

תכנות מתקדם – 150024 תרגיל בית מספר 11 עצים ו-template

1. הוסף למחלקה Tree (שהוגדרה בכיתה) את המתודות הבאות:
- `int leaves();` פונקציה המחזירה את מספר העלים בעץ (עלה הוא קודקוד שאין לו בנים בכלל).
 - `int height();` פונקציה המחזירה את גובה העץ (נגדיר: גובה העץ הוא אורך המסלול הכי ארוך בעץ מהשורש עד לעלה, עץ ריק גובהו 1-, עץ עם שורש בלבד גובהו 0 וכו')
 - `void reflect();` פונקציה המחליפה בין הבנים של כל קודקוד בעץ ויוצרת 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`. במידה והערך `val` לא נמצא בעץ יש לזרוק חריגה "value not exist on tree"
 - `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-).

שימו לב – סטודנטים הניגשים לתרגיל זה לפני שלמדו `template` יגדירו את הפונקציות עם `int` במקום `T`:

```
void remove(int val)
int level(int val)
```

3. הורד מהמודל את הקובץ `main.cpp` המצורף והראה את נכונות המחלקות שהגדרת בסעיפים 1,2. אין לשנות שום דבר בקובץ `main`.
- שימו לב – סטודנטים הניגשים לתרגיל זה לפני שלמדו `template` יחליפו את השורה הראשונה בתוכנית מ-`SearchTree<int> T1;` ל-`SearchTree T1;`.

4. א. הערה מקדימה עבור סטודנטים שנגשו לשאלות 1-3 לפני שלמדו `template` – בשלב זה עליכם לחזור למחלקות `Tree` ו-`SearchTree` שהגדרתם בשאלות 1-2 ולעדכן אותם כך שייצגו עצים מטיפוס

כלשהו כלומר במקום int key יש לכתוב T key - ולתקן את הנדרש בהתאם. העזרו בקבצים המצורפים במודל בתיקיה 'מבנה נתונים - קוד'.

ב. נתונה (במודל) הגדרה חלקית של מחלקה בשם Student לייצוג תלמיד.

השדות במחלקה הם :

- מספר הזהות (int) – חייב להיות ייחודי,
- שם משפחה (string)
- ושם פרטי (string)

המתודות הנתונות הם :

- בנאי עם ערכי ברירת מחדל.
- פונקציה חברה עבור קלט
- ופונקציה חברה עבור פלט.

```
#include<iostream>
#include<string>
using namespace std;
class Student
{
protected:
    int id;
    string firstName;
    string lastName;

public:
    Student(int id = 0, string fname = "", string lname = "");
    friend ostream& operator<<(ostream&, const Student&);
    friend istream& operator>>(istream&, Student&);
};
```

הוסף למחלקה אופרטורים של השוואה לפי הצורך (אין להוסיף שום מתודה נוספת חוץ מאשר אופרטורים בלבד!)

ג. הגדר מחלקה בשם School לניהול רישום של תלמידים לבית הספר.

השדה היחיד בחלקה זו יהיה עץ חיפוש בינארי - SearchTree<Student> – השומר בתוכו את פרטי התלמידים מסודרים לפי מספר הזהות שלהם - השתמש במחלקות עץ שהגדרת (שאלות 1,2) – אין לשנות או להוסיף מחלקות אלו בכלל!

הוסף למחלקה School את המתודות הבאות :

- School() - בנאי שמאתחל עץ חיפוש בינארי ריק.
- void addStudent() - הוספת סטודנט לעץ. על המתודה לבקש - "enter id first and last name" ולקלוט את פרטי הסטודנט ולהכניסו לעץ.

במידה ומספר הזהות כבר קיים בעץ יש להדפיס "id must be unique" ולחזור על הבקשה
"enter id first and last name"

- void remove(int) – מתודה המקבלת מספר זהות של תלמיד ומחוקת אותו מהעץ, במידה והתלמיד לא קיים בעץ יש להדפיס "student not found"
- bool search(int) – מתודה המקבלת מספר זהות של תלמיד ובודקת האם התלמיד רשום במערכת או לא.
- void print() – הדפסה של נתוני כל התלמידים הרשומים לבית הספר ממוינים לפי מספר הזהות שלהם.

ד. הורד מהמודל את הקובץ main.cpp המצורף והראה את נכונות המחלקות שהגדרת, אין לשנות שום דבר בקובץ main,