**תרגיל 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 שהגדרנו עבור רשימה לינארית, 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 בחוברת הקורס בעמוד 17). ממש את כל הפעולות הנדרשות.

הנחיות נוספות:

* לצורך התרגיל, יש למחוק את השיטה search המופיעה במחלקה List שבחוברת הקורס.
* אין להוסיף תכונות פרטיות חדשות למחלקה List – השדה היחיד במחלקה יהיה מצביע לראש הרשימה כפי שמוגדר במחלקה List הבסיסית.
* עלייך להכריע אילו שיטות במחלקהList יש לדרוס במחלקה RoundList היורשת, ואילו שיטות אין צורך לדרוס.

השתמש בתכנית הראשית הבאה, הבוחנת את נכונות המחלקה שהגדרת:

#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;

}

1. בשאלה זו נגדיר מחלקה בסיסית עבור המצולעים סגורים, ונגדיר מחלקות יורשות לייצוג משולש ומלבן.
   1. הגדר מחלקה Polygon עבור מצולע סגור.   
      המחלקה תכלול את השדות הבאים:
      * numSides - מספר הצלעות במצולע (int)
      * sides - מערך של שלמים באורך מספר הצלעות, המכיל את אורכי כל אחת מהצלעות (int \*)

וכן את הפעולות הבאות:

* בנאים:
  + - בנאיהמקבל כפרמטר ערך אחד עבור מספר הצלעות. הפונקציה תבנה את המערך המתאים עבור אורכי הצלעות.   
      במידה והמצולע אינו בעל 3 או 4 צלעות - הפונקציה תדפיס enter sides for polygon: ,ותקלוט מהמשתמש ערכים עבור אורכי הצלעות. (ניתן להניח שהנתונים תקינים. אין צורך לבדוק היתכנות מתמטית לבניית המצולע בעל אורכי הצלעות שנקלטו)
    - בנאי העתקה.
  + פונקציות אחזור (get) לפי הצורך. (אין צורך להגדיר שיטות set, מאחר ולא ניתן לשנות את אחד מהשדות בלי לשנות את השדה השני)
  + פונקציה המחזירה את היקף המצולע.
  + אופרטור ==. הפונקציה תחזיר אמת במידה ושני המצולעים שווים, אחרת הפונקציה תחזיר שקר.

שני מצולעים יוגדרו כשווים, כאשר:

* + - מספר הצלעות של שני המצולעים שווה.
    - סכום הצלעות של שני המצולעים שווה.

לדוגמא, המצולע A(3,4,5) זהה למצולע B(4,4,4) אך אינו זהה למצולע C(3,4,4) ואינו זהה למצולע D(1,2,4,5)

* 1. הגדר שתי מחלקות יורשות ממחלקת המצולע הסגור:
     + מחלקה עבור משולש (Triangle) – משולש הינו מצולע סגור בעל 3 צלעות.  
       המחלקה כוללת בין השאר:
     + בנאי המקבל אך ורק שלושה ערכים כפרמטרים, המייצגים את אורכי הצלעות של המשולש.
     + מחלקה עבור מלבן (Rectangle) – מלבן הינו מצולע סגור, בעל 2 זוגות של צלעות מקבילות זהות.

המחלקה כוללת בין השאר:

* + - בנאי המקבל אך ורק שני ערכים כפרמטרים, המייצגים את אורכי 2 צלעות סמוכות של המלבן. הבנאי יציב ערכים עבור כל ארבעת הצלעות במערך אורכי הצלעות.

שים לב! את יצירת מערך הצלעות יש לכתוב פעם אחת בלבד, בבנאי של מחלקת מצולע.

בנוסף, עליך לשקול האם יש צורך להגדיר את השיטות הבאות. במידה וכן, הגדר אותם במחלקה המתאימה:

* + - בנאי העתקה
    - פונקציות הצבה (set)
    - פונקציות אחזור (get)
    - שיטות הדורסות שיטות ממחלקת Polygon
    - כל פונקציה נוספת שתמצא לנכון (אם קטע תכנית מסוים נכתב מספר פעמים יש לכתבו כפונקציה).
  1. כתוב תכנית ראשית אשר תקלוט את נתוניהם של שני מצולעים. על התכנית לבדוק אם שני המצולעים שווים (כהגדרתם בסעיף א – מספר הצלעות שווה, וההיקף שווה), ולהדפיס הודעה מתאימה.

הנחיות נוספות:

* התכנית תציג הודעה למשתמש enter number of sides for polygon 1: ותקלוט את מספר הצלעות במצולע הראשון.
* התוכנית תציג הודעה למשתמש enter sides for polygon: ותקלוט את ערכי הצלעות של המצולע הראשון.   
  במידה ומספר הצלעות הוא 3 או 4, הדפסת ההודעה והקלט הינם מהתכנית הראשית, אחרת ההדפסה והקלט יתבצעו ע"י הבנאי של מחלקת מצולע.

ניתן להניח שכל מצולע בעל 4 צלעות הוא מלבן.

* לאחר מכן, התכנית תציג למשתמש את ההודעה enter number of sides for polygon 2: ותקלוט את מספר הצלעות של המצולע השני.
* רק במידה ומספר הצלעות במצולע הראשון שווה למספר הצלעות במצולע השני, התוכנית תציג למשתמש הודעה enter sides for polygon: ותקלוט את ערכי הצלעות של המצולע השני.
* אם שני המצולעים שווים, התכנית תדפיס: equal, אחרת התכנית תדפיס: not equal.

הערה: ניתן להניח שהקלט תקין, כלומר – הערכים הינם שלמים וחיוביים, ומספר אורכי הצלעות הנקלט תואם למספר הצלעות (פרט למקרה של מלבן – בו מספר הצלעות הוא 2, אך מספר אורכי הצלעות הנקלט הוא 2 בלבד)

דוגמת הרצה 1 :

enter number of sides for polygon 1:

3

enter sides for polygon:

20 10 15

enter number of sides for polygon 2:

3

enter sides for polygon:

11 14 20

equal

דוגמת הרצה 2 :

enter number of sides for polygon 1:

3

enter sides for polygon:

20 10 15

enter number of sides for polygon 2:

3

enter sides for polygon:

10 14 20

not equal

דוגמת הרצה 3 :

enter number of sides for polygon 1:

4

enter sides for polygon:

20 15

enter number of sides for polygon 2:

4

enter sides for polygon:

15 10

not equal

דוגמת הרצה 4 :

enter number of sides for polygon 1:

6

enter sides for polygon:

20 10 5 5 5 15

enter number of sides for polygon 2:

6

enter sides for polygon:

10 10 10 10 5 15

equal

דוגמת הרצה 5 :

enter number of sides for polygon 1:

6

enter sides for polygon:

20 10 5 5 5 15

enter number of sides for polygon 2:

2

not equal