Using or(..) operation in Predicate.

This is part 4 of Predicate Interface.

In previous posts we saw basics of Predicate Interface, how to filter the collection using predicate, how to join two different predicate and how to use negate() operation on predicate.

In this post we will see how use or(..) operation on two different predicates. This operation returns composed predicate that represents short-circuit logical or of this predicate and specified predicate.

We will again take example of Person class.

**package** com.example.javase8.filtercollections;

**public** **class** Person {

**private** String name;

**private** **int** age;

**public** Person(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "[name=" + name + ", age=" + age + "]";

}

}

Now let us insert some Person objects in ArrayList.

List<Person> list = **new** ArrayList<>();

list.add(**new** Person("Arya", 10));

list.add(**new** Person("Agon", 20));

list.add(**new** Person("Walder", 30));

list.add(**new** Person("Walda", 35));

list.add(**new** Person("Ramsay", 37));

list.add(**new** Person("Roy", 41));

Now let us define a predicate that will give us Person objects whose name starts with “A”.

Predicate<Person> predicate1 = **(p) -> p.getName().startsWith("A");**

Output

[name=Arya, age=10]

[name=Agon, age=20]

Now let us define another predicate that will give us Person objects whose name starts with “W”.

Predicate<Person> predicate2 = **(p) -> p.getName().startsWith("W");**

Output

[name=Walder, age=30]

[name=Walda, age=35]

Now we will use negate operation that will define short-circuit logical or between above two predicates.

Predicate<Person> predicate1 = **(p) -> p.getName().startsWith("A");**

Predicate<Person> predicate2 = **(p) -> p.getName().startsWith("W");**

Predicate<Person> predicate1or2 = **predicate1.or(predicate2);**

list.forEach(p -> {

**if** (predicate1or2.test(p)) {

System.***out***.println(p);

}

});

Output

[name=Arya, age=10]

[name=Agon, age=20]

[name=Walder, age=30]

[name=Walda, age=35]

This post was intended to make you understand how to or(..) method of predicate interface works.

In upcoming posts I will write about other features of java 8.