

Практическая работа №4

Связные списки

Цель: Изучить структуру данных «связные списки» и реализовать её в программе на C++

Ход работы

Связный список – это структура данных, представляющая собой конечное множество упорядоченных элементов (узлов), связанных друг с другом посредством указателей, называется связным списком. Каждый элемент связного списка содержит поле с данными, а также указатель (ссылку) на следующий и/или предыдущий элемент. Эта структура позволяет эффективно выполнять операции добавления и удаления элементов для любой позиции в последовательности.

Структура DoubleList

```
struct DoubleList //описание узла списка
{
    int data; //информационное поле
    DoubleList *next; //указатель на следующий элемент
    DoubleList *prev; //указатель на предыдущий элемент
};
```

Добавление узла

					<i>АиСД.09.03.02.050000 ПР</i>					
Изм.	Лист	№ докум.	Подпись	Дата						
Разраб.		Ермошина В.А			Практическая работа №4 Линейные списки			Лит.	Лист	Листов
Провер.		Берёза А.Н.							2	
Реценз								ИСОиП (филиал) ДГТУ в г.Шахты ИСТ-Тб21		
Н. Контр.										
Утверд.										

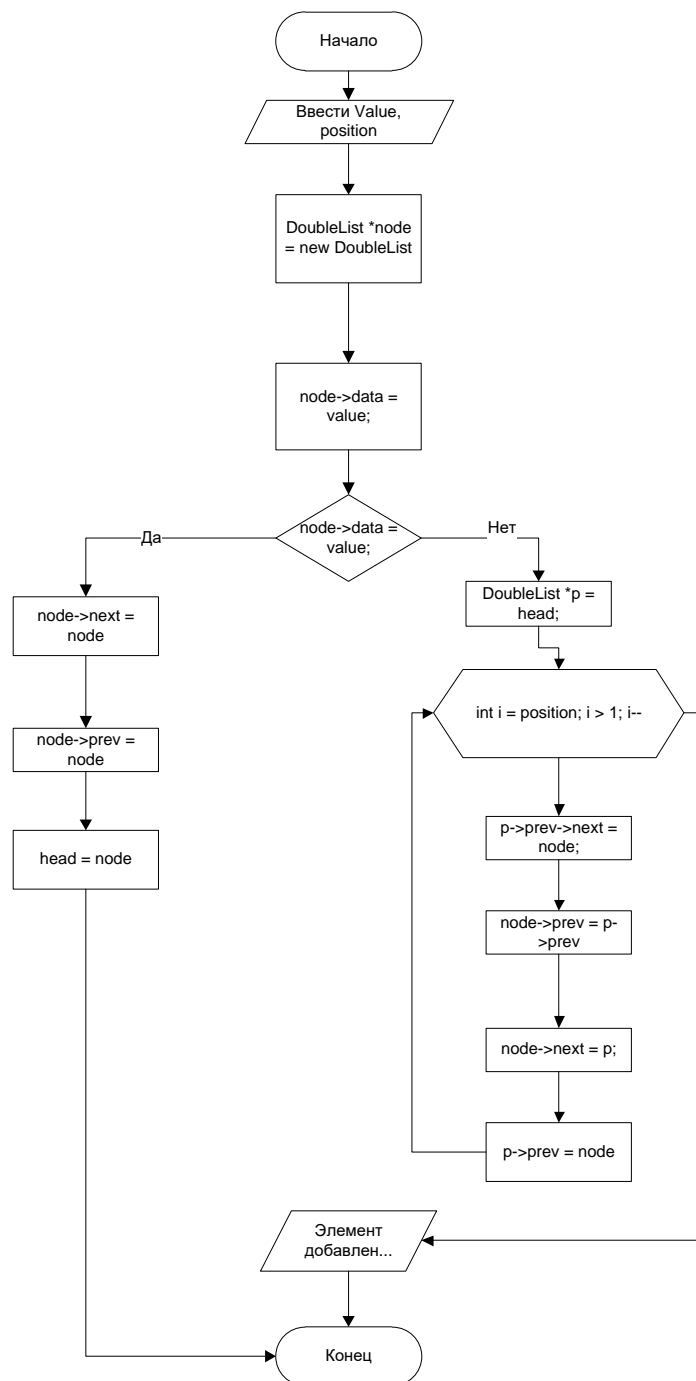


Рисунок 1 - Диаграмма для функции AddList

```

void AddList(int value, int position)
{
    DoubleList *node = new DoubleList; //создание нового элемента
    node->data = value; //присвоение элементу значения
    if (head == NULL) //если список пуст
    {
        node->next = node; //установка указателя next
        node->prev = node; //установка указателя prev
        head = node; //определяется голова списка
    }
    else
    {
        DoubleList *p = head;

```

```

    for (int i = position; i > 1; i--) p = p->next;
    p->prev->next = node;
    node->prev = p->prev;
    node->next = p;
    p->prev = node;
}
cout << "\nЭлемент добавлен...\n\n";
}

```

Удаление узла

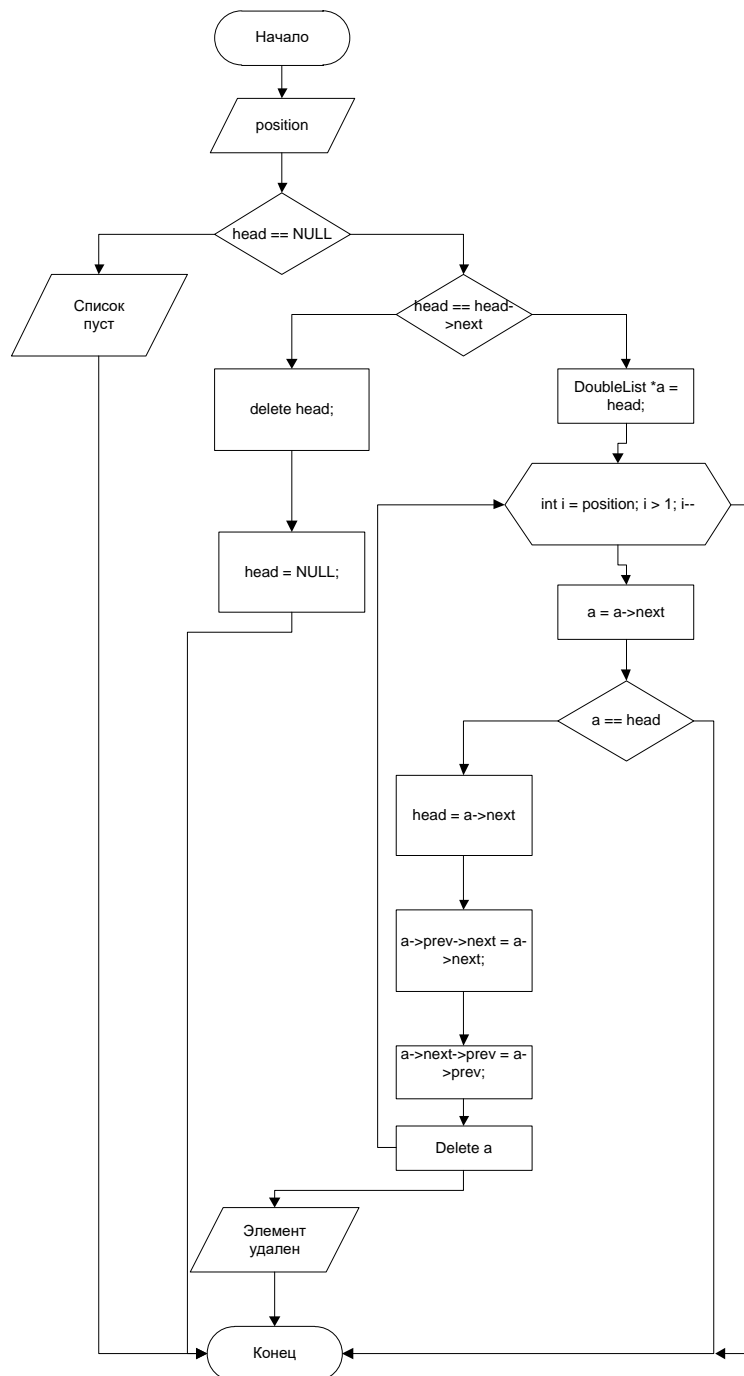


Рисунок 2 - Диаграмма для функции DeleteList

```

int DeleteList(int position)
{
    if (head == NULL) { cout << "\nСписок пуст\n\n"; return 0; }
}

```

```

if (head == head->next)
{
    delete head;
    head = NULL;
}
else
{
    DoubleList *a = head;
    for (int i = position; i > 1; i--) a = a->next;
    if (a == head) head = a->next;
    a->prev->next = a->next;
    a->next->prev = a->prev;
    delete a;
}
cout << "\nЭлемент удален...\n\n";
}

```

Вывод элементов списка

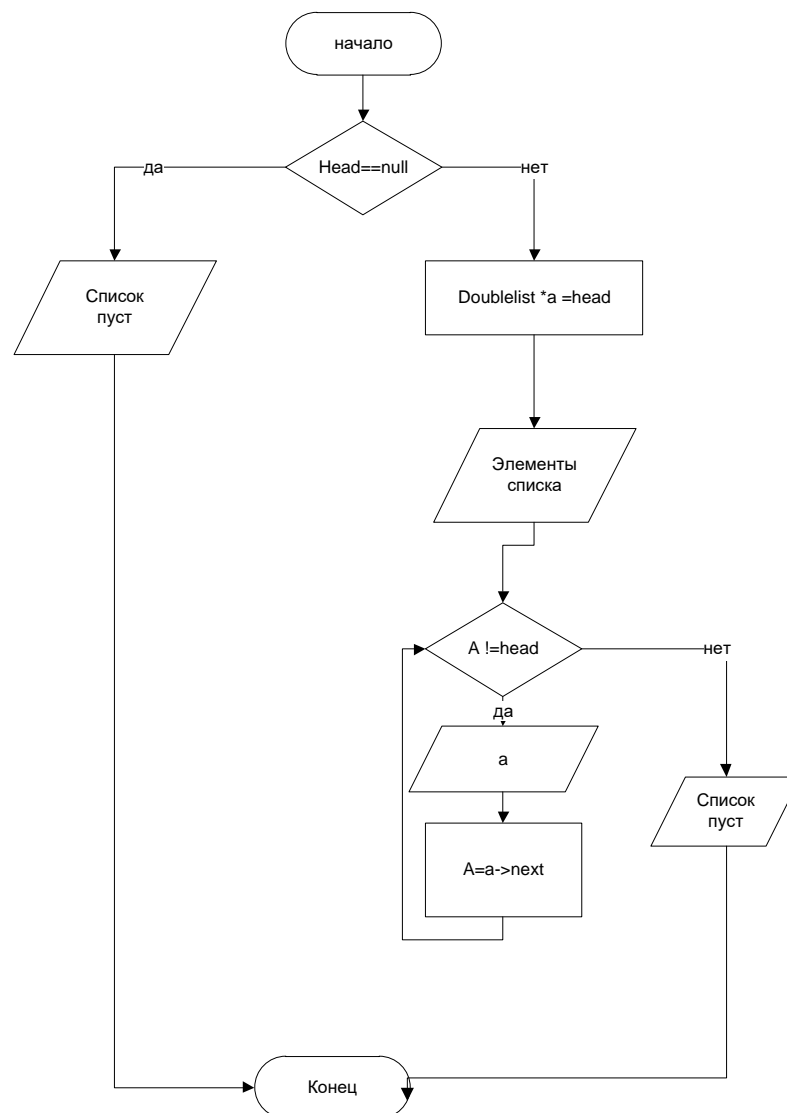


Рисунок 3 - Диаграмма для функции PrintList

```

void PrintList()
{

```

```

if (head == NULL) cout << "\nСписок пуст\n\n";
else
{
    DoubleList *a = head;
    cout << "\nЭлементы списка: ";
    do
    {
        cout << a->data << " ";
        a = a->next;
    } while (a != head); cout << "\n\n";
}
}

```

Вывод: В данной практической работе изучила сд «линейные списки» и реализовала её на языке C++

					<i>АиСД.09.03.02.050000 ПР</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		