**МИНИСТЕРСТВО НАУКИ и ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Коломенский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Московский политехнический университет»**

**Направление:** Информатика и вычислительная техника

**Профиль:** Программное обеспечение вычислительной техники

и автоматизированных систем

**Форма обучения:** очная

**КУРСОВОЙ ПРОЕКТ**

По дисциплине «Структуры данных и алгоритмы их обработки»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_ / М.С. Хоботов

(подпись) И.О. Фамилия

Руководитель

курсового проекта

старший преподаватель

кафедры АП и ИТ \_\_\_\_\_\_\_\_\_\_\_\_\_ / И.Н.Филоненко

(подпись) И.О. Фамилия

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Коломна

2021

– организация хеш-таблицы с разрешением коллизий методом внешних цепочек;

– создание хеш-таблицы с использованием хеш-функции универсального хеширования, поиск и вставка элемента в хеш-таблицу, удаления элемента из хеш-таблицы.

Написать программу, иллюстрирующую работу с организованной библиотекой.

**Библиотека**

unit HobLibrary;

interface

uses

System.SysUtils;

type

pElem = ^Elem;

Elem = record

key: integer;

next: pElem;

end;

const

m = 20;

G = 19;

var

mas: array [0..1000] of pElem;

mas2: array [0..1000] of integer;

index: integer;

t\_val, u\_val: integer;

procedure AddElem(out out\_pos:integer; key:integer);

function h(t, u, G, x, R:integer): integer;

function FindElem(key:integer):string;

implementation

function h(t, u, G, x, R:integer): integer;

var

z: integer;

begin

z := (((t\*x)+u) mod G) mod R;

h := z;

end;

function FindElem(key:integer):string;

var position: integer;

table, p: pElem;

find: boolean;

begin

find:=false;

position := h(t\_val, u\_val, G, key, M) + 1;

table := mas[position];

while (find = false) and (table <> nil) do

begin

if table^.key = key then find:=true;

table:=table^.next;

end;

if not find then

FindElem := 'Ничего не найдено'

else

FindElem := 'Найдено в строке '+ inttostr(position);

end;

procedure AddElem(out out\_pos:integer; key:integer);

var position:integer;

p,table: pElem;

begin

position := h(t\_val, u\_val, G, key, M) + 1;

out\_pos := position;

GetMem(p,sizeof(Elem));

p^.key := key;

p^.next := nil;

table := mas[position];

if table = nil then

mas[position]:= p

else

begin

while (table^.next<>nil) do

begin

table := table^.next;

end;

table^.next := p;

end;

end;

end.

**Форма:**

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Grids, Vcl.StdCtrls, Math, HobLibrary;

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Button1: TButton;

Button2: TButton;

Edit1: TEdit;

Button4: TButton;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

out\_pos\_add: integer;

implementation

{$R \*.dfm}

{function h(k:integer):integer;

var x: integer;

a: real;

begin

a := (sqrt(5)-1)/2;

x:=Floor(m\*Abs(Floor(k\*A)-k\*A));

h := x;

end; }

procedure AddElemForm(pos\_l, key\_l: integer);

begin

if(Form1.StringGrid1.Cells[2,pos\_l] <> '') then

Form1.StringGrid1.Cells[2,pos\_l] := Form1.StringGrid1.Cells[2,pos\_l]+','+inttostr(key\_l)

else Form1.StringGrid1.Cells[2,pos\_l] := Form1.StringGrid1.Cells[2,pos\_l]+inttostr(key\_l);

if Form1.StringGrid1.Cells[3,pos\_l] = '' then

Form1.StringGrid1.Cells[3,pos\_l] := '1'

else Form1.StringGrid1.Cells[3,pos\_l] := inttostr(strtoint(Form1.StringGrid1.Cells[3,pos\_l])+1);

end;

procedure DeleteElem(key:integer);

var position, i, j: integer;

p,table: pElem;

s, s2: string;

begin

j := 0;

position := h(t\_val, u\_val, G, key, M) + 1;

table := mas[position];

while table <> nil do

begin

if table^.key = key then

begin

s:=Form1.StringGrid1.Cells[2,position];

s2 := IntToStr(key);

i := Length(s2) + 1;

if Form1.StringGrid1.Cells[2,(Pos(IntToStr(key),Form1.StringGrid1.Cells[2,position])-1)] = ',' then

delete(s,Pos(IntToStr(key),Form1.StringGrid1.Cells[2,position])-1,i)

else delete(s,Pos(IntToStr(key),Form1.StringGrid1.Cells[2,position]),i);

Form1.StringGrid1.Cells[2,position]:=s;

Form1.StringGrid1.Cells[3,position] := IntToStr(StrToInt(Form1.StringGrid1.Cells[3,position])-1);

table^.key := -1;

repeat

begin

if mas2[j] = key then

mas2[j] := -1;

inc(j);

end

until mas2[j] <> key;

break;

end

else

table:=table^.next;

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

var ch, i: integer;

flag: boolean;

begin

flag := false;

ch:=strtoint(Edit1.Text);

for i := 0 to index do

if mas2[i] = ch then flag := true;

if flag = false then

begin

mas2[index] := ch;

inc(index);

AddElem(out\_pos\_add, ch);

end;

AddElemForm(out\_pos\_add, ch);

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

showmessage(FindElem(strtoint(Edit1.Text)));

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

DeleteElem(strtoint(Edit1.Text));

end;

procedure TForm1.FormCreate(Sender: TObject);

var i:integer;

begin

index := 0;

t\_val := randomrange(1,G);

u\_val := randomrange(1,G);

for i := 0 to m do

begin

StringGrid1.Cells[0,i]:=inttostr(i);

StringGrid1.Cells[1,i+1]:=inttostr(i);

StringGrid1.Cells[2,i]:='';

StringGrid1.Cells[3,i]:='';

end;

StringGrid1.Cells[0,0]:='№ строки';

StringGrid1.Cells[1,0]:='Значение хеш-функции';

StringGrid1.Cells[2,0]:='Значение ключа';

StringGrid1.Cells[3,0]:='Количество коллизий';

end;

end.





