

תכנות מתקדם – 150024 תרגיל בית מספר 6 ירושה

1. נגדיר רשימה מעגלית **RoundList**, כרשימה לינארית שבה החוליה האחרונה מצביעה על החוליה הראשונה.
על רשימה מעגלית ניתן לבצע את כל הפעולות המוגדרות עבור רשימה לינארית (מתודות),
ובנוסף ניתן לבצע את הפעולות הבאות:

- הוספת איבר בסוף הרשימה **addToEnd(int val)**. הפונקציה מקבלת כפרמטר מספר שלם **val**, ומוסיפה חוליה בסוף הרשימה עם הערך **val**.
- חיפוש ברשימה **search(int n)**. הפונקציה מקבלת כפרמטר מספר שלם לא שלילי, **n**. הפונקציה מחזירה את ערכו של האיבר הנמצא במקום **n** ברשימה המעגלית. (שים לב: **n** יכול להיות מספר גדול יותר ממספר האיברים הקיימים ברשימה. מאחר והרשימה מעגלית יש להמשיך למנות את האיברים בשנית עד לאינדקס **n**. מספור הרשימה מתחיל מאינדקס 0). במידה והרשימה ריקה, הפונקציה מחזירה -1.

הגדר את המחלקה **RoundList** כמחלקה היורשת מ-**List** (המחלקה **List** המגדירה רשימה לינארית, נמצאת במודל, ביחידת מבוא, תחת התיקיה מבנה נתונים קוד). ממש את כל הפעולות הנדרשות. (כלומר יש לוודא שכל הפעולות שכרגע ניתנות לביצוע על **List** יפעלו נכון על **RoundList** ובנוסף לממש את המתודות החדשות המצוינות לעיל).

הנחיות נוספות:

- לצורך התרגיל, יש למחוק את השיטה **search** המופיעה במחלקה **List** שהורדתם מהמודל.
- אין להוסיף תכונות פרטיות חדשות למחלקה **List** – השדה היחיד במחלקה יהיה מצביע לראש הרשימה כפי שמוגדר במחלקה **List** הבסיסית.
- עליך להכריע אילו שיטות במחלקה **List** יש לדרוס במחלקה **RoundList** היורשת, ואילו שיטות אין צורך לדרוס.
- כפי שנאמר לעיל נתבקשת להוסיף שתי מתודות נוספות למחלקת **RoundList** שלא היו במחלקה **List**.

השתמש בתכנית הראשית הבאה בדיוק כמו שהיא, הבוחנת את נכונות המחלקה שהגדרת (על מנת שהמקרי בדיקה ייצאו נכון)

```
#include "RoundList.h"
#include <iostream>
using namespace std;
```

```
enum CHOICES{
    EXIT, ADD, ADD_TO_END, REMOVE_FIRST, SEARCH, CLEAR, EMPTY
};
```

```
int main(){
    RoundList ls1;
    int choice;
    cout << "Enter your choice: ";
    cin >> choice;
    while(choice != EXIT)
    {
        int num;
        switch(choice){
            case ADD :    cout << "Enter 5 numbers: ";
                          for(int i=0; i < 5; i++)
                          {
                              cin >> num;
                              ls1.add(num);
                          }
                          break;
            case ADD_TO_END :cout << "Enter 5 numbers: ";
                          for(int i=0; i < 5; i++)
                          {
                              cin >> num;
                              ls1.addToEnd(num);
                          }
                          break;
            case REMOVE_FIRST : ls1.removeFirst();
                          break;
            case SEARCH: cout << "Enter a number: ";
                          cin >> num;
                          cout << ls1.search(num)<<endl
                          break;
            case CLEAR: ls1.clear();
                          break;
            case EMPTY: if(ls1.isEmpty())
                          cout << "Empty"<<endl;
                          else
                          cout << "Not empty" << endl;
                          break;
            default: cout<< "ERROR!"<<endl;
        }
        cout << "Enter your choice: ";
        cin >> choice;
    }
    return 0;
}
```

2. בתרגיל זה יש לכתוב תכנית לניהול מערך המלגות במוסד להשכלה גבוהה.

במוסד להשכלה גבוהה לומדים סטודנטים לתואר ראשון, שני ושלישי.

על מנת לייצג את הסטודנטים, עליך לבנות היררכיה של מחלקות, באופן הבא:

א. מחלקה בסיסית בשם **Student** לייצוג סטודנט.

השדות במחלקה יהיו: מספר זהות, שם פרטי, שם משפחה, מספר הקורסים שהוא לומד.

הוסף/י למחלקה את הפונקציות הבאות:

- constructor לאיתחול כל השדות.
- void input() פונקציה המקבלת כקלט את נתוני הסטודנט. את נתוני הסטודנט יש להתאים לשדות שיש לו, ולקלוט את כל השדות. בראש תהיה שורת כותרת הבאה:

```
"enter id, first name, last name, number of courses\n";
```

ולאחר מכן יש לקלוט את כל השדות.
- void print() להדפיס נתוני הסטודנט. את נתוני הסטודנט יש להתאים לשדות שיש לו, כאשר יש להדפיס כותרת לשם השדה ולאחר מכן את הנתונים של השדה, כל שדה בשורה נפרדת. ניתן להיעזר בדוגמה למטה כדי לדעת כיצד יש להדפיס כל אחד מן הסטודנטים השונים.

ב. מחלקה **BA היורשת מ-Student** לייצוג סטודנט מתואר ראשון,

שמכיל בנוסף לפרטים האישיים גם מצביע למערך של ציונים ואת גודלו.

וכן מתודה `bool milga()`, פונקציה בוליאנית שבודקת האם הסטודנט זכאי למלגה. הקריטריונים

לחישוב המילגה יובאו בהמשך. בנוסף יש לדאוג שהמתודות `input` ו `print` שנכתבו במחלקה

`Student` יותאמו להדפיס את נתוני המחלקה. (נא להתאים לפי דוגמת הפלט בסוף המסמך)

ג. מחלקה **MA היורשת מ-BA** לייצוג סטודנט מתואר שני,

שמכיל בנוסף לפרטים הקיימים שדה בוליאני שמציין אם הסטודנט משתתף במחקר או לא. וכן

מתודה `bool milga()`, פונקציה בוליאנית שבודקת האם הסטודנט זכאי למלגה. הקריטריונים

לחישוב המילגה יובאו בהמשך. בנוסף יש לדאוג שהמתודות `input` ו `print` שנכתבו במחלקה

`Student` יותאמו להדפיס את נתוני המחלקה. (נא להתאים לפי דוגמת הפלט בסוף המסמך)

ד. מחלקה **PHD היורשת מ-Student** לייצוג סטודנט מתואר שלישי,

שמכיל בנוסף לפרטים האישיים גם את מספר השעות השבועיות שהוא עוסק במחקר. וכן מתודה

`bool milga()`, פונקציה בוליאנית שבודקת האם הסטודנט זכאי למלגה. הקריטריונים לחישוב

המילגה יובאו בהמשך. בנוסף יש לדאוג שהמתודות `input` ו `print` שנכתבו במחלקה `Student`

יותאמו להדפיס את נתוני המחלקה. (נא להתאים לפי דוגמת הפלט בסוף המסמך)

ה. הנהלת המוסד קבעה קריטריונים לקבלת מלגה לכל אחד ממסלולי הלימוד:

- **תואר ראשון:** לומד לפחות 10 קורסים עם ממוצע ציונים גבוה מ- 95 .
 - **תואר שני:** לומד לפחות 7 קורסים עם ממוצע מעל 90 , ובנוסף עוסק במחקר.
 - **תואר שלישי:** לומד לפחות 2 קורסים ומשקיע מעל 25 שעות שבועיות במחקר.
- ו. בכל אחת מן המחלקות שהגדרת יש לממש את כל הפונקציות הרלוונטיות, (כמו כן גם לממש פונקציות גישה (getters and setter)

השתמש בתכנית הראשית הבאה, הבוחנת את נכונות המחלקה שהגדרת :

```
#include "MA.h"
#include "PHD.h"

int main()
{
    BA ba[2];
    cout << "Enter details for 2 BA students\n";
    for (int i = 0; i < 2; i++)
        ba[i].input();
    cout << "BA students entitled for milga are: \n";
    for (int i = 0; i < 2; i++)
    {
        if (ba[i].milga())
            ba[i].print();
        //else cout << " Student not entitled for milga\n";
    }

    cout << "Enter details for 2 MA students\n";
    MA ma[2];
    for (int i = 0; i < 2; i++)
        ma[i].input();
    cout << "MA students entitled for milga are: \n";
    for (int i = 0; i < 2; i++)
    {
        if (ma[i].milga())
            ma[i].print();
    }

    cout << "Enter details for 3 PHD students\n";
    PHD phd[3];
    for (int i = 0; i < 3; i++)
        phd[i].input();
    cout << "12PHD students entitled for milga are: \n";
    for (int i = 0; i < 3; i++)
    {
        if (phd[i].milga())
            phd[i].print();
    }

    return 0;
}
```

כדי לקבל את הפלט המופיע בתיבת ההגשה,

הוסיפו את 7 הסטודנטים הבאים

BA

(2 סטודנטים)

תז	שם משפחה	שם פרטי	מספר קורסים	ציונים בקורסים (שמאל לימין)
123	avraham	abe	3	90, 100, 90
234	avrahamson	yitzhak	10	100, 100, 90, 100, 90, 100, 90, 100, 90, 100

MA

(2 סטודנטים)

תז	שם משפחה	שם פרטי	מספר קורסים	ציונים (משמאל לימין)	מחקר?
345	jacobson	yaacov	7	90, 100, 90, 90, 100, 90, 100, 90	0
456	emanu	sara	7	90, 100, 90, 90, 100, 90, 100, 90	1

PHD

(3 סטודנטים)

תז	שם משפחה	שם פרטי	מספר קורסים	שעות מחקר
567	imanu	rivka	1	30
678	jacobs	rachel	2	20
789	jacobs	leah	2	30

עבור הסטודנטים הנתונים הפלט המדויק חייב להיות :

Enter details for 2 BA students

enter id, first name, last name, number of courses

and enter a list of student grades

enter id, first name, last name, number of courses

and enter a list of student grades

BA students entitled for milga are:

ID: 234

Name: yitzhak avramson

Grades: 100 100 90 100 90 100 90 100 90 100

Enter details for 2 MA students

enter id, first name, last name, number of courses

and enter a list of student grades

enter 1 if the student does research and 0 if not

enter id, first name, last name, number of courses

and enter a list of student grades

enter 1 if the student does research and 0 if not

MA students entitled for milga are:

ID: 456

Name: sara emanu

Grades: 90 100 90 100 90 100 90

research: 1

Enter details for 3 PHD students

enter id, first name, last name, number of courses

enter number of research hours:

enter id, first name, last name, number of courses

enter number of research hours:

enter id, first name, last name, number of courses

enter number of research hours:

PHD students entitled for milga are:

ID: 789

Name: leah jacobs

number of research Hours: 30