*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | ***«Московский государственный технический университет имени Н.Э. Баумана***  ***(национальный исследовательский университет)»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА: КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

**Отчет**

**по лабораторной работе № 9**

**Название лабораторной работы: *Динамические структуры данных. Списки.***

**Дисциплина:** Основы программирования

Студент гр. ИУ6-12Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В.Астахов**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

I вариант

Задание 1

С клавиатуры вводятся фамилии участников некоторых соревнований. Составить список участников, упорядочив его по алфавиту. Полученный список вывести на экран. Удалить из списка участника, фамилия которого вводится с клавиатуры.

Текст программы:

program Project2;

{$APPTYPE CONSOLE}

uses

SysUtils;

type

UchPointer = ^uchastnik;

uchastnik = record

fam: string;

nxt: UchPointer;

end;

var

p1, p2, pst, pfn: UchPointer;

uch1: uchastnik;

a, dTarget: string;

found: boolean;

procedure SortList(pf: UchPointer);

var

ptarget, buffer: UchPointer;

sorted, swapped: boolean;

bufferF: string;

begin

{ ptarget:=pf; }

sorted := false;

while not sorted do

begin

sorted := true;

ptarget := pf;

while (ptarget^.nxt <> nil) do

begin

{ swapped := false; }

if (ptarget^.fam > ptarget^.nxt^.fam) then

begin

sorted := false;

bufferF := ptarget^.fam;

ptarget^.fam := ptarget^.nxt^.fam;

ptarget^.nxt^.fam := bufferF;

{ swapped := true; }

end;

ptarget := ptarget^.nxt;

end;

end;

end;

procedure Disposer(pf: UchPointer);

var

p1, p2: UchPointer;

begin

p1 := pf;

while (p1.nxt <> nil) do

begin

p2 := p1^.nxt;

p1^.nxt := nil;

dispose(p1);

p1 := p2;

end;

dispose(p1);

end;

begin

writeln('> Enter surnames');

readln(a);

new(pst);

pst^.fam := a;

p1 := pst;

readln(a);

while (a <> '') do

begin

new(p2);

p2^.fam := a;

p1^.nxt := p2;

p1 := p2;

readln(a);

end;

p1^.nxt := nil;

p2^.nxt := nil;

{ pfn := p1; }

SortList(pst);

writeln('> Enter surname to delete');

readln(dTarget);

found := false;

if (pst^.fam = dTarget) then

pst := pst^.nxt

else

begin

p1 := pst;

while p1^.nxt <> nil do

begin

if p1^.nxt^.fam = dTarget then

begin

found := true;

p1^.nxt := p1^.nxt^.nxt;

end;

p1 := p1^.nxt;

end;

end;

if not found then

writeln('>> Surname not found');

writeln('> List:');

p1 := pst;

while (p1^.nxt <> nil) do

begin

writeln(p1^.fam);

p1 := p1^.nxt;

end;

writeln(p1^.fam);

Disposer(pst);

readln;

end.

Тесты

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| bob  april  may  jack  selena  jack | > List:  april  bob  may  selena | > List:  april  bob  may  selena |
| helen  helga  rob  jhon  sergey  maxim  bob | >> Surname not found  > List:  helen  helga  jhon  maxim  rob  sergey | >> Surname not found  > List:  helen  helga  jhon  maxim  rob  sergey |
| beta  alpha  gamma  zeta  delta  zeta | alpha  beta  delta  gamma | alpha  beta  delta  gamma |

Вывод:

* Линейные динамические структуры данных удобно реализовать через списки
* В Delphi списки реализуются через записи, одно из полей которых ссылается на запись того же типа
* Динамическую память выделяемую под указатели лучше очищать перед завершение программы

Схема алгоритма:





