Лабораторная работа №4

***I Код (написан на языке Pascal)***

**Program** lab4;

**uses** crt;

**Type**

MedArray = **array** [1..2] **of** real;

Elev = **Record**

ord: integer;

nume: string[24];

mediile: MedArray;

anuala: real;

**end**;

Tree = ^Nod;

Nod = **Record**

inf: Elev;

st, dr: Tree;

**end**;

**Var**

arb: Tree;

key: 0..6;

caut: string[24];

val1, val2: real;

**Procedure** Inserare(**var** arb: Tree; x: Elev);

**begin**

**if** arb = Nil **then**

**begin**

**new**(arb);

arb^.inf:= x;

arb^.st:= Nil;

arb^.dr:= Nil;

**end**

**else**

**begin**

**if** x.nume < arb^.inf.nume **then** Inserare(arb^.st, x)

**else** Inserare(arb^.dr, x);

**end**;

**end**;

**Procedure** Adaugare;

**var** elev: Elev;

**begin**

write('Nr. ordin: '); readln(elev.ord);

write('Nume: '); readln(elev.nume);

write('Media pentru semestru 1: '); readln(elev.mediile[1]);

write('Media pentru semestru 2: '); readln(elev.mediile[2]);

write('Media anuala: '); readln(elev.anuala);

Inserare(arb, elev);

**end**;

**Procedure** Afisare(arb: Tree);

**begin**

**if** arb <> Nil **then begin**

Afisare(arb^.st);

writeln(arb^.inf.nume);

Afisare(arb^.dr);

**end**;

**end**;

**Procedure** CautSub(**var** curent, f: Tree);

**var** aux: Tree;

**begin**

**if** f^.dr <> Nil **then**

CautSub(curent, f^.dr)

**else begin**

curent^.inf:= f^.inf;

aux:= f;

f:= f^.st;

dispose(aux);

**end**;

**end**;

**Procedure** Eliminare(**var** arb: Tree; x: string[24]);

**var** aux: Tree;

**begin**

**if** arb <> Nil **then**

**if** arb^.inf.nume = x **then**

**if** (arb^.st = Nil) **and** (arb^.dr = Nil) **then begin**

dispose(arb);

arb:= Nil;

**end**

**else**

**if** arb^.st = Nil **then begin**

aux:= arb^.dr;

dispose(arb);

arb:= aux;

**end**

**else**

**if** arb^.dr = Nil **then begin**

aux:= arb^.st;

dispose(arb);

arb:= aux;

**end**

**else**

CautSub(arb, arb^.st)

**else**

**if** x < arb^.inf.nume **then** Eliminare(arb^.st, x)

**else** Eliminare(arb^.dr, x)

**else**

writeln('Numele nu exista');

**end**;

**Function** Cautare(arb: Tree; x: string[24]): Tree;

**begin**

**if** arb = Nil **then** writeln('Numele nu exista')

**else**

**if** x = arb^.inf.nume **then begin**

writeln('Elev ', arb^.inf.nume, ' ordin: ', arb^.inf.ord, ' media la sem 1: ', arb^.inf.mediile[1],

' media la sem 2: ', arb^.inf.mediile[2], ' media anuala: ', arb^.inf.anuala);

Cautare:= arb;

**end**

**else**

**if** x < arb^.inf.nume **then** Cautare:= Cautare(arb^.st, x)

**else** Cautare:= Cautare(arb^.dr, x);**end**;

**Procedure** Modificare(**var** arb: Tree; x: string[24]);

**var** aux: Tree; media: real;

**begin**

aux:= Cautare(arb, x);

write('Introduceti media anuala noua: ');

readln(media);

aux^.inf.anuala:= media;

**end**;

**Procedure** AfisareSem(arb: Tree; val1, val2: real);

**begin**

**if** arb <> Nil **then begin**

Afisare(arb^.st);

**if** ((arb^.inf.mediile[1] = val1) **and** (arb^.inf.mediile[2] = val2)) **then** writeln(arb^.inf.nume);

Afisare(arb^.dr);

**end**;

**end**;

**Begin**

arb:= Nil;

**Repeat**

clrscr;

writeln('1. Adaugarea unui elev nou');

writeln('2. Eliminarea unui elev');

writeln('3. Modificarea mediei anuale a unui elev');

writeln('4. Cautarea unui elev dupa nume');

writeln('5. Afisarea elevilor cu mediile semestriale cuprinse intre doua valori');

writeln('6. Afisarea arborelui');

writeln('0. Stop');

write('Indica operatia necesara: ');

readln(key);

**case** key **of**

1:

**begin**

Adaugare;

readln;

**end**;

2:

**begin**

write('Introduceti numele elevului: ');

readln(caut);

Eliminare(arb, caut);

readln;

**end**;

3:

**begin**

write('Introduceti numele elevului: ');

readln(caut);

Modificare(arb, caut);

readln;

**end**;

4:

**begin**

write('Introduceti numele elevului: ');

readln(caut);

Cautare(arb, caut);

readln;

**end**;

5:

**begin**

write('Introduceti valoarea 1: ');

readln(val1);

write('Introduceti valoarea 2: ');

readln(val2);

AfisareSem(arb, val1, val2);

readln;

**end**;

6:

**begin**

Afisare(arb);

readln;

**end**;

**end**;

**Until** key = 0;

**End**.