**Новосибирский Государственный Технический Университет**

**Кафедра АСУ**

Отчёты по лабораторным работам

по дисциплине «Языки программирования»

Факультет: **АВТФ**

Преподаватель:

Лауферман О.В.

Группа: **АВТ-813**

Выполнили:

Бригада №6

Гаськов Николай

Аверьянов Роман

Шумкин Владислав

**2019 год**

**Лабораторная работа № 5. Шаблоны функций и шаблонные функции. Шаблоны структур.**

**Задание:** оставить в списке только последние вхождения одинаковых элементов, вернуть количество удаленных.

Параметризация: список

Тривиальный случай: список пуст

Декомпозиция общего случая: проверка числа в списке на наличие повтора и его удаление при условии нахождения повторяющегося числа

**Тестовые данные**

|  |  |
| --- | --- |
| 1 | **Ввод:**  Ввод количества элементов в списке: 10  Ввод элементов с списке: 1 2 1 1 3 2 1 3 3 2 |
| **Вывод:**  Удалено элементов: 7  Преобразованный список: 1 3 2 |
| 2 | **Ввод:**  Ввод количества элементов в списке: 5  Ввод элементов с списке: 1 2 3 4 5 |
| **Вывод:**  Удалено элементов: 0  Преобразованный список: 1 2 3 4 5 |
| 3 | **Ввод:**  Ввод количества элементов в списке: 10  Ввод элементов с списке: 5 3 1 9 1 2 2 9 9 9 |
| **Вывод:**  Удалено элементов: 5  Преобразованный список: 5 3 1 2 9 |
| 4 | **Ввод:**  Ввод количества элементов в списке: 7  Ввод элементов с списке: 2 2 8 4 8 3 2 |
| **Вывод:**  Удалено элементов: 3  Преобразованный список: 4 8 3 2 |

**Текст программы на языке С (С++)**

#include <fstream>

#include <iostream>

#include <cstdlib>

#include <string.h>

using namespace std;

#define BUFFER\_SIZE 100

template <class T> // шаблон структуры

struct list

{

T value;

list<T> \*next;

};

template <class T>

list<T>\* Del\_value(T n, list<T> \*head)

{

list<T> \*p = head, \*t;

if (head == NULL)

{

cout << "Список пуст!" << endl;

return NULL;

}

// удаляем первый элемент

if (head->value == n)

{

t = head;

head = head->next;

delete t;

return head;

}

// ищем в списке значение

while (p->next != NULL)

if (p->next->value == n)

{

t = p->next;

p->next = p->next->next;

delete t;

return head;

}

else p = p->next;

cout << "Такого значения нет!" << endl;

return head;

};

template <class T>

list<T> \*delete\_same(list<T> \*head, int &count)

{

if (head == NULL)

{

cout << "0" << endl;

return NULL;

}

list<T> \*q = head;

list<T>\*tmp = q->next;

count = 0;

int k = 0;

while (q->next != NULL)

{

k = 0;

while (tmp != NULL)

{

if (tmp->value == q->value)

{

head = Del\_value(q->value, head);

q = head;

count++;

k = 1;

}

tmp = tmp->next;

}

if (k == 0) q = q->next; //не было удалений в списке

tmp = q->next;

}

return head;

};

template <class T>

list<T> \*Ins\_last(list<T> \*head, T n) // шаблон функции Включение в конец списка

{

list<T> \*q = new list<T>, \*t;

q->value = n;

q->next = NULL;

if (head == NULL)

return q;

t = head;

while (t->next != NULL)

t = t->next;

t->next = q;

return head;

};

template <class T>

void Print\_list(list<T> \*head) // шаблон функции Печать списка

{

list<T> \*t = head;

if (head == NULL)

cout << "Список пуст" << endl;

while (t != NULL)

{

cout << " -> " << t->value;

t = t->next;

}

cout << endl;

};

template<class T>

list<T> \*create\_list()

{

T tmp;

int count;

list<T> \*first = NULL;

cout << "Введите количество элементов в списке: ";

cin >> count;

cout << "Введите элементы списка: " << endl;

for (int i = 1; i <= count; i++)

{

cin >> tmp;

first = Ins\_last(first, tmp);

}

cout << "Исходный cписок: " << endl;

Print\_list(first);

return first;

};

template <class T>

void lastsame()

{

int count = 0;

list<T>\*spisok = create\_list<T>();

if (spisok == NULL) return;

else {

spisok = delete\_same(spisok, count);

cout << "Число удаленных элементов: " << count << endl;

cout << "Список без повторяющихся элементов: " << endl;

Print\_list(spisok);

}

};

int main()

{

setlocale(LC\_ALL, "rus");

lastsame<int>();

system("pause");

return 0;

}