (name));
        Node <Team> pos=this.union ;
        System.out.println ("enter team name");
        name = input.next();
        while (!name.equals("aaa"))
        {
            t = new Node <Team> (new Team (name));
            pos.setNext(t);
            pos=pos.getNext();
            System.out.println ("enter team name");
            name = input.next();
        }
    }
}

```

תכונה אחת union = משתנה שתהיה בו הפניה לחוליה גנרית שבתכונה value שלה תהיה הפניה לעצם מטיפוס Team. כרגע, ב- union אין כלום

t = משתנה שתהיה בו הפניה לחוליה גנרית שבתכונה value שלה תהיה הפניה לעצם מטיפוס Team

בניית עצם חדש מטיפוס Team, בניית חוליה שבתכונה value שלה יש הפניה לעצם החדש, ושמירת ההפניה לחוליה החדשה כחוליה ראשונה ברשימה המקושרת

pos = משתנה מטיפוס חוליה גנרית שבתכונה value שלה תהיה הפניה לעצם מטיפוס Team. כרגע ב- pos יש הפניה זהה להפניה שב- this.union

בניית עצם חדש מטיפוס Team, בניית חוליה שבתכונה value שלה יש הפניה לעצם החדש, והשמת הפניה לחוליה החדשה

בניית חוליה נוספת ברשימה: העתקת ההפניה שב t לתכונה next של החוליה שההפניה אליה היא ב pos

שינוי הערך שבמשתנה pos כך שתהיה בו כעת הפניה לחוליה החדשה

המשך המחלקה UnionNode בעמוד הבא

```

public void printBiggest ()
{
    double big= this.union.getValue().getRatio();
    Node<Team> pos=this.union;
    while (pos!=null)
    {
        if (pos.getValue().getRatio()>big)
            big= pos.getValue().getRatio();
        pos=pos.getNext();
    }
    System.out.println ("biggest ratio:");
    pos=this.union;
    while (pos!=null)
    {
        if (pos.getValue().getRatio()<=big)
            System.out.println (pos.getValue().getName());
        pos=pos.getNext();
    }
}

public void underAverage ()
{
    double sum=0;
    int count=0;
    Node<Team> pos=this.union;
    while (pos!=null)
    {
        sum=sum+ pos.getValue().getRatio();
        count++;
        pos=pos.getNext();
    }
    double ave = sum/count;
    System.out.println ("under average:");
    pos=this.union;
    while (pos!=null)
    {
        if (pos.getValue().getRatio()<ave)
            System.out.println (pos.getValue().getName());
        pos=pos.getNext();
    }
}
}

```

- משמעות הביטוי:

pos.getValue().getRatio()

היא הפעלת הפעולה `getRatio` על

העצם המוחזר מהפעולה

`getValue` כאשר היא מופעלת על

החוליה (העצם) שההפניה אליה

נמצאת במשתנה _____.

- בפעולה `printBiggest` מבצעים

שני מעברים על הרשימה. במעבר

הראשון מוצאים את היחס הגדול

ביותר בין מספר הניצחונות לבין

מספר המשחקים, ומציבים אותו

במשתנה _____.

ובמעבר השני _____

- גם בפעולה `underAverage`

מבצעים שני מעברים על הרשימה.

במעבר הראשון

ובמעבר השני _____

- בפעולות `printBiggest` ו-

`underAverage`, התפקיד של `pos` הוא _____

- בפרויקט בו הקלדתם את המחלקה הראשית ואת המחלקה `Team`, פתחו מחלקה חדשה והעתיקו אליה את המחלקה `UnionNode` (שמחכה לכם באתר הקורס).

- כדי שהמחלקה הראשית תתייחס למחלקה UnionNode, הכנסו למחלקה הראשית ושנו בה את השורה
Union u=new Union();
כך שבמקום Union יהיה רשום UnionNode.
- מכיוון שהמחלקה UnionNode משתמשת בעצם מטיפוס Node, היא צריכה להכיר את המחלקה Node, לשם כך, עימדו עם העכבר על השם של הפרויקט הנוכחי, ולחצו על המקש הימני. בחלון שנפתח, בחרו ב-
Properties, בצד השמאלי של החלון שנפתח כעת לחצו על Java Build Path. באותו חלון, בלשוניות
למעלה לחצו על Libraries, ובצד הימני של אותו חלון לחצו על Add Class Folder. כעת נפתח חלון
שמופיעים בו השמות של כל הפרויקטים שכתבתם עד כה, לחצו על הריבוע שמשמאל לפרויקט בו הקלדתם
את המחלקה Node (שימו לב שמופיע הסימן V). אשרו (OK) ואשרו גם את המסך הבא (שוב OK).
- הריצו כעת את המחלקה הראשית, כשתתבקשו הקישו נתונים (שמות של קבוצות, ניצחונות, הפסדים)
וודאו כי התכנית מבצעת את הנדרש.

רשימה מקושרת (שרשרת חוליות) היא **מבנה נתונים דינמי**. השימוש בו שיחרר אותנו מהצורך
להניח מראש מהו המספר מקסימלי של קבוצות. באמצעות שרשרת חוליות שמרנו אוסף של נתונים.
כך שכל נתון נשמר בתוך חוליה.

משימה 11 – חלק א'

במשימה 26 מעבדה 12 כתבנו תכנה עבור "קניין הזהב". בחלק ב' פתרנו את המשימה מבלי להניח מראש מהו
המספר המקסימלי של הנתונים (מספר החנויות). בחלק זה יצרנו דינמיות במספר הנתונים בכך שבכל פעם
שהמערך בו שמרנו את ההפניות לעצמים מטיפוס Shop היה מלא, עברנו לעבוד עם מערך גדול יותר. שנו כעת
את הפתרון. הפעם שמירת ההפניות לעצמים מטיפוס Shop תהיה באמצעות רשימה מקושרת (במקום מערך).
תזכורות:

1. צריך לשנות רק את המחלקה "קניין".
 2. אם אתם מחליטים לשנות את השם של המחלקה בה מוגדר הקניין (כמו שעשינו במשימה האחרונה), דאגו
לשנות גם את יצירת העצם מטיפוס קניין (במחלקה הראשית) כך שיווצר עצם מטיפוס המחלקה קניין
החדשה (ולא המחלקה הישנה בה משתמשים במערך של חנויות).
 3. דאגו שהפרויקט בו מוגדרות המחלקות יכיר את המחלקה Node.
- הריצו ובדקו כי התכנה מבצעת את הנדרש.

משימה 11 – חלק ב'

בסעיף הקודם קלטנו עבור כל חנות את שמה ואת מספר המוצרים הנמכרים בה. עבור כל מוצר קלטנו את שמו
ואת מחירו. שנו כעת את התכנה כך שבמקום לקלוט את מספר המוצרים בכל חנות, התכנה תקלוט עבור כל
חנות את שמה ולאחריו עבור כל המוצרים: שם ומחיר. קליטת הנתונים עבור כל חנות תסתיים עם קליטת מוצר
בשם "sof" (כמובן שאין לקלוט מחיר עבור מוצר זה).

הנחיה: במקום במערך של מוצרים, יש להשתמש ברשימה מקושרת של מוצרים.
שימו לב, הפעם הפעולה הבונה במחלקה מוצר מוכרחה לקבל את השם של המוצר. למה?
הערות: 1. הקפידו שלא תהיה לחנות תכונה של מספר המוצרים. 2. אין צורך למיין את המוצרים!!

- הריצו ובדקו כי התכנה מבצעת את הנדרש.

משימה 12

בבית הספר "שומרים על הירוק" שומרים עבור כל תלמיד את שמו, את שם משפחתו ואת הכיתה בה הוא לומד (מספר 9-12). לכל כיתה שומרים את התלמידים שבה.

בית הספר מקפיד על שמירת הסביבה ולכן צריך לבנות תכנה שתשלח כל הודעה רק פעם אחת לכל משפחה. כתבו תכנה שתקלוט קוד:

1 – הוספת תלמיד חדש לבית הספר. התכנה תקלוט את שמו, את שם משפחתו ואת הכיתה אליה הוא משובץ. הנחיה: צריך לבדוק אם יש כבר תלמיד מהמשפחה. אם יש, מוסיפים את התלמיד לרשימה המשפחה. אם אין, בונים עצם חדש מטיפוס משפחה.

2 – שליחת הודעה לכל המשפחות. התכנה תדפיס את כל שמות המשפחה. שמות המשפחה יודפסו בסדר אלפבית עולה.

3 – הדפסת שמות תלמידים ממשפחה. התכנה תקלוט שם משפחה ותדפיס את השמות של התלמידים במשפחה.

4 – הדפסת שמות תלמידים בכיתה. התכנה תקלוט מספר כיתה (9-12) ותדפיס עבור כל אחד מתלמידי הכיתה את שמו ואת שם משפחתו.

שמות תלמידי הכיתה יודפסו בסדר אלפבית עולה של שמות המשפחה.

0 – סיום.

הנחיות:

1. **השמות של התלמידים משמשים כמזהים** (אין שני תלמידים או יותר בעלי אותו שם ואותו שם משפחה ואי אפשר לשנות את השמות של התלמידים).

2. התכנה תכיל 5 מחלקות:

Student מחלקה שתכיל אינפורמציה עבור תלמיד אחד (שם התלמיד, שם משפחתו, ומספר הכיתה),

Family מחלקה שתכיל את שם המשפחה ורשימה של תלמידים מאותה משפחה (רשימה של Student),

Grade מחלקה שתכיל את מספר הכיתה ורשימה של תלמידי הכיתה,

School מחלקה שתכיל רשימה של משפחות ורשימה של כיתות,

מחלקה ראשית.

3. הפעולה הבונה של School תבנה רשימה ריקה של משפחות, ורשימה של כיתות שבה לכל כיתה יש מספר ורשימה ריקה של תלמידים.

4. **כל עצם מטיפוס Student ישולב בשתי רשימות. רשימה של המשפחה שלו ורשימה של הכיתה שלו.**

5. יש לבנות את רשימת המשפחות במחלקה School ואת רשימת התלמידים במחלקה Grade כך שתהיינה ממוינת בסדר עולה של שמות המשפחה.

בעמוד הבא מופיעה דוגמה להרצת התוכנה.

```

Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
1
Enter Student name
Tal
Enter Student Family name
Lev
Enter Student grade
10
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
1
Enter Student name
Gil
Enter Student Family name
Ziv
Enter Student grade
11
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
1
Enter Student name
Tom
Enter Student Family name
Shavit
Enter Student grade
10
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
1
Enter Student name
Or
Enter Student Family name
Ziv
Enter Student grade
10
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
2
The families are:
Lev
Shavit
Ziv

```

```

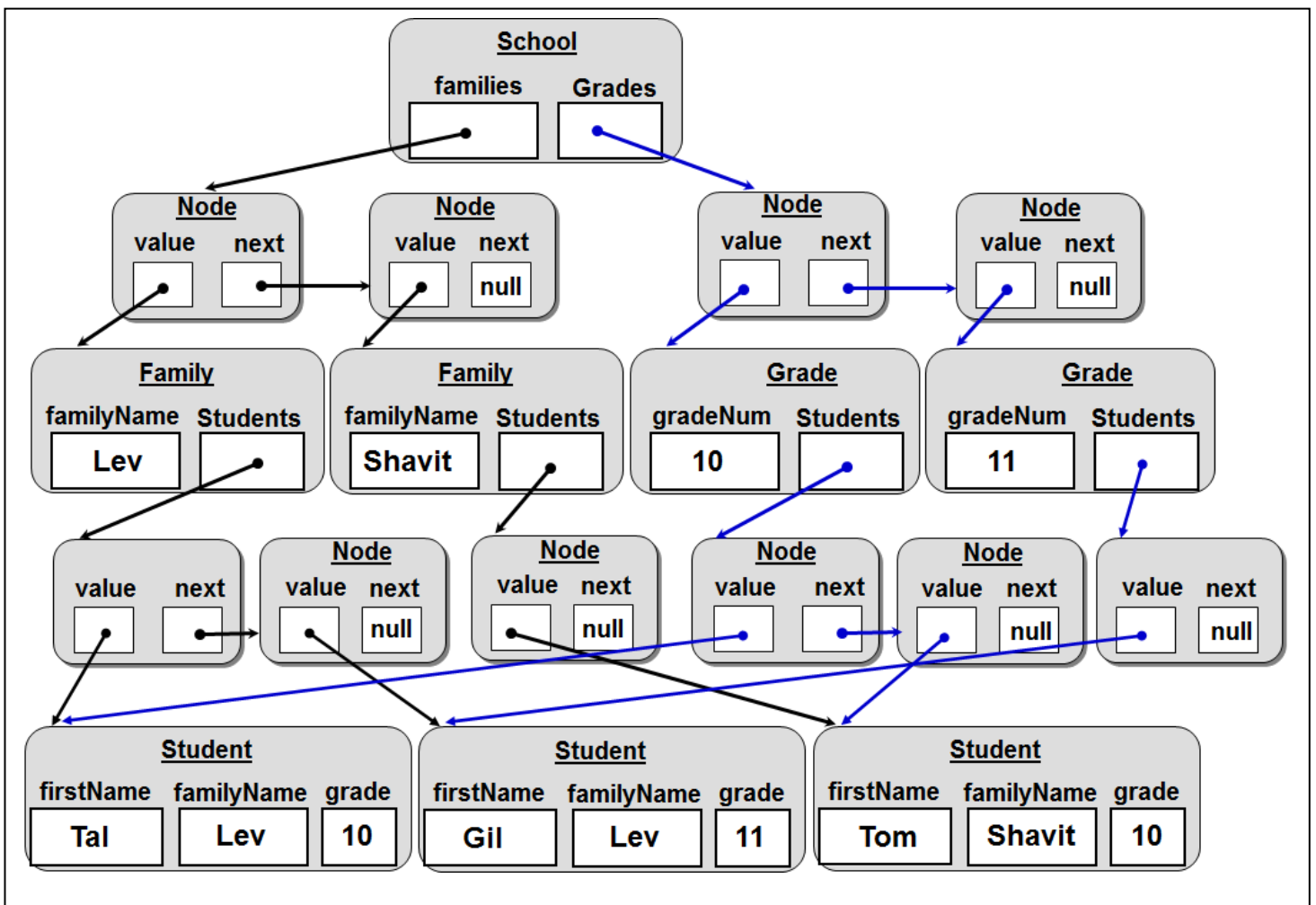
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
3
Enter Family name
Lev
Children in Family Lev:
Tal
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
3
Enter Family name
Ziv
Children in Family Ziv:
Gil
Or
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
4
Enter Grade 9-12
9
No students in grade 9
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
4
Enter Grade 9-12
10
Students in grade 10:
Lev Tal
Shavit Tom
Ziv Or
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
4
Enter Grade 9-12
11
Students in grade 11:
Ziv Gil
Enter:
1 -- Add new student
2 -- Print all families name
3 -- Print students from specific family
4 -- Print students from specific grade
0 -- Finish
0
Thank you

```

עצם יכול להשתתף ברשימות שונות!

למשל, באיור הבא,

- העצם Student שהערך של התכונה firstName שלו הוא Tal, משתתף גם ברשימה של משפחת _____ וגם ברשימה של כיתה מספר _____.
- העצם Student שהערך של התכונה firstName שלו הוא Tom, משתתף גם ברשימה של משפחת _____ וגם ברשימה של כיתה מספר _____.



משימה 13

בספריה העירונית שומרים לכל ספר את המספר הידורי שלו, את שמו ואת שם המחבר (סופר / משורר) שכתב אותו.

המספר הידורי של הספר הראשון שהוכנס לספריה הוא 1, השני 2 וכן הלאה.

כתבו תכנה לניהול הספריה. התכנה תאפשר למשתמש לבצע את הפעולות הבאות:

1. להוסיף ספר לספריה. המשתמש יתבקש לספק את השמות של הספר ושל המחבר.

במידה והספר כבר נמצא בספריה, תודפס הודעה מתאימה ולא תבצע ההוספה.

2. לקבל רשימה של כל הספרים שבספריה. לכל ספר את המספר שלו, שמו ושם המחבר שלו. הספרים יודפסו בסדר עולה של מספרם.

3. לקבל רשימה של שמות המחברים שהספריה מחזיקה ספרים שכתבו. שמות המחברים יודפסו בסדר אלפבית עולה.

4. לקבל רשימה של ספרים למחבר. המשתמש יספק שם של מחבר והתכנה תדפיס את השמות של כל

הספרים שכתב ונמצאים ברשות הספריה. שמות הספרים יודפסו בסדר אלפבית עולה.

במידה ואין ברשות הספריה אף ספר שהמחבר המבוקש כתב, תודפס הודעה מתאימה.

5. לקבל מספר סידורי של ספר. המשתמש יספק את שם הספר ואת שם המחבר והתכנה תדפיס את המספר שלו. במידה והספר אינו נמצא בספריה, תודפס הודעה מתאימה.

הנחה: הקלט תקין

הנחיות:

1. השמות של המחברים משמשים כמזהים.

2. מחבר לא כתב שני ספרים או יותר בעלי אותו שם. אבל יתכן שמחברים שונים כתבו ספר בעל אותו שם.

3. התכנה תכיל 4 מחלקות:

Book מחלקה שתכיל אינפורמציה עבור ספר אחד (מספר הספר, שם ספר, שם המחבר),

Author מחלקה שתכיל אינפורמציה עבור מחבר אחד: שם המחבר ורשימה של הספרים שחיבר ושנמצאים ברשות הספריה.

Library מחלקת ספריה שתכיל שתי רשימות: רשימה של הספרים שברשות הספריה, ורשימה של מחברים שספרים שכתבו נמצאים ברשות הספריה.

מחלקה ראשית.

4. כל עצם מטיפוס **Book** ישולב בשתי רשימות. רשימה של ספרים ורשימה של המחבר שלו.

בעמוד הבא מופיעה דוגמה להרצת התוכנה.

```

Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
1
Enter book name   TheIsland
Enter author name Hislop
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
1
Enter book name   SonyaDance
Enter author name Hislop
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
1
Enter book name   Vicktoria
Enter author name Michael
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
1
Enter book name   TheHelp
Enter author name Stockett
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
1
Enter book name   Aida
Enter author name Michael
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
2
Book Number  Book Name      Author Name
1             TheIsland     Hislop
2             SonyaDance    Hislop
3             Vicktoria     Michael
4             TheHelp      Stockett
5             Aida         Michael

```

```

Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
3
Authore are:   Hislop Michael Stockett
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
4
Enter author name Michael
Books are:     Vicktoria Aida
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
4
Enter author name Shalev
The library does not has any book of
this author
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
5
Enter book name   Roman
Enter author name Michael
The Book did not found
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
5
Enter book name   Aida
Enter author name Michael
Book number is 5
Enter:
1 -- Add book to library
2 -- Print all books
3 -- Print all authors
4 -- Print books of author
5 -- Get Number of Book
6 -- Exit the program
6
Bye Bye

```