

# PROGRAMMAZIONE 2

## Pro Terza Esercitazione

# Il metodo get

---

- Si consideri un metodo che permette di ottenere il valore associato ad una chiave
- Signature
  - `public V get(K key)`
- Come viene specificato?

# Alcune specifiche

---

```
/** SPEC A
```

```
* @requires: key diverso da null & key è presente in this  
* @effects: restituisce il valore associato con la chiave key  
*/
```

```
/** SPEC B
```

```
* @require:s key è presente in this  
* @throws: NullPointerException se key è null  
* @effects: restituisce il valore associato con la chiave key  
*/
```

```
/** SPEC C
```

```
* @effects: restituisce il valore associato con la chiave key se key è presente in this  
*           oppure null se key non ha associato alcun valore in this  
*/
```

```
/** SPEC D
```

```
* @throws: NullPointerException se key è null  
* @effects: restituisce il valore associato con la chiave key  
*/
```

# Alcune implementazioni, 1

---

```
// implementazione 1
public V get(K key) {
    return values.get(indexOf(key));
}
```

```
// implementazione 2
public V get(K key) {
    if (key==null)
        throw new NullPointerException("null key to get");
    return values.get(indexOf(key));
}
```

# Alcune implementazioni, 2

---

```
// implementazione 3
public V get(K key) {
    if (key == null || indexOf(key) == -1)
        return null;
    return values.get(indexOf(key));
}
```

```
// implementazione 4
public V get(K key) {
    if (key == null)
        throw new NullPointerException("null key to get");
    if (indexOf(key) == -1)
        throw new NoSuchElementException("key not found");
    return values.get(indexOf(key));
}
```

# Verifica specifica

---



	IMPL1	IMPL 2	IMPL 3	IMPL 4
SPEC A				
SPEC B				
SPEC C				
SPEC D				

# Verifica specifica

---



	IMPL1	IMPL 2	IMPL 3	IMPL 4
SPEC A	OK			
SPEC B	OK	OK		
SPEC C	OK		OK	
SPEC D	OK	OK		OK