

**Курс «Алгоритмы и алгоритмические языки»
1 семестр 2015/2016**

Лекция 16

Списки

- ◇ **Односвязный список** – это динамическая структура данных, каждый элемент которой содержит ссылку на следующий элемент (либо **NULL**, если следующего элемента нет).
- ◇ Доступ к списку осуществляется с помощью указателя на его первый элемент.

```
struct list {  
    struct data info;        /* Данные */  
    struct list *next;      /* Ссылка на след. элемент */  
};
```

- ◇ Выделение элемента

```
struct list *phead = NULL;  
phead = (struct list *) malloc (sizeof (struct list));
```

Списки

◆ Добавление элемента в начало

```
struct list *phead = NULL;
```

```
struct list *add_element (struct list *phead, struct  
                          data *elem) {  
    struct list *new = malloc (sizeof (struct list));  
    new->info = *elem;  
    new->next = phead;  
    return new;  
}
```

Списки

◆ Добавление элемента в конец

```
struct list *phead = NULL;
```

```
struct list *add_element (struct list *phead, struct  
                           data *elem) {
```

```
    if (! phead) {
```

```
        phead = malloc (sizeof (struct list));
```

```
        phead->info = *elem;
```

```
        phead->next = NULL;
```

```
        return phead;
```

```
    }
```

```
    struct list *ph = phead; // сохраним голову списка
```

```
    while (phead->next != NULL)
```

```
        phead = phead->next;
```

```
    phead->next = malloc (sizeof (struct list));
```

```
    phead->next->info = *elem;
```

```
    phead->next->next = NULL;
```

```
    return ph; // phead затерт, вернем сохраненный указатель
```

```
}
```

СПИСКИ

◆ Поиск элемента

```
struct list * phead;
```

```
int equals (struct data *, struct data *);
```

```
struct list * search (struct list *phead, struct data  
                    *elem) {
```

```
    while (phead && ! equals (&phead->info, elem))
```

```
        phead = phead->next;
```

```
    return phead;
```

```
}
```

СПИСКИ

◆ Удаление элемента

```
struct list *remove (struct list *phead,  
                    struct data *elem) {  
    struct list *prev = NULL, *ph = phead;  
    while (phead && ! equals (&phead->info, elem)) {  
        prev = phead;  
        phead = phead->next;  
    }  
    if (! phead)  
        return ph;  
    if (prev)  
        prev->next = phead->next;  
    else  
        ph = phead->next;  
    free (phead);  
    return ph;  
}
```

СПИСКИ

◇ Удаление элемента (двойной указатель)

```
void remove (struct list **pphead,  
             struct data *elem) {  
    struct list *prev = NULL, *phead = *pphead;  
    while (phead && ! equals (&phead->info, elem)) {  
        prev = phead;  
        phead = phead->next;  
    }  
    if (! phead)  
        return;  
    if (prev)  
        prev->next = phead->next;  
    else  
        *pphead = phead->next;  
    free (phead);  
}
```