Lecture 5: ListNodes

07/01/22
New Topic: Linked Lists!

• Last week: **clients** of `LinkedList<E>`
  • `List<String> list = new LinkedList<>();`
  • `Queue<Integer> q = new LinkedList<>();`

• Today and next week: **implementers** of `LinkedList`
  • Today, Tuesday: `ListNodes`
  • Wednesday, Thursday, Friday: `LinkedIntList`
Memory for a list

- Contiguous in memory (array)

- Spread in memory
ListNode Pseudocode

- List: 42, -3, 17, 9

```c
struct ListNode {
    int data;
    struct ListNode* next;
}
```

Diagram:
- ListNode 0 (front) connected to 42
- 42 connected to -3
- -3 connected to 17
- 17 connected to 9
- 9 connected to null
Suppose we had the following `ListNodes`:

```
	p

q
```

What would the lists look like if we ran this line of code?

```
p.next = q.next;
```
Before

```
p -→ 1 -→ 2
```

```
q -→ 3 -→ 4 -→ null
```

```
p.next.next = q;
q = q.next;
p.next.next.next = null;
```
Announcements

• Monday, July 4th – No Lecture, IPL closed
• Checkpoint 2 due Wednesday, July 6th @ 11:59pm
• A2: GuitarHero due Thursday, July 7th @ 11:59pm