

סדנא ב-C++

תרגיל 6

ירושא

תזכורת אין להשתמש ב- try ו- catch בבדיקה האוטומטית!

שאלה 1.

- RoundList היא רשימה לינארית מעגלית, כלומר רשימה שבה האיבר האחרון מצביע לראשון. הפעולות שניתן לעשות על רשימה זו זהות לפעולות של רשימה רגילה בתוספת:
- addToEnd(int val) אפשרות להוסיף איבר לסוף הרשימה.
 - search(int n) פונקציה המחזירה את ערך האיבר במקום ה-n-י (n מתחיל מ-0).
 - שימו לב ש-n יכול להיות מספר גדול יותר ממספר האיברים הקיימים מרשימה. במידה שהרשימה ריקה הפונקציה מחזירה 0.
- יש להגדיר את המחלקה RoundList כמחלקה היורשת מ-List (הנמצאת בחוברת) ולממש את כל הפעולות הנדרשות. שימו לב:
- אין להוסיף תכונות פרטיות חדשות למחלקה!
 - יש לחשוב על איזה שיטות במחלקה List צריכים לדרוס ב-RoundList ואיזה שיטות לא צריכים לדרוס.
 - עבור תרגיל זה יש למחוק את השיטה search הנמצאת במחלקה List בחוברת הקורס.
 - אפשר למצוא את הקבצים List.h ו-List.cpp (בלי השיטה search) במודל.
- השתמשו בתוכנית הראשית הבאה כדי לבדוק את המחלקה שכתבתם.

סדנא ב-C++

```

#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;
        }
    }
}

```

סדנא ב-++C

```

else
    cout << "Not empty" << endl;
break;

default: cout<< "ERROR!"<<endl;
}
cout << "Enter your choice: ";
cin >> choice;
}
return 0;
}

```

שאלה 2

משפחת המצולעים הסגורים מוגדרת כמחלקה בעלת המשתנים הבאים:

numSides int – מספר הצלעות.

*sides long – מערך בגודל numSides המכיל את אורכי צלעות המצולע.

סעיף א.

הגדירו מחלקה בשם Polygon המכילה את התכונות הפרטיות המתוארות למעלה.

• הוסיפו למחלקה שני בנאים-

○ הבנאי הראשון יקבל ערך אחד עבור numSides ויבנה את המערך sides.

ערכי הצלעות יתקבלו מהקלט (במידה שלא מדובר על מלבן או משולש).

○ הבנאי השני יהיה בנאי העתקה.

• בנוסף כתבו שיטות get לפי הצורך. (שימו לב: בגלל שמדובר על מחלקה המייצגת מערך –

אין צורך להגדיר שיטות set כי אי-אפשר לשנות את מספר הצלעות בלי לשנות את גודל

המערך המכיל את הצלעות ולהפך).

• הוסיפו פונקציה לחישוב היקף המצולע.

• הגדירו את האופרטור ==. הפונקציה תחזיר true במידה ושני המצולעים הם זהים ואחרת

תחזיר false. שימו לב!

שני מצולעים יוגדרו כזהים אם:

○ מספר הצלעות של המצולעים שווה.

○ סכום הצלעות של 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)

סעיף ב

סדנא ב-++C

ממחלקה זו יש לגזור בירושה את המחלקות הבאות: מלבנים (Rectangle) ומשולשים (Triangle).
מלבן מוגדר ע"י אורכי שתי צלעות סמוכות ו**משולש** ע"י אורכי שלוש הצלעות.

הגדירו את 2 המחלקות Rectangle ו-Triangle

- הוסיפו לכל מחלקה בנאים.

- **הבנאי הראשון** של מחלקת המלבנים יקבל אך ורק שני ערכים עבור שתי צלעות סמוכות של המלבן ויציב את ערכי כל ארבעת הצלעות במערך sides.
- **הבנאי הראשון של** מחלקת המשולשים יקבל אך ורק שלושה ערכים עבור שלושת צלעות.
- **הבנאי השני של** שני המחלקות יהיה בנאי העתקה.

האם יש צורך להגדיר במחלקות היורשות בנאי העתקה? שיטות set? שיטות get? שיטות הדורסות את השיטות שנכתבו בסעיף א עבור המחלקה Polygon? עבור כל שאלה עליה עניתם כן – הגדירו את הבנאי/שיטה החדשים במחלקות המתאימות.

שימו לב! את יצירת מערך הצלעות עליכם לכתוב פעם אחת בלבד.
הערה: אם קטע תוכנית מסוים נכתב מספר פעמים יש לכתבו כפונקציה.
 הנכם רשאים להוסיף כל פונקציה נוספת שתבחרו.

כיתבו תוכנית ראשית עבור הבדיקות.

התוכנית תקלוט קודם נתונים עבור מצולע ראשון ואחר כך נתונים עבור מצולע שני.

מותר להניח שכל מצולע בעלת 4 צלעות הוא מלבן (rectangle).

תוכנית מציגה למשתמש הודעה `enter number of sides for polygon 1` וקולטת את מספר הצלעות במצולע הראשון. לאחר מכן התוכנית מציגה למשתמש הודעה `enter sides for polygon` וקולטת את ערכי הצלעות של המצולע הראשון. אחר כך התוכנית בונה את המצולע הנכון בהתאם (במידה שהוכנסו 3 צלעות יבנה Triangle במידה שהוכנסו 4 צלעות יבנה Rectangle בכל מקרה אחר יבנה Polygon).

שימו לב: במידה ומספר הצלעות הוא 3 או 4 התוכנית הראשית תדפיס ההודעה "enter sides for polygon" ותקלוט את ערכי הצלעות – במידה ומספר הצלעות הינו כל מספר אחר הבנאי של Polygon ידפיס הודעה "enter sides for polygon" ויקלוט את ערכי הצלעות.

לאחר מכן, התוכנית מציגה למשתמש הודעה `enter number of sides for polygon 2` וקולטת את מספר הצלעות במצולע השני. התוכנית מציגה למשתמש הודעה `enter sides for polygon` וקולטת את ערכי הצלעות של המצולע השני. התוכנית בונה את המצולע הנכון בהתאם, כנ"ל.

סדנא ב-C++

במידה ומספר הצלעות במצולע הראשון שונה ממספר הצלעות במצולע השני – ערכי הצלעות של המצולע השני לא יקלטו כלל!

ניתן להניח שהקלט תקין, כלומר מספר הצלעות שווה למספר ערכי הצלעות שאחריו, פרט למקרה של מלבן, בו מספר הצלעות הוא 4 ואילו מספר אורכי הצלעות הוא 2, ושהמספרים הם שלמים חיוביים. ניתן גם להניח שהמצולע עם אורכי הצלעות שניתנו הוא אפשרי ושכל מרובע בשאלה הוא מלבן.

לאחר קלט הנתונים של שני המצולעים התוכנית תבדוק האם הם שווים ותדפיס `equal` אם הם שווים או

`not equal` אם לאו.

דוגמאת 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

דוגמאת 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

דוגמאת הרצה 3 :

enter number of sides for polygon 1:

4

enter sides for polygon:

20 15

enter number of sides for polygon 2:

סדנא ב-C++

4

enter sides for polygon:

15 10

not equal

דוגמאת 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

דוגמאת 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