Java SE 8 Programming Language

Lab Guides

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

Hanoi, 06/2019

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1.	01/Oct/2018	Add the new labs	Create new	DieuNT1	VinhNV
2.	01/Jun/2019	Update template	Fsoft template	DieuNT1	VinhNV

Contents

CODE:	JPL.S.L501
TYPE:	SHORT
LOC:	
DURATION:	45 MINUTES

Unit 9: Collections, Streams, and Filters

Knowledge Summary

Java Stream

Java Stream is a sequence of elements from a source that supports aggregate operations. Streams do not store elements; the elements are computed on demand. Elements are consumed from data sources such as collections, arrays, or I/O resources.

Java Stream filter

Java Stream filter method is an intermediate operation, which returns elements of the stream that match the given predicate. A predicate is a function that returns a boolean value.

Java stream provides a method filter() to filter stream elements on the basis of given predicate. Suppose you want to get only even elements of your list then you can do this easily with the help of filter method.

This method takes predicate as an argument and returns a stream of consisting of resulted elements.

```
Stream<T> filter(Predicate<? super T> predicate)
```

predicate: Predicate is a functional interface. So, you can also pass lambda expression here.

Lab Guide 1: Use Java Stream Filter

Objectives:

This lab guide helps trainees know how to use Stream Filter in Java 8 in order to perform some operations:

- Use Streams filter(), findAny() and orElse()
- Filter multiple condition
- Use Streams filter() and map()
- Use filter() with matched patterns
- Use filter map by key
- Use filter map by values

Problem Descriptions:

Create a Java Project named **JPL.S.L501** in Eclipse.

Create package **fa.training.model** that contains:

- Person class

Create package **fa.training.streamfilterdemo** that contains:

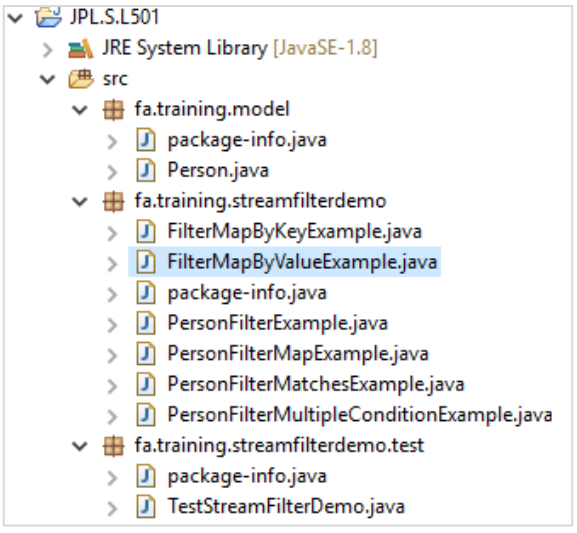
- PersonFilterExample class
- PersonFilterMultipleConditionExample class
- PersonFilterMapExample class
- PersonFilterMatchesExample class
- FilterMapByKeyExample class
- FilterMapByValueExample class

Create package **fa.training.streamfilterdemo.test** that contains:

- TestStreamFilterDemo class

Guidelines:

Step 1: Create a project struture like this:



Step 2: Create **Person** class:

```
1.  package fa.training.model;
2.
3.  /**
4.   * @author hoabt2
5.   *
6.   */
7.  public class Person {
8.
9.          private String name;
10.         private int age;
11.         private String email;
12.
13.
14. /**
15.  * The constructor with three paramters.
16.  */
17. public Person(String name, int age, String email) {
18.         super();
19.         this.name = name;
20.         this.age = age;
21.         this.email = email;
22. }
23.
24.     /**
25.      * The constructor with two paramters
26.      */
27. public Person(String name, int age) {
28.         super();
29.         this.name = name;
30.         this.age = age;
31. }
32. // getter and setter methods
33.
34. @Override
35. public String toString() {
36.         return "Person [name=" + name + ", age=" + age + "]";
37. }
38. }
39.
40.
```

Step 3: Create **PersonFilterExample** class

```
1.  package fa.training.streamfilterdemo;
2.
3.  import java.util.Arrays;
4.  import java.util.List;
5.
6.  import fa.training.model.Person;
7.
8.  /**
9.   * Examples of how to use stream filter.
10.  *
11.  * @author hoabt2
12.  *
13.  */
14. public class PersonFilterExample {
15.
16. /**
17.  * Find person by name (traditional way to iterate over a collection).
18.  *
19.  * @param personList the list of person
20.  * @param name the person name
21.  * @return an instance of Person
22.  */
```

```
23. private static Person findPersonByName(List<Person> personList, String name) {
24.         Person result = null;
25.         for (Person person : personList) {
26.                 if (name.equals(person.getName())) {
27.                         result = person;
28.                 }
29.         }
30.         return result;
31. }
32.
33. /**
34.  * Show person info
35.  */
36. public void showPerson() {
37.         System.out.println("Filter a collection using traditional method.");
38.         List<Person> personList = Arrays.asList(
39.                         new Person("Peter", 30),
40.                         new Person("Smith", 20),
41.                         new Person("Mary", 40));
42.
43.         Person result = findPersonByName(personList, "Peter");
44.         System.out.println(result);
45. }
46.
47. /**
48.  * Show person info using Stream Filter.
49.  *
50.  */
51. public void showPersonWithStreamFilter() {
52.         System.out.println("showPersonWithStreamFilter.");
53.
54.         List<Person> personList = Arrays.asList(
55.                         new Person("Peter", 30),
56.                         new Person("Smith", 20),
57.                         new Person("Mary", 40));
58.
59.         String name = "Peter";
60.         Person person1 = personList.stream() // Convert to steam
61.                         .filter(x -> name.equals(x.getName())) // get "Peter" only
62.                         .findAny() // If 'findAny' then return found
63.                         .orElse(null); // If not found, return null
64.
65.         System.out.println(person1);
66.
67.         Person person2 = personList.stream()
68.                         .filter(x -> "Tom".equals(x.getName()))
69.                         .findAny()
70.                         .orElse(null);
71.
72.         System.out.println(person2);
73. }
74. }
```

Step 4: Create **PersonFilterMultipleConditionExample** class

```
1.  package fa.training.streamfilterdemo;
2.
3.  import java.util.Arrays;
4.  import java.util.List;
5.
6.  import fa.training.model.Person;
7.
8.  /**
9.   * Examples of how to use filter multiple condition.
10.  *
11.  * @author hoabt2
```

```
 *
 */
public class PersonFilterMultipleConditionExample {

/**
 * Filter by multiple condition.
 *
 */
public void filterMultipleCondition() {

	System.out.println("filterMultipleCondition() !!!");

	List<Person> personList = Arrays.asList(
			new Person("Peter", 30),
			new Person("Smith", 20),
			new Person("Mary", 40));

	Person person1 = personList.stream()
		.filter((p) -> "Peter".equals(p.getName()) &&
					30 == p.getAge())
		.findAny()
		.orElse(null);

	System.out.println("Person 1 :" + person1);

	 //or like this
	 Person person2 = personList.stream()
		.filter(p -> {
		    if ("Peter".equals(p.getName()) && 30 == p.getAge()) {
		        return true;
		    }
		    return false;
		}).findAny().orElse(null);

	 System.out.println("Person 1 :" + person2);
}
}

```

Step 5: Create **PersonFilterMapExample** class

```
package fa.training.streamfilterdemo;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

import fa.training.model.Person;

/**
 * Examples of how to use filter and map.
 *
 * @author hoabt2
 *
 */
public class PersonFilterMapExample {

/**
 * Streams filter() and map()
 *
 */
public void filterAndMap() {
```

```
22.
23.         System.out.println("filterAndMap() !!!");
24.
25.         List<Person> personList = Arrays.asList(
26.                         new Person("Peter", 30),
27.                         new Person("Smith", 20),
28.                         new Person("Mary", 40));
29.
30.         String name = personList.stream()
31.                 .filter(x -> "Peter".equals(x.getName()))
32.                 .map(Person::getName)
33.                 .findAny()
34.                 .orElse("");
35.
36.          System.out.println("Name : " + name);
37.
38.          List<String> collect = personList.stream()
39.                 .map(Person::getName)
40.                 .collect(Collectors.toList());
41.
42.          collect.forEach(System.out::println);
43. }
44. }
```

Step 6: Create **PersonFilterMatchesExample** class

```
1.  package fa.training.streamfilterdemo;
2.
3.  import java.util.ArrayList;
4.  import java.util.List;
5.  import java.util.stream.Collectors;
6.
7.  import fa.training.model.Person;
8.
9.  /**
10.  * Examples of how to use filter with matched patterns
11.  *
12.  * @author hoabt2
13.  *
14.  */
15. public class PersonFilterMatchesExample {
16.
17. /**
18.  * Filter with matched patterns.
19.  *
20.  */
21. public void filterWithMatchPattern() {
22.         System.out.println("filterWithMatchPattern() !!!");
23.
24.         List<Person> personList = new ArrayList<>();
25.         Person person1 = new Person("Peter", 25, "peter13@gmail.com");
26.         Person person2 = new Person("Paul", 30, "paul1987@gmail.com");
27.         Person person3 = new Person("Tom", 70, "tom1234@gmail.com");
28.         Person person4 = new Person("Lion", 45, "lioncrone@gmail.com");
29.         personList.add(person1);
30.         personList.add(person2);
31.         personList.add(person3);
32.         personList.add(person4);
33.
34.         List<Person> result = personList.stream()
35.                         .filter(person -> person.getEmail().matches(".*gmail\\.com"))
36.                         .collect(Collectors.toList());
37.
38.         result.forEach(p -> System.out.println(p.getName()));
39. }
40. }
```

Step 7: Create **FilterMapByKeyExample** class

```
1.  package fa.training.streamfilterdemo;
2.
3.  import java.util.HashMap;
4.  import java.util.Map;
5.
6.  /**
7.   * Examples of using filter map by key.
8.   *
9.   * @author hoabt2
10.  *
11.  */
12. public class FilterMapByKeyExample {
13.
14. /**
15.  * Filter map by key.
16.  *
17.  */
18. public void filterMapByKey() {
19.         System.out.println("filterMapByKey() !!!");
20.
21.         Map<String, String> mapCountries = new HashMap<>();
22.
23.         mapCountries.put("de", "Germany");
24.         mapCountries.put("hu", "Hungary");
25.         mapCountries.put("sk", "Slovakia");
26.         mapCountries.put("si", "Slovenia");
27.         mapCountries.put("so", "Somalia");
28.         mapCountries.put("us", "United States");
29.         mapCountries.put("ru", "Russia");
30.
31.         mapCountries.entrySet().stream()
32.                 .filter(map -> map.getKey().startsWith("s"))
33.                 .forEach(m -> System.out.println(m));
34. }
35. }
36.
```

Step 8: Create **FilterMapByValueExample** class

```
1.  package fa.training.streamfilterdemo;
2.
3.  import java.util.HashMap;
4.  import java.util.Map;
5.
6.  /**
7.   * Examples of how to use filter map by values.
8.   *
9.   * @author hoabt2
10.  *
11.  */
12. public class FilterMapByValueExample {
13.
14. /**
15.  * Filter map by values.
16.  *
17.  */
18. public void filterMapByValues() {
19.         System.out.println("filterMapByValues() !!!");
20.
21.         Map<String, String> mapCountries = new HashMap<>();
22.
23.         mapCountries.put("de", "Germany");
24.         mapCountries.put("hu", "Hungary");
25.         mapCountries.put("sk", "Slovakia");
26.         mapCountries.put("si", "Slovenia");
27.         mapCountries.put("so", "Somalia");
```

```
28.         mapCountries.put("us", "United States");
29.         mapCountries.put("ru", "Russia");
30.
31.          mapCountries.entrySet().stream()
32.                      .filter(map -> map.getValue().equals("Slovakia")
33.                  || map.getValue().equals("Slovenia"))
34.              .forEach(m -> System.out.println(m));
35. }
36. }
37.
```

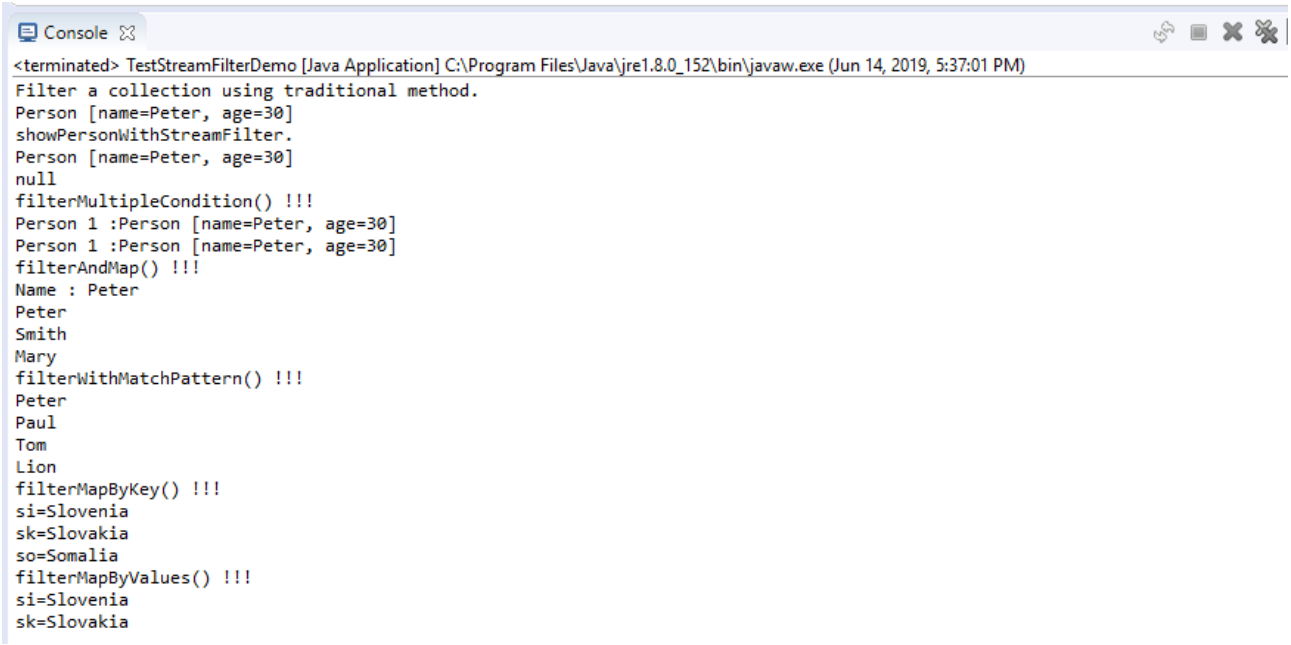
Step 9: Create **TestStreamFilterDemo** class

```
1.  package fa.training.streamfilterdemo.test;
2.
3.  import fa.training.streamfilterdemo.FilterMapByKeyExample;
4.  import fa.training.streamfilterdemo.FilterMapByValueExample;
5.  import fa.training.streamfilterdemo.PersonFilterExample;
6.  import fa.training.streamfilterdemo.PersonFilterMapExample;
7.  import fa.training.streamfilterdemo.PersonFilterMatchesExample;
8.  import fa.training.streamfilterdemo.PersonFilterMultipleConditionExample;
9.
10. /**
11.  * @author hoabt2
12.  *
13.  */
14. public class TestStreamFilterDemo {
15.
16. /**
17.  * @param args
18.  */
19. public static void main(String[] args) {
20.
21.         PersonFilterExample personFilter = new PersonFilterExample();
22.         PersonFilterMultipleConditionExample filterMultipleCondition =
23.                                         new PersonFilterMultipleConditionExample();
24.         PersonFilterMapExample filterMap = new PersonFilterMapExample();
25.         PersonFilterMatchesExample filterMatched = new PersonFilterMatchesExample();
26.         FilterMapByKeyExample filterMapByKey = new FilterMapByKeyExample();
27.         FilterMapByValueExample filterMapByValue = new FilterMapByValueExample();
28.         personFilter.showPerson();
29.         personFilter.showPersonWithStreamFilter();
30.         filterMultipleCondition.filterMultipleCondition();
31.         filterMap.filterAndMap();
32.         filterMatched.filterWithMatchPattern();
33.         filterMapByKey.filterMapByKey();
34.         filterMapByValue.filterMapByValues();
35. }
36. }
37.
```

Step 10: Run **TestStreamFilterDemo** to see the result

You can call corresponding methods separatedly in order to test the result clearly.

Result:



-- THE END --