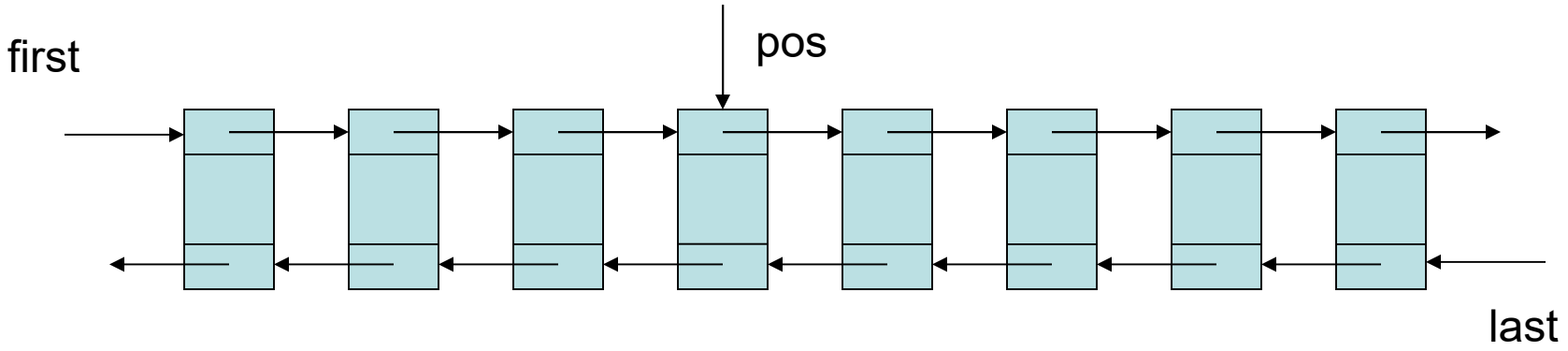
Danh sách liên kết đôi (Doubly Linked List)

Danh sách liên kết đôi

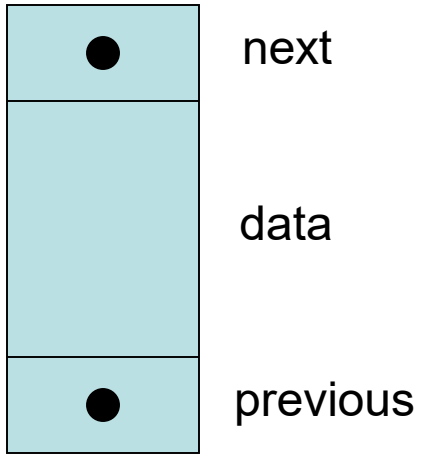
- Là danh sách mà mỗi phần tử có 2 mối liên kết:
  - next: để kết nối với phần tử kế tiếp
  - previous: để kết nối với phần tử trước nó



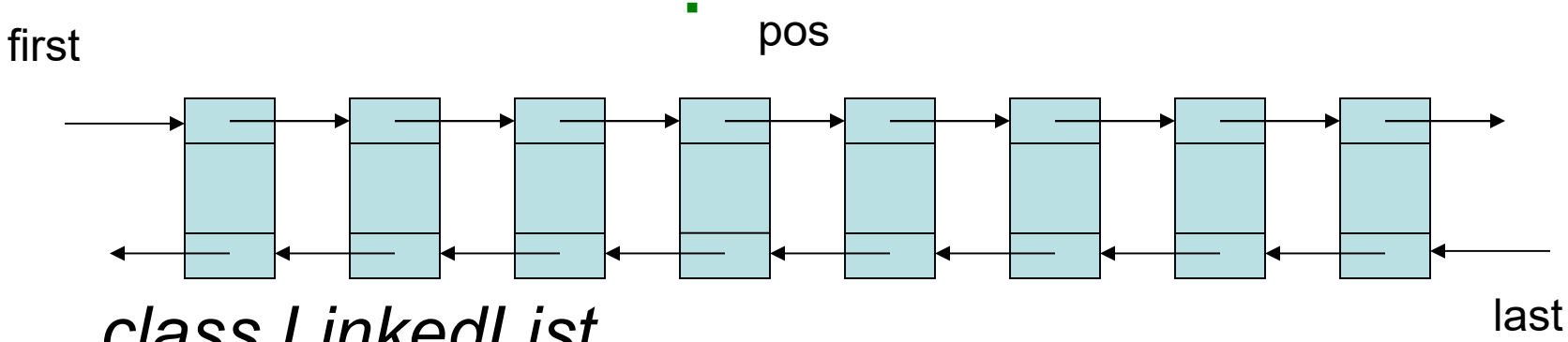
Cài đặt DSLK đôi

- Cài đặt: dựa trên con trỏ, bao gồm:
  - 3 con trỏ: first (đầu ds), pos (phần tử hiện hành), và last (cuối ds)
  - biến count: số phần tử của danh sách

```
class Link {
    public int dData;              // data
    public Link next;              // next link in list
    public Link previous;          // next link in list
    public Link(int d)  {          // constructor
        dData = d;
    }
}
```



Cài đặt DSLK đôi

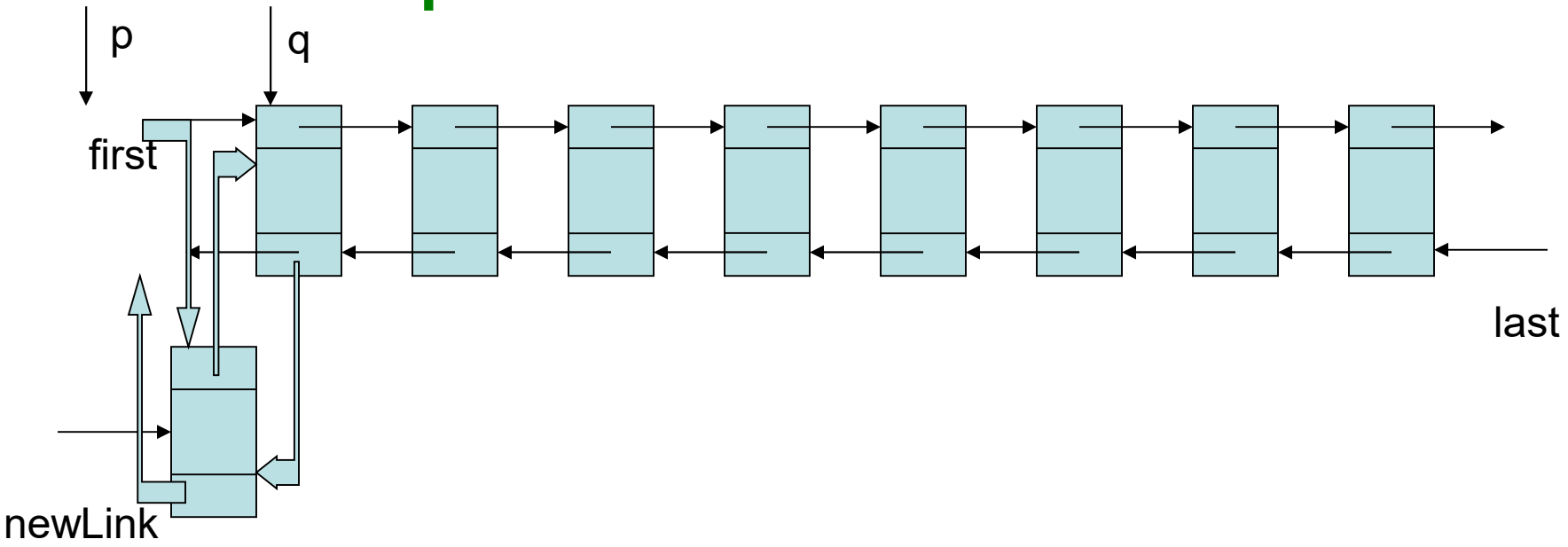



```
class LinkedList
{
    private Link first; // ref to first link
    private Link last; // ref to last link
    private Link pos; // ref to last link
    private int count;
        //==== Các method =====
}
```

Các thao tác trên danh sách liên kết đôi

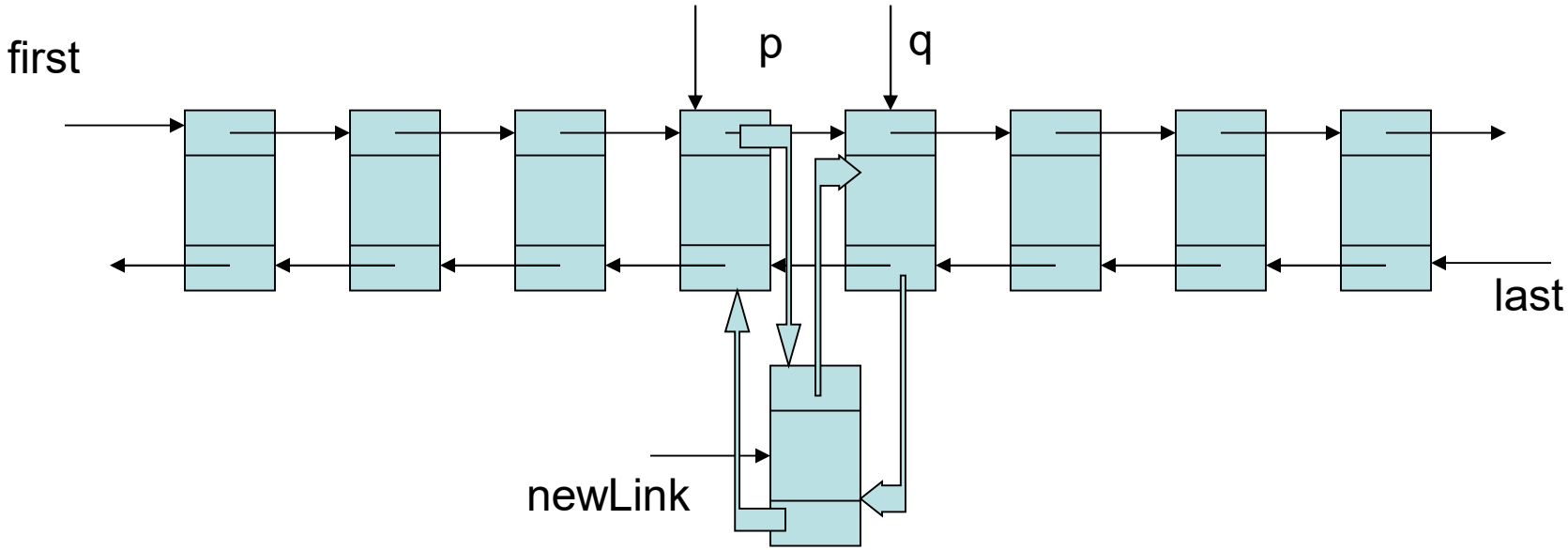
- Tương tự như danh sách liên kết đơn ngoại trừ 2 thao tác (cục bộ) làm thay đổi liên kết:
  - *Chèn phần tử vào danh sách*
  - *Xóa phần tử trong danh sách liên kết*
- Bổ sung thêm một số thao tác như:
  - *Khởi đầu từ cuối danh sách*
  - *Di chuyển qua phần tử trước phần tử hiện hành*

Chèn phần tử x vào danh sách



- Chèn đầu danh sách (xét theo chiều xuôi):
  - *q = first*
  - *newLink.next = q*
  - *first = newLink*

Chèn phần tử x vào danh sách



- Chèn sau p (xét theo chiều xuôi):
  - *q = p.next*
  - *newLink.next = q*
  - *p.next = newLink*

Thuật giải chèn phần tử

1. Cấp phát bộ nhớ cho *newp*, gán dữ liệu
2. Xác định con trỏ *q=(p==null? first: p.next)*
3. Kết nối xuôi

   3.1 newLink.next = q;

   3.2 Nếu p=null thì

   first = newLink

   3.3 Ngược lại

   p.next = newLink
4. Kết nối ngược

   4.1 newLink.previous = p;

   4.2 Nếu q=null thì

   last = newLink

   4.3 Ngược lại

   q.previous = newLink
5. Tăng biến count

Hàm chèn phần tử vào danh sách

```
private void insert(int dd, Link p)
{
    Link newLink, q;
    newLink = new Link(dd);     // make new link
    q = (p==null?  first:  p.next);
    newLink.next  =  q;
    if (p == null)
        first = newLink;
    else
        p.next = newLink;
    newLink.previous  =  p;
    if   (q==null)
        last  =  newLink;
    else
        q.previous  =  newLink;
    count++;
}
```

Hàm xóa phần tử trongdanh sách

```
private void delete(Link p)
{
    Link t, q;
    t = (p==null? first: p.next);
    q = t.next;
    if (p==null)
        first = q;
    else
        p.next = q;
    if (q==null)
        last = p;
    else
        q.previous = p;
    count--;
}
```

Bổ sung 2 hàm

- Khởi đầu tử cuối danh sách

```
public void startEnd()
{
    pos = last;
}
```

- Di chuyển đến phần tử trước phần tử hiện hành

```
void previousLink()
{
    if (pos == null)
        pos = last;
    else
        pos = pos.previous;
}
```

BÀI TẬP

- Thiết kế kiểu số nguyên lớn (bigint) với các phép toán: cộng, nhân. Áp dụng tính 100!, 17^{100}