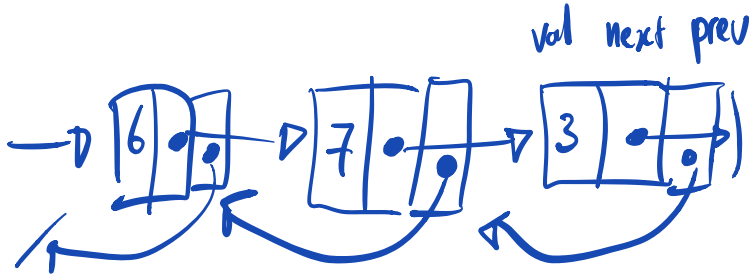


Linked List < int >



Generics

single has
val, next

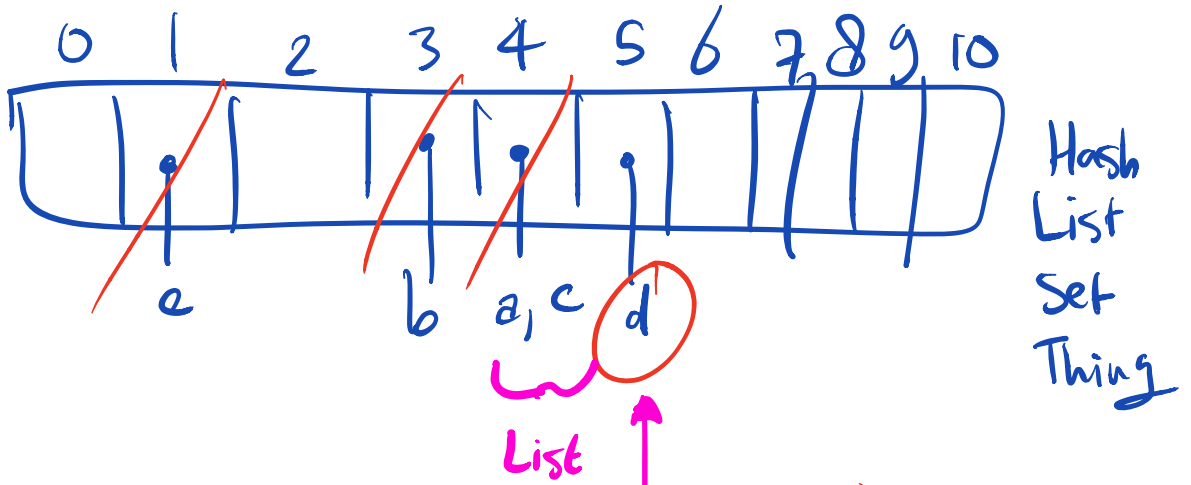
double has
val, next, prev

```
interface Giver < E > {  
    E get It ();  
}  
  
class Integer Giver implements Giver < Integer > {  
    public Integer get It ();  
}
```

Red arrows indicate the following relationships:

- An arrow from the `E` parameter in the `get It ()` method signature to the `Integer` parameter in the `implements Giver < Integer >` line.
- An arrow from the `Integer` parameter in the `implements Giver < Integer >` line to the `Integer` parameter in the `get It ()` method signature.
- An arrow from the `Integer` parameter in the `implements Giver < Integer >` line to the `Integer` parameter in the `get It ()` method signature.

Hashes



0...10

a	b	c	d	e	...
4	3	4	5	1	

f → 5

Maps and mappings

map type K onto value V
key value

$\text{Int} \rightarrow \text{String}$
 $\uparrow \quad \uparrow$
key value

$\text{Person} \rightarrow \text{Set} < \text{Person} >$

word \rightarrow description.
 $\downarrow \quad \downarrow$
 $\text{String} \rightarrow \text{String}$

You could even do:

```
class Word extends String {}
```

```
class Description extends String {}
```

```
Map <Word, Description> dictionary = ...
```