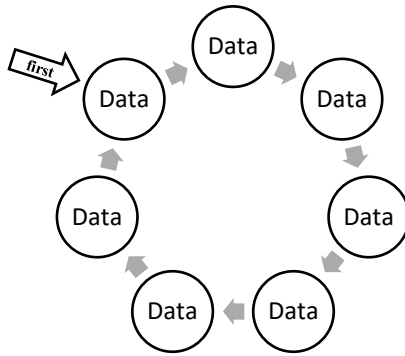
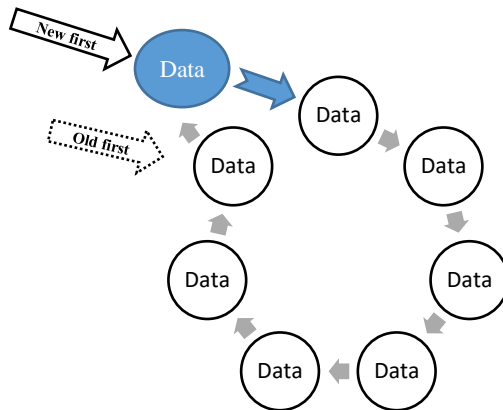


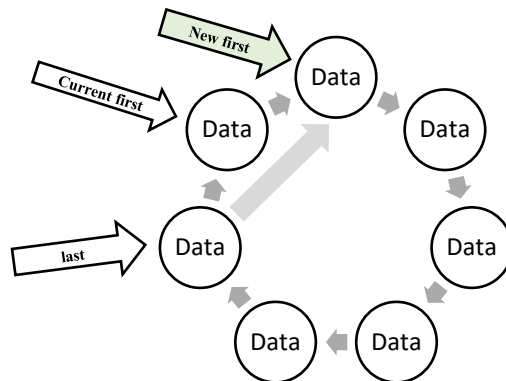
## class Ring



**first** сочи винаги последния елемент, който е бил четен/променян/добавян или следващия елемент, ако последното действие е било изтриване.



Добавяне на нов елемент: 1) запомняме стария **first** в променлива **p**; 2) създаваме нов елемент **first** от тип **Node**, като установяваме свойствата му (следващият му елемент е следващия на **p**); 3) следващият елемент на **p** е новият **first**.



Премахване на елемент: 1) „обикаляме“, за да намерим елемента преди първия (**last**); 2) следващия на **last** е следващия на текущия **first**; 3) **first** също става следващия на текущия **first**.

### Примерна реализация

```
public class Ring<T> {
    //Частен вътрешен клас „възел“
    private class Node{
        T Data;
        Node next;
    }
    private Node first;//Свойство, определящо първия елемент
    //Празен конструктор
    public Ring(){
        first=null;
    }
    //Конструктор по един елемент
    public Ring(T a){
        first=new Node();
        first.Data=a;
    }
}
```

```

    first.next=first; //Сочи към себе си
}
//Метод за достъп-четене до текущия елемент
public T getFirst() throws NullPointerException{
    if (first==null) throw new NullPointerException();
    return first.Data;
}
//Метод за достъп-промяна на текущия елемент
public void setFirst(T a) throws NullPointerException{
    if (first==null) throw new NullPointerException();
    first.Data=a;
}
public String toString(){
    String s="{ ";
    boolean b=false;
    Node p=first;
    while (p!=null){
        if (b) s=s + ",";
        s=s + p.Data.toString();
        b=true;
        p=p.next;
        if (p==first) break;
    }
    s=s + " ";
    return s;
}
//Добавяне на елемент
public void add(T a){
    Node p=first;
    first=new Node();
    first.Data=a;
    first.next=(p==null)?first:p.next;
    p.next=first;
}
//Премахване на елемент (линейно)
public void remove() throws NullPointerException{
    if (first==null) throw new NullPointerException();
    if (first.next==first) {first=null; return;}
    Node p=first, q;
    do{
        q=p;
        p=p.next;
    }while (p!=first);
    q.next=first.next;
    first=first.next;
}
}

```

---

#### Тестване на класа

```

public class Test {
    public static void main(String[] args) {
        Ring<Integer> r=new Ring<Integer>(4);
        r.add(3);
        r.add(7);
        for (int i=10;i<=20;i++) r.add(i);
        System.out.println(r.toString());
        do{
            try{
                r.remove();
                System.out.println(r.toString());
            } catch(Exception e){
                break;
            }
        }while (true);
    }
}

```