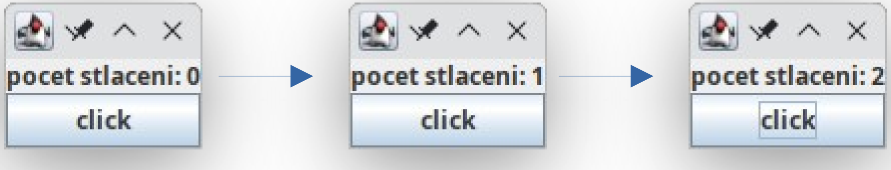
Objektovo orientované programovanie

8. cvičenie

Vladislav Novák

Vnorené triedy

- listener (observer) ako inštancia vnorenej triedy
- reakcia na stlačenie tlačidla



Object

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/Object.html

`protected Object clone()`	Creates and returns a copy of this object.
`public boolean equals(Object obj)`	Indicates whether some other object is "equal to" this one.
`public int hashCode()`	Returns a hash code value for the object.
`public final Class<?> getClass()`	Returns the runtime class of this Object.
`public String toString()`	Returns a string representation of the object.
`public final void notify()`	Wakes up a single thread that is waiting on this object's monitor.
`public final void notifyAll()`	Wakes up all threads that are waiting on this object's monitor.
`public final void wait()`	Causes the current thread to wait until it is awakened, typically by being notified or interrupted.
`public final void wait(long timeoutMillis)`	Causes the current thread to wait until it is awakened, typically by being notified or interrupted, or until a certain amount of real time has elapsed.
`public final void wait(long timeoutMillis, int nanos)`	Causes the current thread to wait until it is awakened, typically by being notified or interrupted, or until a certain amount of real time has elapsed.
`protected void finalize()`	Deprecated, for removal: This API element is subject to removal in a future version. Finalization is deprecated and subject to removal in a future release.

Metódy equals() a hashCode()

- ak definujeme, tak definovať oboje, aby boli konzistentné
- ak equals() vráti true, tak hashCode() oboch objektov musí byť rovnaké
- ak equals() vráti false, tak hashCode() oboch objektov môže ale nemusí byť rôzne

Comparable

```
public interface Comparable<T> {
    public int compareTo(T o);
}
```

<	záporná hodnota
=	nula
>	kladná hodnota

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/Comparable.html

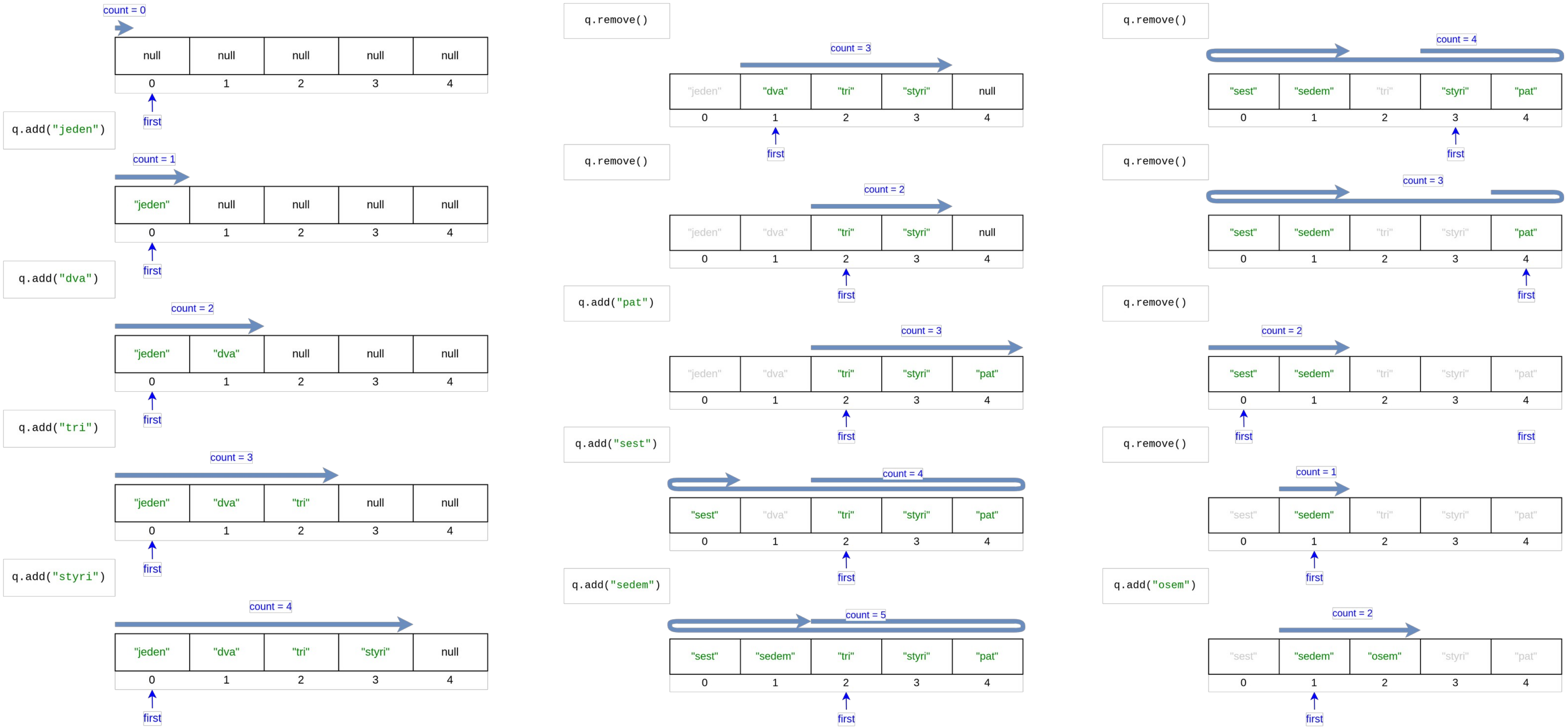
Comparator

```
public interface Comparator<T> {
    int compare(T o1, T o2);
    .
    .
    .
    .
    .
}
```

<	záporná hodnota
=	nula
>	kladná hodnota

https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Comparator.html

Generické typy - Queue



count = 0

null	null	null	null	null
0	1	2	3	4
first				

q.add("jeden")

count = 1

"jeden"	null	null	null	null
0	1	2	3	4
first				

q.add("dva")

count = 2

"jeden"	"dva"	null	null	null
0	1	2	3	4
first				

q.add("tri")

count = 3

"jeden"	"dva"	"tri"	null	null
0	1	2	3	4
first				

q.add("styri")

count = 4

"jeden"	"dva"	"tri"	"styri"	null
0	1	2	3	4
first				

q.remove()

count = 3

"jeden"	"dva"	"tri"	"styri"	null
0	1	2	3	4
	first			

q.remove()

count = 2

"jeden"	"dva"	"tri"	"styri"	null
0	1	2	3	4
		first		

q.add("pat")

count = 3

"jeden"	"dva"	"tri"	"styri"	"pat"
0	1	2	3	4
		first		

q.add("sest")

count = 4

"sest"	"dva"	"tri"	"styri"	"pat"
0	1	2	3	4
		first		

q.add("sedem")

count = 5

"sest"	"sedem"	"tri"	"styri"	"pat"
0	1	2	3	4
		first		

q.remove()

count = 4

"sest"	"sedem"	"tri"	"styri"	"pat"
0	1	2	3	4
			first	

q.remove()

count = 3

"sest"	"sedem"	"tri"	"styri"	"pat"
0	1	2	3	4
				first

q.remove()

count = 2

"sest"	"sedem"	"tri"	"styri"	"pat"
0	1	2	3	4
first				first

q.remove()

count = 1

"sest"	"sedem"	"tri"	"styri"	"pat"
0	1	2	3	4
	first			

q.add("osem")

count = 2

"sest"	"sedem"	"osem"	"styri"	"pat"
0	1	2	3	4
	first			

Generické typy - extends

trieda alebo rozhranie

```
class Person {
   private String name;

   public Person(String name) {
      this.name = name;
   }

   public String getName() {
      return name;
   }
}
```

```
class Employee extends Person {
   private int salary;

   public Employee(String name, int salary) {
      super(name);
      this.salary = salary;
   }

   public int getSalary() {
      return salary;
   }
}
```

```
class Box<TYPE extends Person> {
   private TYPE person;

   public void set(TYPE value) {
      person = value;
   }

   public TYPE get() {
      return person;
   }

   public String getName() {
      return person.getName();
   }
}
```

```
public static void main(String[] args) {
   Box<Employee> b = new Box<Employee>();
   b.set(new Employee("Laco", 2000));
   Employee e = b.get();
   String n = b.getName();
}
```

Genericke typy

```
public interface Map<KEY, VALUE> {

    int size();
    boolean isEmpty();
    boolean containsKey(Object key);
    .
    .
    .
    .
    .

}
```