Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №6**

**«РЕАЛИЗАЦИЯ ЭЛЕМЕНТАРНЫХ СТРУКТУР ДАННЫХ НА ОСНОВЕ СТАТИЧЕСКОЙ ПАМЯТИ»**

Выполнил: студент учебной группы

ИСПк-202-52-00

Домрачев Артём Андреевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

Вариант: 16

Цель работы: изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню.  
Задание:  
1. Написать программу для работы со структурой данных "Кольцевой двусвязный список".  
2. Структура данных должна быть реализована на основе статической памяти.  
3. Работа со структурой должна осуществляться с помощью case-меню. Предусмотреть наглядную визуализацию содержимого структуры.

3. Код программы

**uses** crt;

**type**

TData = Integer;

TPElem = ^TElem;

TElem = **record**

Data: TData;

FDel: Boolean;

PNext, PPrev: TPElem;

**end**;

// Список.

TDList = **record**

Cnt: Integer;

PFirst: TPElem;

**end**;

// Меню.

TMassive = **array** [1..100] **of** string[128];

**var**

punkt: TMassive;

num: integer;

L: TDList;

**procedure** Free(**var** aList: TDList);

**var**

PDel: TPElem;

c: char;

pos: Integer;

**begin**

**if** aList.PFirst = nil **then Exit**;

writeln('Укажите номер элемента, который нужно удалить: ');

readln(pos);

PDel := aList.PFirst;

**repeat**

PDel := PDel^.PNext;

**if** PDel = aList.PFirst **then**

**begin**

writeln('Элемента с таким номером не существует');

writeln('Для перехода к оглавлению нажмите Enter');

**repeat**

c := readkey;

**until** c = #13;

**Exit**;

**end**;

**until** pos <= 1;

**if** PDel^.PNext = PDel **then**

aList.PFirst := nil

**else**

**begin**

**if** PDel = aList.PFirst **then**

aList.PFirst := PDel^.PNext;

PDel^.PPrev^.PNext := PDel^.PNext;

PDel^.PNext^.PPrev := PDel^.PPrev;

**end**;

Dispose(PDel);

Dec(aList.Cnt);

writeln;

writeln('Элемент удален');

writeln('Для перехода к оглавлению нажмите Enter');

**repeat**

c := readkey;

**until** c = #13;

**end**;

**procedure** Add(**var** aList: TDList; **const** aData: TData);

**var**

PElem: TPElem;

**begin**

**New**(PElem);

PElem^.Data := aData;

PElem^.FDel := False;

**if** aList.PFirst = nil **then**

**begin**

aList.PFirst := PElem;

PElem^.PNext := PElem;

PElem^.PPrev := PElem;

**end**

**else**

**begin**

PElem^.PNext := aList.PFirst;

PElem^.PPrev := aList.PFirst^.PPrev;

PElem^.PPrev^.PNext := PElem;

PElem^.PNext^.PPrev := PElem;

**end**;

Inc(aList.Cnt);

**end**;

// Диалог для добавления элементов в конец списка.

**procedure** WorkAdd(**var** aList: TDList);

**var**

c: char;

S: String;

Data: TData;

Code: Integer;

**begin**

clrscr;

Writeln('Добавление элементов в список.');

Writeln('Ввод каждого значения завершайте нажатием Enter.');

Writeln('Чтобы прекратить ввод оставьте пустую строку и нажмите Enter.');

**repeat**

Write('Элемент №', aList.Cnt + 1, ': ');

Readln(S);

**if** S = '' **then**

**begin**

Writeln('Отмена.');

Code := 0;

**end**

**else**

**begin**

Val(S, Data, Code);

**if** Code = 0 **then**

**begin**

Add(aList, Data);

Writeln('Элемент добавлен.');

Code := 1;

**end**

**else**

Writeln('Неверный ввод. Повторите.');

**end**;

**until** Code = 0;

Writeln('Ввод элементов списка завершён.');

writeln;

writeln('Для перехода к оглавлению нажмите Enter');

**repeat**

c := readkey;

**until** c = #13;

**end**;

// Распечатка двунаправленного кольцевого списка.

**procedure** Print(**const** aList: TDList);

**var**

PElem: TPElem;

c: char;

**begin**

**if** aList.PFirst = nil **then Exit**;

PElem := aList.PFirst;

**repeat**

**if** PElem <> aList.PFirst **then** Write(', ');

Write(PElem^.Data);

PElem := PElem^.PNext;

**until** PElem = aList.PFirst;

writeln;

writeln('Для перехода к оглавлению нажмите Enter');

**repeat**

c := readkey;

**until** c = #13;

**end**;

**procedure** menu(**var** punkt: TMassive; **var** num: integer);

**var**

x, y, i: integer;

c: char;

**begin**

clrscr;

x := 1;

y := 1;

gotoxy(x, y);

textcolor(White);

write('Список доступных действий:');

x := 1;

y := 1;

num := 1;

**repeat**

**for** i := 1 **to** 4 **do**

**begin**

gotoxy(x, y + i);

**if** i = num **then**

**begin**

textcolor(0);

textbackground(15);

**end**

**else**

**begin**

textcolor(15);

textbackground(0);

**end**;

write(punkt[i]);

**end**;

c := readkey;

**if** c = #0 **then**

**begin**

c := readkey;

**case** c **of**

#32:

**if** num = 1 **then**

num := 4

**else**

Dec(num);

#40:

**if** num = 4 **then**

num := 1

**else**

Inc(num);

**end**;

**end**;

**until** c = #13;

textcolor(15);

textbackground(0);

**end**;

**begin**

clrscr;

writeln;

pos := 1;

punkt[1] := 'Добавление элемента в список';

punkt[2] := 'Показать список';

punkt[3] := 'Удалить список';

punkt[4] := 'Выход';

**repeat**

menu(punkt, num);

**case** num **of**

1:

WorkAdd(L);

2:

**if** L.PFirst = nil **then**

Writeln('Список пуст.')

**else**

**begin**

writeln;

Writeln('Содержимое списка:');

Print(L);

Writeln;

**end**;

3, 4:

Free(L);

**else**

Writeln('Незарегистрированная команда. Повторите ввод.');

**end**;

**until** num = 4;

writeln;

writeln('Работа программы завершена. Для выхода нажмите Enter');

Readln;

**end**.

4. Результаты выполнения программы

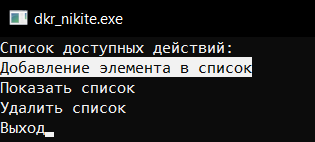


Рисунок 1 – Главное меню программы

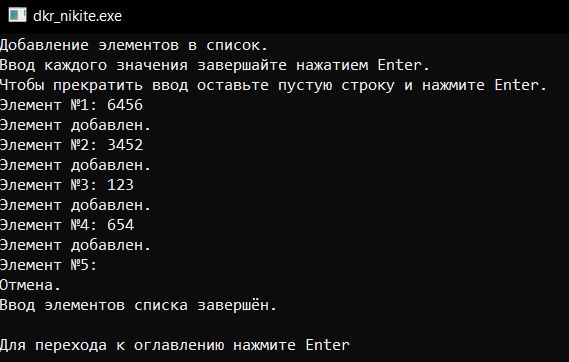


Рисунок 2 – Добавление элементов в список.

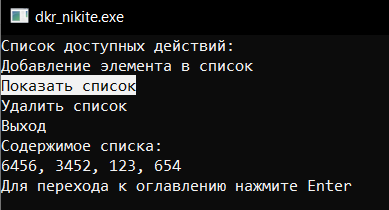


Рисунок 3 – Содержимое списка

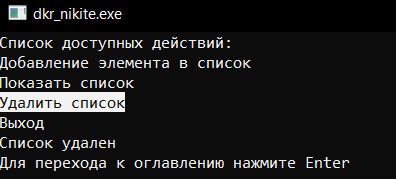


Рисунок 4 – Удаление списка

5. Вывод

При выполнении шестой домашней контрольной работы "Реализация элементарных структур данных на основе статической памяти" мы создали программу, которая реализует кольцевой двусвязный список. Пользователь имеет возможность выбрать операции из меню, такие как вставка данных, удаление данных, вывод списка данных или выход из меню. Для создания новых узлов в списке программа использует статическую память, и удаление узлов также осуществляется с помощью статической памяти. Вывод списка данных происходит циклически, проходя по всем узлам списка и выводя значение каждого узла.

Мы успешно справились с выполнением работы в полном объёме. В процессе выполнения нам помогли электронные ресурсы, знания, полученные на занятиях благодаря преподавателю, а также знания, полученные в результате самостоятельного изучения.