

סדנא ב-C++

תרגיל 10

template ועצים

1. הוסף למחלקה Tree (שהוגדרה בכיתה) את הפונקציות הבאות:
 - `int leaves();` פונקציה המחזירה את מספר העלים בעץ (עלה הוא כל קודקוד שאין לו בנים בכלל).
 - `int height();` פונקציה המחזירה את גובה העץ (לצורך התרגיל נגדיר: עץ ריק גובהו 0, עץ עם שורש בלבד גובהו 1 וכו').
 - `void reflect();` פונקציה המחליפה בין הבנים של כל קודקוד בעץ ויוצר עץ חדש שהוא תמונת הראי של העץ המקורי.
 - `int onlyLeftSon();` פונקציה המחזירה את מספר הבנים השמאליים בעץ שהם בנים יחידים.

2. הוסף למחלקה SearchTree את הפונקציות הבאות
 - `void remove(T val);` פונקציה שמוחקת מהעץ את הקודקוד שמכיל את הערך `val`. **שים לב!** עבור מתודה זו תצטרך למצוא את היורש של הקודקוד הנמחק לפי כללי המחיקה מעץ.
 - `int level(T val);` פונקציה המחזירה את הרמה בעץ של הקודקוד שמכיל את הערך `val`. (שורש העץ נמצא ברמה 0 וכו' אם האיבר לא נמצא הפונקציה תחזיר -1).
3. צרף את המחלקות שכתבת בשאלות אחת ושתיים לתוכנית הראשית הבאה והראה את נכונותם:

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

int main()
{
    SearchTree<int> T1;
    cout<<"enter 10 numbers\n";
    int x,y;
    for (int i=0;i<10; i++)
    {
        cin>>x;
        T1.add(x);
    }
    cout<<"inorder: ";
    T1.inOrder();
    cout<<"\nenter 0-6:\n";
    cin>>x;
    while(x!=0)
    {
        switch (x)
        {
            case 1: cout<<"# of leaves: "<<T1.leaves()<<endl;
```

סדנא ב- C++

```

        break;
    case 2: cout<<"height of tree: "<<T1.height()<<endl;
        break;
    case 3:T1.reflect();
        cout<<"reflected tree: ";
        T1.inOrder();
        T1.reflect();
        cout<<endl;
        break;
    case 4: cout<<"# left sons only: "<<T1.onlyLeftSon()<<endl;
        break;
    case 5: cout<<"enter a number ";
        cin>>y;
        cout<<"level of "<<y<<" on tree: "<<T1.level(y)<<endl;
        break;
    case 6: cout<<"enter a number ";
        cin>>y;
        T1.remove(y);
        cout<<"after removing "<<y<<": ";
        T1.inOrder();
        cout<<endl;
    }
    cout<<"enter 0-6:\n";
    cin>>x;
}
return 0;
}

```

4. א. כתוב מחלקה לייצוג סטודנט. השדות במחלקה יהיו:

- מספר הזהות,
- שם משפחה
- ושם פרטי.

הוסף למחלקה לפחות את הפונקציות הבאות:

- קונסטרקטור
- אופרטורים <=<,<,>,>.
- הסדר הוא: ת.ז., שם משפחה, שם פרטי.
- אופרטורי יחס נוספים לפי הצורך.
- הסדר הוא: ת.ז., שם משפחה, שם פרטי.

השתמש במחלקות עץ (שאלות 1,2) וכתוב תוכנית אשר תנהל את רישום הסטודנטים במוסד. לצורך כך, השתמש במחלקת עץ חיפוש כמבנה הנתונים המאחסן את רשימת הסטודנטים.

על מנת לבדוק נכונות, יש להשתמש ולהגיש את הקבצים יחד עם הקובץ הבא:

```

#include <iostream>
using namespace std;
#include "SearchTree.h"
#include "Student.h"
int main(){
    SearchTree<Student> sList;
    Student tmp;

```

סדנא ב-C++

```
char choice = 'i';
while (choice != 'e'){
    cout<<"enter a-e\n";
    cin>>choice;
    switch (choice){
    case 'a':
        cout<<"enter a student\n";
        cin>>tmp;
        sList.add(tmp);
        break;
    case 'b':
        cout<<"enter a student\n";
        cin>>tmp;
        sList.remove(tmp);
        break;
    case 'c':
        cout<<"enter a student\n";
        cin>>tmp;
        if(sList.search(tmp))
            cout<<"exist\n";
        else
            cout<<"not exist\n";
        break;
    case 'd':
        sList.inOrder();
        break;
    case 'e':
        break;

    default:
        cout<<"error\n";
        break;
    }
}
return 0;
}
```