

**Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный
университет»**

Кафедра технологий
программирования

Основы алгоритмизации и программирования
Отчет по лабораторной работе №7
Вариант 11

Выполнил

Ланцев Евгений Николаевич.
21-ИТ-1, ФИТ

Проверил

Пантелейко А.Ф.
Преподаватель-стажер

Полоцк
2022 г.

Лабораторная работа № 6

“Введение в абстрактные структуры”

Цель работы: изучить методы организации списочных структур в динамической памяти. Реализовать алгоритмы добавления, удаления и сортировки списков.

Вариант 11

“Линейный кольцевой список типа float”

```
struct node {  
    float data; node *next; node(float c_data, node *c_next){data = c_data; next = c_next;}  
};
```

You, 1 second ago • Uncommitted changes

Рисунок 1 - Структура узла.

```
class linked_list {  
    node *root; // указатель на последний и  
public:  
    linked_list(){ root = new node(0, root);}
```

Рисунок 2 - Структура линейного кольцевого списка.

```

node* get_last(node* temp){
    return temp->next != root ? get_last(temp->next) : temp;
}
void insert_root_node(float data) { // Вставка элемента сначала, меняем корневой элемент
    node* new_node = new node(data,root); // Создаем новый корень.
    node *last_node = get_last(root); // Ищем последний элемент.
    root = new_node; // Меняем корень списка на новый узел
    last_node->next = root; //
}
void insert_last_node(float data) {
    node* new_node = new node(data,root); // Создаем новый корень.
    node *last_node = get_last(root); // Ищем последний элемент.
    last_node->next = new_node; // теперь последний элемент это новый узел
}
void print_list(){
    node* temp = root;
    do { std::cout << temp->data << " "; temp = temp->next; } while (temp != root);
}
void remove_node(float data){
    if(root->data == data){ // указатель на последний и корневой элемент;
        node *p_l = get_last(root); // Последний
        node *t = root->next;
        delete root;
        root = t; // теперь корневой это тот, который шел после корневого
        p_l->next = t;
        return;
    }

    node* temp = root;
    while(temp->next != root){
        if(temp->next->data == data){ // Если следующий элемент равен искомому
            node *ptr = temp->next->next;
            delete temp->next;
            temp->next = ptr;
            return;
        }
        temp = temp->next;
    }
    std::cout << "Value not found" << std::endl;
}
}

```

Рисунок 3 - Методы базовой работы со списком.

```

int main(){
    linked_list* list = new linked_list();
    list->insert_root_node(2);
    list->insert_last_node(3);
    list->insert_last_node(8);
    list->insert_last_node(6);
    list->insert_root_node(12);
    list->insert_root_node(123);
    list->remove_node(6);
    list->print_list();
    delete list;
}

```

Рисунок 4 - Тестирование линейного кольцевого списка.

Вывод: Я изучил методы организации списочных структур в динамической памяти.