Using negate() operation in Predicate.

This is part 3 of Predicate Interface.

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

In this post we will see how to convert given Predicate to its logical negation.

Let us consider the old example of Person class. In previous post we saw the Person class. I will paste the code here for convenience.

**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 class.

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

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

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

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

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

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

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

list.add(**new** Person("Joshua", 50));

Now let us define a predicate that will give us Person objects with age less than or equal to 35.

Predicate<Person> predicate1 = **(p) -> p.getAge() <= 35;**

Now let us iterate through the collections and test the predicate against all objects in ArrayList.

list.forEach(p -> {

**if** (predicate1.test(p)) {

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

}

});

Output

[name=Ned, age=10]

[name=Ryan, age=20]

[name=James, age=30]

[name=Walder, age=35]

Now let us write negation of this predicate.

Predicate<Person> predicate1 = **(p) -> p.getAge() <= 35;**

Predicate<Person> notPredicate1 = **predicate1.negate();**

Now the output will be reversed i.e. age will be greater than 35.

list.forEach(p -> {

**if** (notPredicate1.test(p)) {

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

}

});

Output

[name=Roose, age=37]

[name=Roy, age=41]

[name=Joshua, age=50]

We can also use negate on even odd number too.

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

In next post we will see the or(..) method. It returns a composed predicate that represents the short-circuited and operation on either of the Predicates.