Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Лабораторна робота № 1**

**з дисципліни**

**«Системне програмування»**

Виконала: студентка групи ІО-91

Косейкіна Ганна Сергіївна

2011 р.

Мета: Вивчення типів таблиць (для пошуку за прямою адресою та лінійного і двійкового пошуку) в системних програмах і конструкцій базової мови програмування для їх визначення та обробки відповідними методами.

Залікова книжка №9123.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № вар. | Тип ключа для прямої адреси | Тип ключа для інших видів пошуку | Тип функціо­нального поля | Тип вибірки | |
| 3 | unsigned char | char\*\_ unsigned long | Struct | Черговий | |
| № вар. | Тип функції порівняння за мірою схожості | | | |
| 3 | Спiвпадiння максимальної кількості літер з врахуванням послідовності їх входження у ключі. | | | |

Лістінг:

**public** **class** lab2 {

**public** **static** **void** main(String[] args) {

record[] records = **new** record[10];

**for**(**int** i = 0; i < 10; i++)

records[i] = **new** record("r"+i, i, 10+i);

table t = **new** table(records[0]);

**for**(**int** i = 1; i < 10; i++)

t.add(records[i]);

t.show();

System.*out*.println("------FindNum------");

t.showElement(t.findNum(5));

System.*out*.println("------FindLin------");

t.showElement(t.findLin(records[6].keys));

System.*out*.println("------FindBin------");

t.showElement(t.findBin(records[9].keys));

System.*out*.println("------AddBin------");

record add = **new** record("10", 10, 100);

t.addBin(add.keys, add);

t.show();

System.*out*.println("------Del------");

t.del(9);

t.show();

System.*out*.println("------Pack------");

t.pack();

t.show();

}

}

**public** **class** table {

**public** table(record head){

**this**.head = **new** element(head);

**this**.last = **this**.head;

ln = 1;

}

**public** **void** add(record rec){

last.next = **new** element(rec);

last = last.next;

ln++;

}

**public** **void** addLin(key k, record rec){

element e = head;

**while**(e.next != **null**){

**if**((e.rec.keys.ARG2 < k.ARG2) &&(e.next.rec.keys.ARG2 >= k.ARG2)){

element t = **new** element(rec);

t.next = e.next;

e.next = t;

ln++;

**break**;

}

e = e.next;

}

}

**public** **void** addBin(key k, record rec){

element e = findBin(k);

**if**(e == **null**)

add(rec);

**else**{

element t = **new** element(rec);

t.next = e.next;

e.next = t;

}

ln++;

}

**public** element findLin(key k){

element e = head;

**while**(e.next != **null**){

**if**(e.rec.keys.ARG1 == k.ARG1)

**return** e;

e = e.next;

}

**return** **null**;

}

**public** element findNum(**int** index){

element e = head;

**for**(**int** i = 1; i < index+1; i++)

**if**(e.next == **null**)

**return** **null**;

**else**

e = e.next;

**return** e;

}

**public** element findBin(key k){

element t = head;

element e = t;

**int** ln = **this**.ln;

**while**(ln != 0){

e = t;

**for**(**int** i = 0; i <(ln/2); i++)

e = e.next;

**if**(e.rec.keys.ARG2 == k.ARG2)

**return** e;

**else** **if**(e.rec.keys.ARG2 < k.ARG2)

t = e;

ln /= 2;

}

e = e.next;

**if**(e.rec.keys.ARG2 == k.ARG2)

**return** e;

**return** **null**;

}

**public** **void** del(**int** index){

element e = findNum(index);

**if**(e != **null**)

e.rec.mode = -1;

**else**

System.*out*.println("----index incorrect----");

}

**public** **void** pack(){

element e = head;

**while**(e.next != **null**){

**if**(e.next.rec.mode == -1){

e.next = e.next.next;

ln--;

}

**else**

e = e.next;

}

**if**(head.rec.mode == -1){

head = head.next;

ln--;

}

}

**public** **void** showElement(element e){

**if**(e != **null**){

**if**(e.rec.mode == -1)

System.*out*.println("is deleted");

**else**

System.*out*.println(e.rec.keys.ARG1+" "+e.rec.keys.ARG2+" "+e.rec.fun.fun);

}**else**

System.*out*.println("record not found");

}

**public** **void** show(){

**int** i = 0;

element e = head;

System.*out*.println("------Table------");

**do**{

System.*out*.print(i+") ");

showElement(e);

e = e.next;

i++;

}

**while**(e != **null**);

}

**public** **int** ln;

**public** element head;

**public** element last;

}

**public** **class** element {

**public** element(record rec){

**this**.rec = rec;

}

**public** record rec;

**public** element next = **null**;

}

**public** **class** record {

**public** record(String ARG1, **long** ARG2, **int** fun){

**this**.keys = **new** key(ARG1, ARG2);

**this**.fun = **new** func(fun);

**this**.mode = 0;

}

**public** key keys;

**public** func fun;

**public** **int** mode;

}

**public** **class** func {

**public** func(**int** fun){

**this**.fun = fun;

}

**public** **int** fun;

}

**public** **class** index {

**public** index(**int** N){

ind = **new** index[N];

e = **new** element[N];

index = **new** **int**[N];

}

index[] ind;

element[] e;

**int**[] index;

}

**public** **class** key {

**public** key(String ARG1, **long** ARG2){

**this**.ARG1 = ARG1;

**this**.ARG2 = ARG2;

}

**public** String ARG1;

**public** **long** ARG2;

}