

Лабораторная работа № 4

Тема: «Связный список»

Цель работы: Изучение связного списка

Основное назначение связного списка — предоставление механизма для хранения и доступа к произвольному количеству данных. Как следует из названия, это достигается связыванием данных вместе в список.

Список — это линейная динамическая структура данных, у каждого элемента может быть только один предок и только один потомок. По сути своей это очень похоже на обыкновенный массив, с той лишь разницей, что размер его не имеет ограничений. Списки также подразделяются на несколько типов.

Односвязный список — элемент имеет указатель только на своего потомка.

Двусвязный список — Каждый узел ДЛС содержит два поля указателей: на следующий и на предыдущий узел. Поле указателя на следующий узел последнего узла содержит нулевое значение (указывает на NULL). Поле указателя на предыдущий узел первого узла (корня списка) также содержит нулевое значение (указывает на NULL).

					АиСД 090000.000 ПР								
Изм	Лист	№ докум.	Подпись	Дата									
Разраб.		Климова Ю.В.			Лабораторная работа № 4 «Связный список»				Лит	Лист	Листов		
Провер.		Береза А.Н.									1	8	
									ИСОиП(ф)ДГТУ ИСТ-Tb21				
Н.контр.													
Утв.													

Программный код связного списка

```
#include <iostream>

using namespace std;

struct list { // структура списка

    int data; /*переменная, хранящая значение элемента*/

    list *next; /*ссылку на следующий элемент*/

};

void func(list *&begin, int num) {

    if (begin == nullptr) { // если список пуст, то мы создаём элемент в заголовке

        begin = new list();

        begin->data = num;

        begin->next = nullptr;

    }

    else {

        if (num <= begin->data)

        {

            list * new_elem = new list();

            new_elem->data = num;

            new_elem->next = begin;

            begin = new_elem;

        }

        else

        {

            list *ptr = begin;

            while (ptr->next && num >= ptr->next->data ) { // сравниваем

                ptr = ptr->next;
```

```

    }

    list * next = ptr->next;

    list * new_elem = new list();

    new_elem->data = num;

    ptr->next = new_elem;

    new_elem->next = next;

    }

}

}

void list_print(list *begin) {

    if (begin == nullptr) {

        cout << "Ошибка! Список пуст!" << endl;

        exit(1);

    }

    list *n = begin;

    while (n) {

        cout << n->data << " -> ";

        n = n->next;

    }

    cout << "NULL" << endl;

}

int main()

{

    setlocale(LC_ALL, "ru");

    list *begin = nullptr;

    int size;

    cout << "Введите размер списка: ";

```

					100000.000 IP	Лист
						3
Изм	Лист	№ докум.	Подпись	Дата		

```

cin >> size;

for (int i = 0; i < size; i++) {

    cout << "Введите элемент: ";

    int num;

    cin >> num;

    func(begin, num);

}

list_print(begin);

return 0;

}

```

Результат выполнения «Связный список»:

```

Введите размер списка: 5
Введите элемент: 17
Введите элемент: 56
Введите элемент: 14
Введите элемент: 92
Введите элемент: 34
14 -> 17 -> 34 -> 56 -> 92 -> NULL

```

Рисунок 2-Результат выполнения «Связный список»

					100000.000 ПР	Лист
						4
Изм	Лист	№ докум.	Подпись	Дата		

					100000.000 ПР	Лист
						5
Изм	Лист	№ докум.	Подпись	Дата		

```
#include <iostream>

using namespace std;

struct list { // структура списка

    int data; /*переменная, хранящая значение элемента*/

    list *next; /*ссылку на следующий элемент*/

};

void func(list *&begin, int num) {

    if (begin == nullptr) { // если список пуст, то мы создаём элемент в заголовке

        begin = new list();

        begin->data = num;

        begin->next = nullptr;
```

```

    }

    else {

        if (num <= begin->data)

        {

            list * new_elem = new list();

            new_elem->data = num;

            new_elem->next = begin;

            begin = new_elem;

        }

        else

        {

            list *ptr = begin;

            while (ptr->next && num >= ptr->next->data ) { // сравниваем

                ptr = ptr->next;

            }

            list * next = ptr->next;

            list * new_elem = new list();

            new_elem->data = num;

            ptr->next = new_elem;

            new_elem->next = next;

        }

    }

}

}

void list_print(list *begin) {

```

					100000.000 ИП	Лист
						7
Изм	Лист	№ докум.	Подпись	Дата		

```

if (begin == nullptr) {

    cout << "Ошибка! Список пуст!" << endl;

    exit(1);

}

list *n = begin;

while (n) {

    cout << n->data << " -> ";

    n = n->next;

}

cout << "NULL" << endl;

}

```

```

int main()

{

    setlocale(LC_ALL, "ru");

    list *begin = nullptr;


    int size;

    cout << "Введите размер списка: ";

    cin >> size;


    for (int i = 0; i < size; i++) {

        cout << "Введите элемент: ";

        int num;

        cin >> num;

        func(begin, num);

    }

```

					100000.000 ПР	Листм
						8
Изм	Лист	№ докум.	Подпись	Дата		


```
list_print(begin);
```

```
return 0;
```

```
}
```

					100000.000 IP	Лист
						9
Изм	Лист	№ докум.	Подпись	Дата		