ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ#3

за курсом «Алгоритмы и структуры данных»

студентки группы ПА-19-2

Мовсисян Лауры Ростомовны

кафедра компьютерных технологий, ДНУ

2019/2020уч. Год

Односвязные списки

Вариант №8

1. Постановка задачи:

Реализовать односвязный список с основными операциями. Выполнить задачу по варианту.

1. Описание решение

Разбил задачи на функции, которые вызываются с меню.Функции с задачами вызываются через Switch-caseпо соответствующему номеру задачи. Реализовал односвязный список с основными его операциями.

1. Исходный текст программы

#include<iostream>

usingnamespace std;

//variant 8

void Menu();

void AddElem();

void RMElem();

void ShowElem();

structSll//Singly linked list

{

int elem;

Sll\* next;

public:

Sll()

{

next = NULL;

}

};

Sll\* sll;

void AddElem(intelem\_)

{

if (sll != NULL)

{

Sll\* x = newSll;

x->elem = elem\_;

x->next = sll;

Sll\* buff = x;

x = sll;

sll = buff;

}

else

{

sll = newSll();

sll->elem = elem\_;

}

}

void RMElem(intind)

{

Sll\* buff = sll;

Sll\* prev = NULL;

bool ident = 1;

for (int i = 1; i != ind;)

{

if (buff->next != NULL)

{

prev = buff;

buff = buff->next;

i++;

}

else

{

cout <<"Такого элемента не существует"<< endl;

ident = 0;

system("Pause");

break;

}

}

if(ident)

if (prev == NULL)

{

buff = sll->next;

delete(sll);

sll = buff;

}

elseif (buff->next == NULL)

{

delete(buff);

prev->next = NULL;

}

else

{

prev->next = buff->next;

delete(buff);

}

}

void Task()

{

ShowElem();

int count = 0;

Sll\* buff = sll;

for (int i = 1; buff->next!=NULL ; i++)

{

buff = buff->next;

if (i % 3 == 0)

{

RMElem(i-count);

count++;

}

}

cout << endl << endl;

ShowElem();

}

void ShowElem()

{

Sll\* buff = sll;

for (int i = 1; buff != NULL; i++)

{

cout << buff->elem << endl;

buff = buff->next;

}

}

void RMSll()

{

if(sll != NULL)

if (sll->next == NULL)

{

RMElem(1);

sll = NULL;

return;

}

else

{

Sll\* buff1 = sll;

Sll\* buff2 = buff1->next;

for (; buff2->next != NULL;)

{

delete(buff1);

buff1 = buff2;

buff2 = buff2->next;

}

delete(buff1);

sll = NULL;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

sll = newSll;

sll = NULL;

Menu();

system("Pause");

}

void Menu()

{

bool menu\_i = true;

while (menu\_i)

{

system("cls");

cout <<"1) Добавитьэлемент"<< endl;

cout <<"2) Удалитьэлемент"<< endl;

cout <<"3) Вывести элементы на экран"<< endl;

cout <<"4) Очистить список"<< endl;

cout <<"5) Инд. задание"<< endl;

cout <<"0) Выход"<< endl;

cout << endl;

cout <<"Введите номер пункта, к которому хотите перейти - ";

int n\_; cin >> n\_; cout << endl;

switch (n\_)

{

default:

menu\_i = 0;

break;

case 1:

cout <<"Введитеэлемент (int): ";

int el; cin >> el;

AddElem(el);

system("Pause");

break;

case 2:

cout <<"Введитеномерэлемента: ";

int ind; cin >> ind;

RMElem(ind);

system("Pause");

break;

case 3:

ShowElem();

system("Pause");

break;

case 4:

RMSll();

system("Pause");

break;

case 5:

Task();

system("Pause");

break;

case 0:

menu\_i = 0;

system("Pause");

break;

}

}

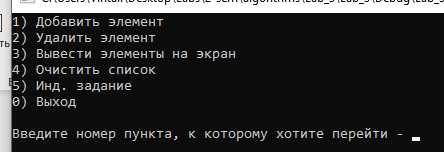
}

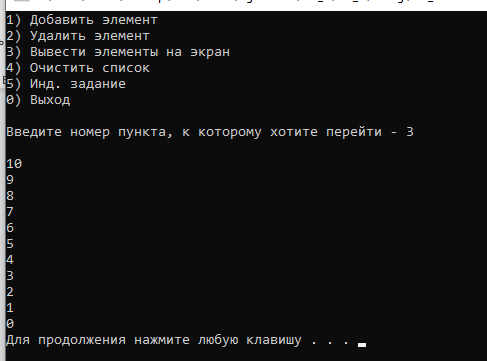
1. Описание интерфейса

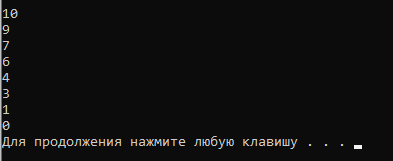
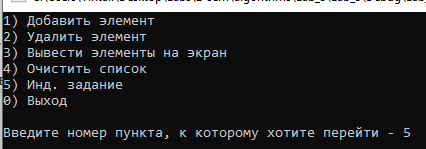
При запуске программы выводится меню. Чтобы перейти к пункту меню, необходимо ввести номер пункта с клавиатуры и нажать ENTER. Дальше следовать инструкциям.

1. Описание тестовых примеров

Вводил разные значения, после чего добавлял ограничения на вводимые данные. Если данные не соответсвовали требуемым, консоль очищалась, выводилось сообщение об ошибке ввода. После этого подпрограмма перезапускается.







Выводы: Структура данных имеет специфическое назначение. Поиск элементов в такой структуре имеет линейную сложность.Преемущество заключается в быстрой манипуляции элементами (удалить, добавить элемент в определённом месте).