# **Національний технічний університет України**

# **«Київський політехнічний інститут»**

# **Факультет інформатики та обчислювальної техніки**

# **Кафедра обчислювальної техніки**

# **Лабораторна робота №2**

### **З програмування**

**виконав студент першого курсу**

**група ІО-91**

**Нечитайло Олег Андрійович**

**Тема:** Списки в мові програмування Pascal.

**Мета:** Здобуття навичок у створенні процедур та функцій для роботи зі списками. Розробка діалогового інтерфейсу користувача та функціонального

наповнення. Розробка засобів обробки та редагування даних.

**Варіант 12**

**Завдання:**

**Тип даних:** Інформація про гірські вершини.

**Тип списку:** Однозв’язний не кільцевий.

**Спосіб створення:**

Додаванням n записів до списку. Кожен наступній запис додається в початок списку.

**Функції для додавання:**

Додавання 1 запису в початок списку.

Додавання n записів у вказану позицію списку.

**Функції для видалення:**

Видалення 1 запису з початку списку.

Видалення n записів з вказаної позиції списку.

**Функції перегляду:**

Перегляд записів починаючи з вказаної позиції до кінця списку.

Перегляд усіх записів списку.

**Лістинг модуля:**

unit lists;

interface

type

TPtr = ^TlistItem; *{Тип вказівника на елемент списку}*

TListItem = record *{Назва, висота, гірський хребет, країна, перше сходження}*

mname, height, mountains, country, ascent:string;

next:TPtr; *{Вкахівник на наступний елемент}*

end;

var

pfirst:TPtr; *{вказівник на початок списку}*

function create(n:integer):boolean;*{Створення. Повертає 0 якщо вже створено список}*

procedure addtostart; *{Додає 1 елемент в початок списку}*

function insert(n, pos:integer):boolean;*{Додає n елементів у позицію pos. 0 якщо її немає}*

function deletefromstart:boolean; *{видаляє елемент з початку. 0 – якщо немає списку}*

function deletefrompos(n, pos:integer):boolean;

*{Видаляє з позиції n елементів або поки не досягне кінця. 0 – якщо позиції немає}*

function showfrompos(pos:integer):boolean;

*{Виводить усі елементи, починаючи з pos. 0 – якщо позиції не існує}*

procedure showall;*{виводить усі елементи списку}*

*{Частина реалізації}*

implementation

procedure readfields(var p:TPtr);{Організує введення з клавіатури}

begin

write('name : ');

readln(p^.mname);

write('height : ');

readln(p^.height);

write('mountains range : ');

readln(p^.mountains);

write('country : ');

readln(p^.country);

write('first ascent : ');

readln(p^.ascent);

end;

function create(n:integer):boolean;

var

i:integer;

tmp:TPtr;

begin

if pfirst=nil then {Якщо списку немає то}

begin

for i:=1 to n do

begin

tmp:=pfirst;

new(pfirst);

readfields(pfirst);

pfirst^.next:=tmp;

end;

create:=true; {Функція відробила успішно}

end

else

create:=false; {Список вже створено}

end;

procedure addtostart;

var

tmp:TPtr;

begin

tmp:=pfirst;

new(pfirst);

pfirst^.next:=tmp;

readfields(pfirst);

end;

function insert(n, pos:integer):boolean;

var

curs, tmp:TPtr;

i:integer;

begin

curs:=pfirst;

if pfirst<>nil then

begin

if pos<=2 then {виключна ситуація, яка не вирішується в загальному випадку}

begin

if pos=1 then {якщо позиція=1 – то додаємо 1 елемент в початок і n-1 елемент на другу позицію}

begin

addtostart;

n:=n-1;

insert:=true;

end;

tmp:=curs^.next; {додавання на другу позицію}

insert:=true;

for i:=1 to n do

begin

new(curs^.next);

readfields(curs^.next);

curs^.next^.next:=tmp;

curs:=curs^.next;

end;

end

else

begin

{Загальний випадок. Відмотуємо до елементу, що стоїть перед позицією. -1 через те, що елемент перед позицією, ще -1 через те, що до циклу курсор стоїть на першому елементі}

for i:=1 to pos-2 do

begin

if curs=nil then

begin

insert:=false;

break;{якщо досягнено кінця списку – вийти з циклу}

end;

curs:=curs^.next;

end;

if curs<>nil then

begin

for i:=1 to n do {додавання елементів}

begin

tmp:=curs^.next;

new(curs^.next);

readfields(curs^.next);

curs^.next^.next:=tmp;

curs:=curs^.next;

end;

end;

end;

end;

end;

function deletefromstart:boolean;

var

tmp:TPtr;

begin

if pfirst=nil then {Якщо списку не існую – повертається помилка}

deletefromstart:=false

else

begin

tmp:=pfirst;

pfirst:=tmp^.next;

dispose(tmp);

deletefromstart:=true;

end;

end;

function deletefrompos(n, pos:integer):boolean;

var

curs, tmp:TPtr;

i:integer;

begin

if pfirst<>nil then

begin

deletefrompos:=true;

if pos=1 then

for i:=1 to n do{Якщо позиція = 1 то видалити н елементів з початку списку}

deletefromstart

else

begin

curs:=pfirst^.next;

tmp:=pfirst;

for i:=2 to pos-1 do {«Йдемо» до необхідної позиції}

begin

if curs=nil then

begin

deletefrompos:=false;

break;

end;

tmp:=curs;{запам’ятовуємо елемент перед курсором}

curs:=curs^.next;

end;

i:=1;

while (curs<>nil)and(i<=n) do {видаляємо н елементів або до кінця списку}

begin

curs:=curs^.next;

dispose(tmp^.next);

tmp^.next:=curs;

i:=i+1;

end;

end;

end

else

deletefrompos:=false;

end;

function showfrompos(pos:integer):boolean;

var

curs:TPtr;

i:integer;

err:boolean;

begin

curs:=pfirst;

if curs<>nil then

begin

err:=true;

for i:=1 to pos-1 do

begin

if curs^.next=nil then

begin

err:=false;

break;

end;

curs:=curs^.next;

end;

if err then

while curs<>nil do

begin

writeln('name : ', curs^.mname);

writeln('height : ', curs^.height);

writeln('mountains range : ', curs^.mountains);

writeln('country : ', curs^.country);

writeln('first ascent : ', curs^.ascent);

writeln('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');

curs:=curs^.next;

end;

end

else

err:=false;

showfrompos:=err;

end;

procedure showall;

var

curs:TPtr;

begin

curs:=pfirst;

while (curs<>nil) do

begin

writeln('name : ', curs^.mname);

writeln('height : ', curs^.height);

writeln('mountains range : ', curs^.mountains);

writeln('country : ', curs^.country);

writeln('first ascent : ', curs^.ascent);

writeln('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');

curs:=curs^.next;

end;

end;

end.

program S2Lab2;

uses Lists, crt;

var

menuloc, n, pos:integer; {Змінні для введення параметрів}

answ:integer;{Змінна для зчитування відповіді користувача}

quit:boolean;{Флаг виходу}

begin

quit:=false;

menuloc:=1; {Головне меню}

ClrScr;

while not quit do{Головний цикл}

begin

case menuloc of {Вивід меню}

1: begin

writeln('Main menu');

writeln('0 - Create');

writeln('1 - Add item menu');

writeln('2 - Delete item menu');

writeln('3 - Show item menu');

writeln('4 - Exit');

write('choose : ');

readln(answ);

ClrScr;

case answ of

0: begin

write('number : ');

readln(n);

if n<1 then

writeln('ERROR: NUMBER < 1')

else

if not create(n) then

{Тут і надалі: усі функції, що мають повертати код помилки, викликаються всередині оператора if , і якщо повернено 0, то виводиться повідомлення про помилку}

writeln('ERROR: ALREADY CREATED');

end;

1: menuloc:=2;

2: menuloc:=3;

3: menuloc:=4;

4: quit:=true;{exit}

end;

end;

2: begin

writeln('Add item menu');

writeln('1 - Add item to start');

writeln('2 - insert items inside');

writeln('3 - Back');

write('choose : ');

readln(answ);

case answ of

1: addtostart;

2: begin

write('number of elements : ');

readln(n);

if n<1 then

writeln('ERROR: NUMBER < 1')

else

begin

write('position : ');

readln(pos); {Правильність позиції аналізується всередині коду функції}

if not insert(n, pos) then

writeln('ERROR: POSITION NOT FOUND');

end;

end;

3: menuloc:=1;

end;

end;

3: begin

writeln('Delete items menu');

writeln('1 - Delete from start');

writeln('2 - Delete from position');

writeln('3 - Back');

write('choose : ');

readln(answ);

case answ of

1: begin

if not deletefromstart then

writeln('ERROR: LIST DOESN`T EXIST');

end;

2: begin

write('number : ');

readln(n);

if n<1 then

writeln('ERROR: NUMBER < 1')

else

begin

write('position : ');

readln(pos);

if not deletefrompos(n, pos) then

writeln('ERROR: POSITION MNOT FOUND');

end;

end;

3: menuloc:=1;

end;

end;

4: begin

writeln('Show items menu');

writeln('1 - Show from position');

writeln('2 - Show all');

writeln('3 - Back');

write('choose : ');

readln(answ);

case answ of

1: begin

write('position : ');

readln(pos);

if not showfrompos(pos) then

writeln('ERROR: POSITION NOT FOUND');

end;

2: ShowAll;

3: menuloc:=1;{Return to menu 1}

end;

end;

end;

end;

end.