БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Проверка примеров**

**По:** Основы алгоритмизации и программирования

**Выполнил:** Корнелюк

Валентин Владимирович

1 курс 4 группа ПОИТ

**Преподаватель:**

Белодед Николай Иванович

**г. Минск**

2023 г.

Построение однонаправленного списка с заглавным звеном и его просмотр.

#include<iostream>

using namespace std;

class Spisok

{

private:

struct node

{

int elem;

node\* sled;

} \*phead; //Указатель на начало списка.

public:

Spisok() { phead = new (node); (\*phead).sled = NULL; } //Конструктор.

~Spisok() { delete phead; } //Деструктор.

void POSTROENIE();

void VYVOD();

void OCHISTKA();

};

void main()

{

setlocale(LC\_ALL, "Rus");

Spisok A;

A.POSTROENIE();

A.VYVOD();

A.OCHISTKA();

cout << "\n";

system("PAUSE");

}

void Spisok::POSTROENIE()

//Построение однонаправленного списка с заглавным звеном.

// phead - указатель на заглавное звено списка.

{

node\* t;

int el;

t = phead;

cout << "Вводите элементы списка: ";

cin >> el;

while (el != 0)

{

(\*t).sled = new (node);

t = (\*t).sled; (\*t).elem = el; (\*t).sled = NULL;

cin >> el;

}

}

void Spisok::VYVOD()

//Вывод содержимого однонаправленного линейного списка

//с заглавным звеном.

// phead - указатель на заглавное звено списка.

{

node\* t;

t = (\*phead).sled;

cout << "Список: ";

while (t != NULL)

{

cout << (\*t).elem << " ";

t = (\*t).sled;

}

cout << endl;

}

void Spisok::OCHISTKA()

//Удаление однонаправленного списка из памяти.

// phead - указатель на заглавное звено списка.

{

node\* q, \* q1;// Рабочие указатели.

q = phead;

q1 = (\*q).sled; // Указатель q1 "опережает" указатель q.

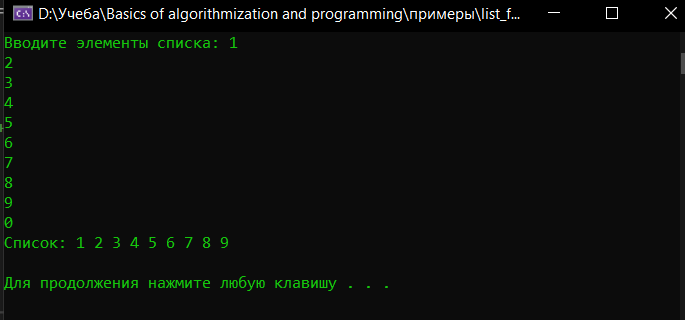
while (q1 != NULL)

{

q = q1; q1 = (\*q1).sled; delete q;

}

}



Включение звена с заданным элементом в однонаправленный список с заглавным звеном.

#include<iostream>

using namespace std;

struct node

{

int elem;

node\* sled;

};

class Spisok

{

private:

node\* phead; //Указатель на заглавное звено списка.

node\* Res; //Указатель на найденное звено списка.

public:

Spisok() { phead = new(node); (\*phead).sled = NULL; Res = NULL; }

~Spisok() { delete phead; }

void POSTROENIE();

void VYVOD();

node\* POISK(int);

void VSTAV(int);

void VSTAV1(int);

void OCHISTKA();

};

void main()

{

setlocale(LC\_ALL, "Rus");

Spisok A;

int el, el1;

A.POSTROENIE();

A.VYVOD();

cout << "\nВведите элемент звена, после которого ";

cout << "осуществляется вставка:\n";

cin >> el;

cout << "Введите элемент вставляемого звена:\n";

cin >> el1;

if (A.POISK(el) != NULL)

{

A.VSTAV(el1); A.VYVOD();

}

else cout << "Звена с заданным элементом в списке нет!";

cout << "\nВведите элемент звена, перед которым ";

cout << "осуществляется вставка:\n";

cin >> el;

cout << " Введите элемент вставляемого звена:\n";

cin >> el1;

if (A.POISK(el) != NULL)

{

A.VSTAV1(el1); A.VYVOD();

}

else cout << " Звена с заданным элементом в списке нет!";

A.OCHISTKA();

cout << "\n";

system("PAUSE");

}

void Spisok::POSTROENIE()

// Построение однонаправленного списка с заглавным звеном.

// phead - указатель на заглавное звено списка

{

node\* t;

int el;

t = phead;

cout << "Вводите элементы звеньев списка: ";

cin >> el;

while (el != 0)

{

(\*t).sled = new (node);

t = (\*t).sled; (\*t).elem = el; (\*t).sled = NULL;

cin >> el;

}

}

void Spisok::VYVOD()

// Вывод содержимого однонаправленного списка с

// заглавным звеном. phead - указатель на заглавное звено списка.

{

node\* t;

t = (\*phead).sled;

cout << "Список: ";

while (t != NULL)

{

cout << (\*t).elem << " "; t = (\*t).sled;

}

cout << endl;

}

node\* Spisok::POISK(int el)

//Поиск звена с элементом el в списке, заданном указателем

//phead. В случае успешного поиска в Res находится адрес

//звена списка, содержащего элемент el, в случае неуспеха

//в Res помещается NULL.

{

node\* t;

Res = NULL; t = (\*phead).sled;

while (t != NULL && Res == NULL)

if ((\*t).elem == el)

Res = t;

else t = (\*t).sled;

return Res;

}

void Spisok::VSTAV(int el)

// Включение звена с информационным полем el

// после звена, на которое указывает ссылка Res.

{

node\* q;

q = new(node);

(\*q).elem = el;

(\*q).sled = (\*Res).sled; (\*Res).sled = q;

}

void Spisok::VSTAV1(int el)

// Включение звена с информационным полем el

// перед звеном, на которое указывает Res.

{

node\* q;

q = new (node);

(\*q).elem = (\*Res).elem; (\*q).sled = (\*Res).sled;

(\*Res).elem = el; (\*Res).sled = q;

}

void Spisok::OCHISTKA()

//Удаление однонаправленного списка из памяти.

// phead - указатель на заглавное звено списка.

{

node\* q, \* q1;// Рабочие указатели.

q = phead;

q1 = (\*q).sled; // Указатель q1 "опережает" указатель q.

while (q1 != NULL)

{

q = q1; q1 = (\*q1).sled; delete q;

}

}

