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»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-204-52-00

Бакулин Семён Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. Цель домашней контрольной работы:

Изучение принципов работы с базовыми структурами данных, получение навыков организации case-меню.

1. Вариант: 2.

Формулировка задания:

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.

1. 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.//

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

**var**

punkt: massiv;

num: integer;

pos, t: integer;

L: TDList;

Cmd, Cnt: Integer;

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

**var**

PElem, PDel: TPElem;

c: char;

pos: Integer;

**begin**

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

PElem := aList.PFirst;

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

readln(pos);

**begin**

PElem := PElem^.PNext;

**if** PElem = 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**;

**end**;

**if** PElem^.PNext = PElem **then**

aList.PFirst := nil

**else**

**begin**

PDel := PElem^.PNext;

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

aList.PFirst := PElem;

PElem^.PNext := PDel^.PNext;

PDel^.PNext^.PPrev := PElem;

**end**;

Dec(aList.Cnt);

Dispose(PDel);

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: massiv; **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;

{Init(L); }

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('Спиосок 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**.

1. Результат выполнения программы:

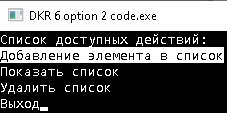


Рисунок 1 – Главное меню программы

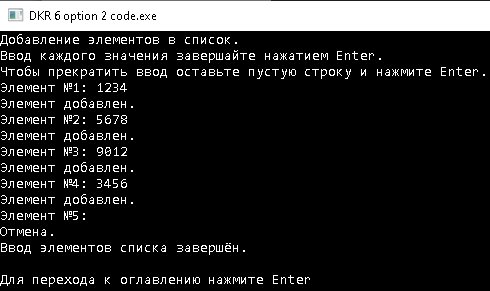


Рисунок 2 – Добавление элементов в список

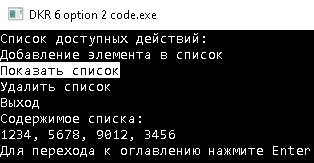


Рисунок 3 – Содержимое списка

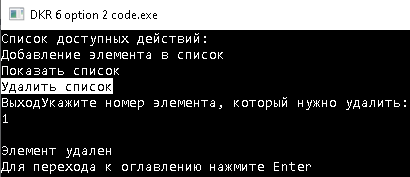


Рисунок 4 – Удаление списка

1. Вывод:

В ходе выполнения домашней контрольной работы №6 «Реализация элементарных структур данных на основе статической памяти»:

Данная программа реализует кольцевой двусвязный список (двунаправленный список). Пользователь может выбрать действие из меню: вставка данных, удаление данных, вывод списка данных или выход из меню. Программа использует статическую память для создания новых узлов в списке. Удаление узла также реализовано с помощью статической памяти. Вывод списка данных происходит циклически, проходя по всем узлам списка и выводя значение каждого узла.

С работой справился в полном объёме. В работе помогали электронные ресурсы, знания, полученные во время занятий благодаря преподавателю и знания, полученные в процессе саморазвития.