



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)


О т ч е т

по лабораторной работе № 6

Название лабораторной работы: Списковые структуры

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-15Б


(Подпись, дата)

В.А. Бирюков
(И.О. Фамилия)

Преподаватель


(Подпись, дата)

О.А. Веселовская
(И.О. Фамилия)

Москва, 2024

Вариант 2

Цель: научиться работать с списковыми структурами

Задание: С клавиатуры вводятся целые числа. Организовать список с одновременным удалением повторяющихся чисел. Вывести на печать полученный список. Из полученного списка удалить максимальное число. Вывести на экран найденное число и новый список. При завершении программы освободить динамическую память.

Напишем код программы, вынеся добавление элементов в список, удаление наибольшего элемента, вывод списка и очищение памяти в подпрограммы для удобства

```
#include <iostream>
#include <Windows.h>

struct Node {
    int data;
    Node* next;
};

void addUniq(Node* &head, int value) {
    Node* current = head;
    Node* last = nullptr;
    bool flag_ununiq = false;
    while (current != nullptr) {
        if (current->data == value) {
            flag_ununiq = true;
        }
        last = current;
        current = current->next;
    }
    if (flag_ununiq == false) {
        Node* newNode = new Node{.data = value, .next = nullptr};
        if (last == nullptr) {
            head = newNode;
        } else {
            last->next = newNode;
        }
    }
}

void printList(Node* head) {
    while (head != nullptr) {
        std::cout << head->data << " ";
        head = head->next;
    }
    std::cout << std::endl;
}
```

Рисунок 1 – код программы, часть 1

```

void removeMax(Node* &head) {
    if (head == nullptr) {
        std::cout << "Список пустой" << std::endl;
    }
    else {
        Node* current = head;
        Node* previous = nullptr;
        Node* maxNode = head;
        Node* prevMax = nullptr;

        while (current != nullptr) {
            if (current -> data > maxNode -> data) {
                maxNode = current;
                prevMax = previous;
            }
            previous = current;
            current = current -> next;
        }

        int MaxValue = maxNode -> data;
        std::cout << "Максимальное значение: " << MaxValue << std::endl;
        if (maxNode == head) {
            head = head -> next;
        }
        else
        if (maxNode -> next == nullptr) {
            prevMax -> next = nullptr;
        }
        else {
            prevMax -> next = maxNode -> next;
        }
        delete maxNode;
    }
}

void clearList(Node* &head) {
    while (head != nullptr) {
        Node* current = head;
        head = head -> next;
        delete current;
    }
}

```

Рисунок 2 – код программы, часть 2

```

> int main(){
    SetConsoleOutputCP( wCodePageID: CP_UTF8);
    Node* head = nullptr;

    std::cout << "Введите целые числа (\\"end\\" для завершения ввода): " << std::endl;

    bool flag = true;
    std::string input;
    while (flag) {
        std::cin >> input;
        if (input == "end") {
            flag = false;
        } else {
            int value = std::stoi(input);
            addUniq(&head, value);
        }
    }

    std::cout << "Список без повторов: " << std::endl;
    printList(head);

    removeMax(&head);

    std::cout << "Список без максимального элемента: " << std::endl;
    printList(head);

    clearList(&head);
    return 0;
}

```

Рисунок 3 – код программы, часть 3

Протестируем программу, сделав несколько запусков с разными значениями

```

D:\Лабы\АиП\laba_6\сmake-build-debug\laba_6.exe
Введите целые числа ("end" для завершения ввода):
-1 -2 4 5 7 100 -2 4 52 0 35 0 end
Список без повторов:
-1 -2 4 5 7 100 52 0 35
Максимальное значение: 100
Список без максимального элемента:
-1 -2 4 5 7 52 0 35

```

Рисунок 4 – результат работы программы

```

D:\Лабы\АиП\laba_6\cmake-build-debug\laba_6.exe
Введите целые числа ("end" для завершения ввода):
1 end
Список без повторений:
1
Максимальное значение: 1
Список без максимального элемента:

```

Рисунок 5 – результат работы программы с одним введённым элементом

```

D:\Лабы\АиП\laba_6\cmake-build-debug\laba_6.exe
Введите целые числа ("end" для завершения ввода):
end
Список без повторений:

Список пустой
Список без максимального элемента:

```

Рисунок 6 – результат работы программы без введённых элементов

Видим, что программа корректно работает на введённых данных.

Составим схемы алгоритмов для программы и подпрограмм:

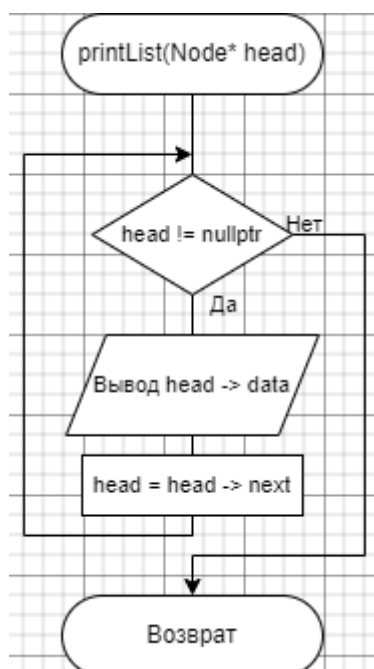


Рисунок 7 – схема процедуры для вывода списка

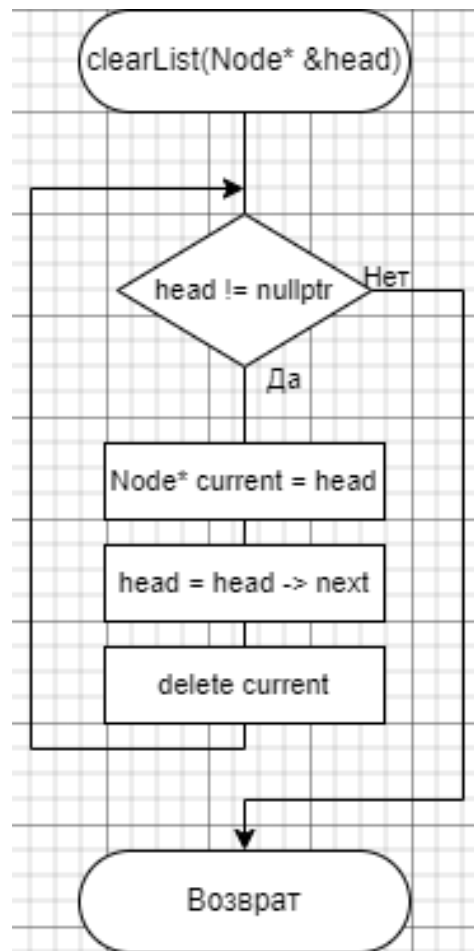


Рисунок 8 – схема алгоритма процедуры для освобождения памяти от списка

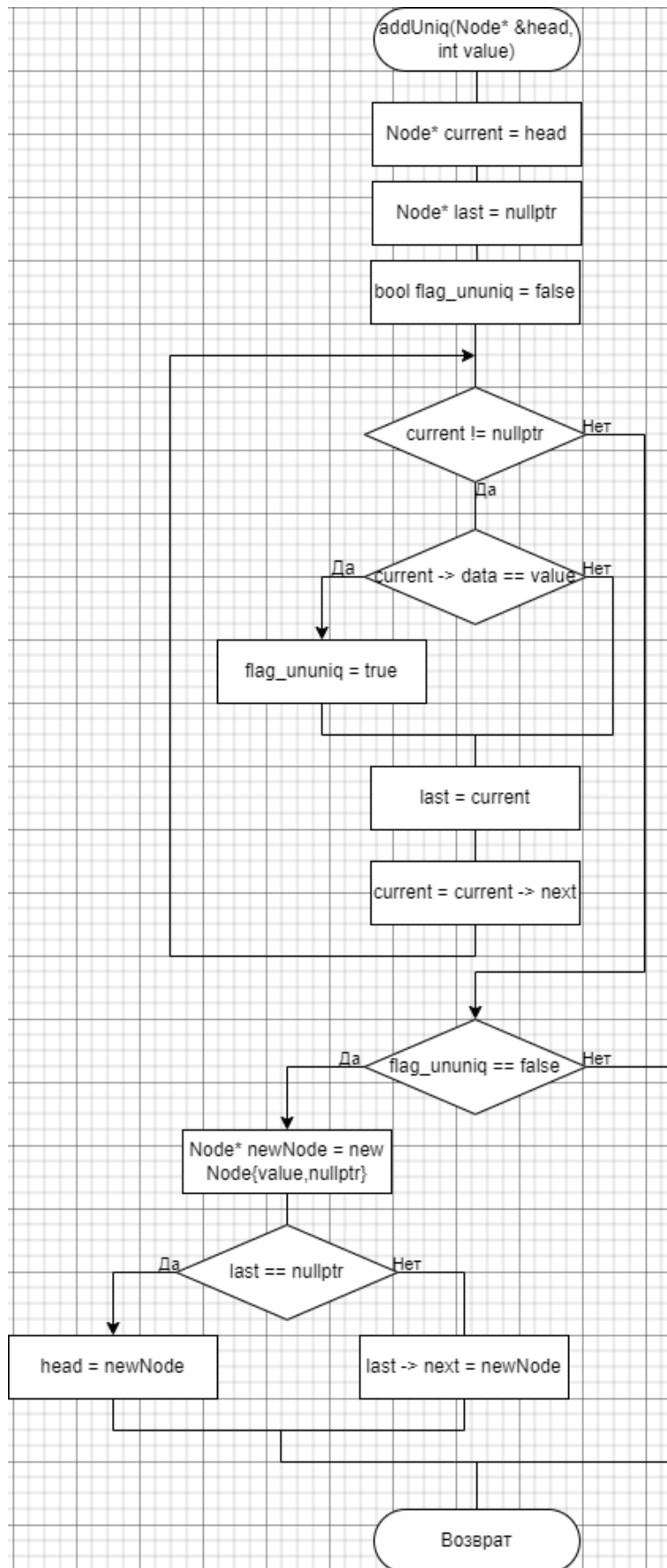


Рисунок 9 – схема алгоритма процедуры для добавления элементов в список

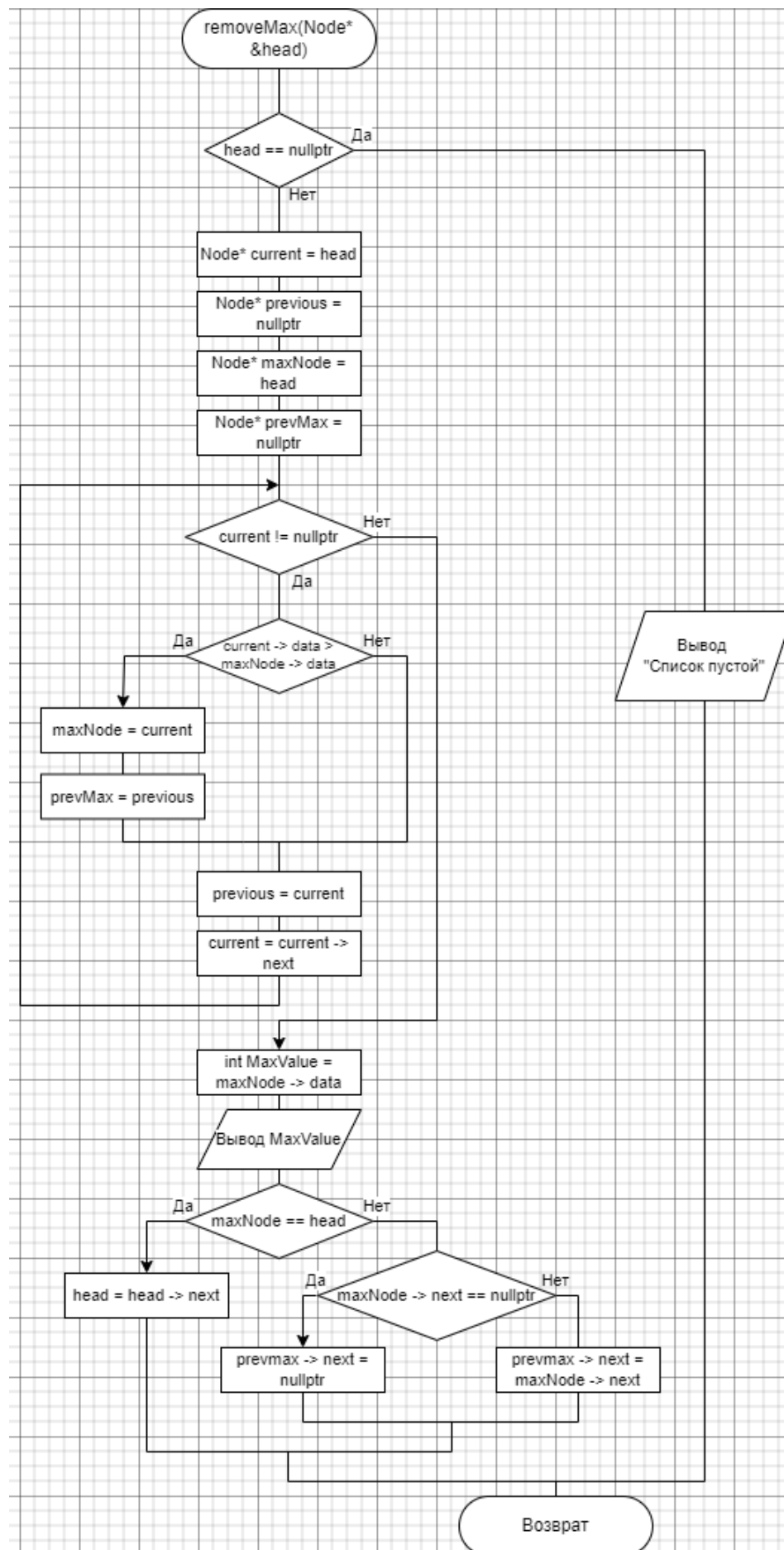


Рисунок 10 – схема алгоритма процедуры для удаления наибольшего элемента

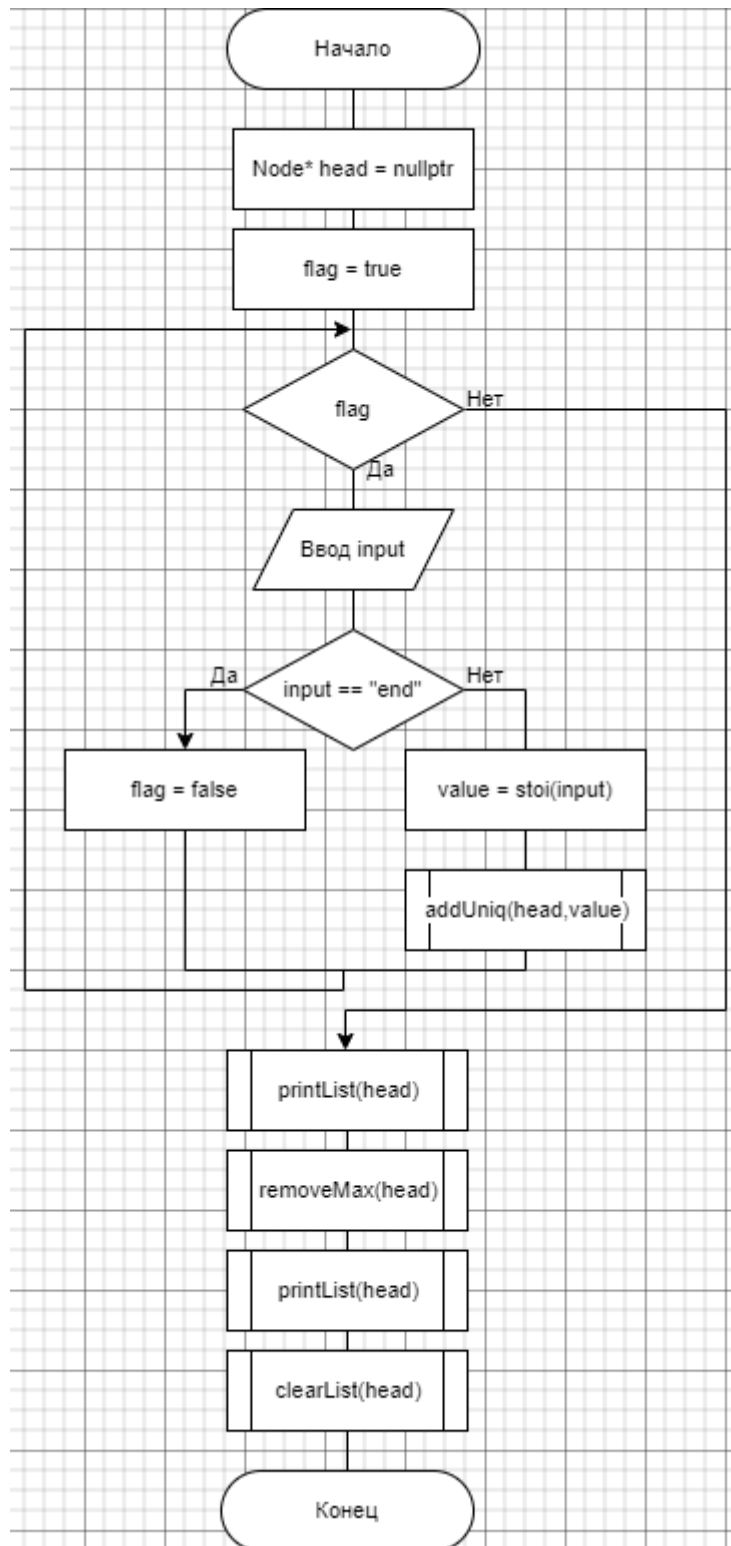


Рисунок 11 - схема алгоритма основной программы

Вывод: в ходе выполнения лабораторной работы я получил базовые навыки работы со списковыми структурами