JAVA LAMBDA ASSIGNMENT 8

1)

**interface** arithmetic

{

**int** operation(**int** a, **int** b);

}

**public** **class** lambda1

{

**public** **static** **void** main(String arg[])

{

arithmetic add = (a,b) -> (a+b);

System.***out***.println("addition of 1 and 1 is " +add.operation(1, 1));

arithmetic minus = (a,b) -> (a-b);

System.***out***.println("subtarction of 1 and 1 is " +minus.operation(1, 1));

arithmetic multiply = (a,b) -> (a\*b);

System.***out***.println("multiplication of 1 and 1 is " +multiply.operation(1, 1));

arithmetic divide = (a,b) -> (a/b);

System.***out***.println("division of 1 and 1 is " +divide.operation(1, 1));

}

}

2)

**public** **class** lambdaorder

{

**int** id;

**int** amt;

String status;

**public** lambdaorder(**int** id, **int** amt, String status)

{

**super**();

**this**.id = id;

**this**.amt = amt;

**this**.status = status;

}

**public** **int** getId()

{

**return** id;

}

**public** **void** setId(**int** id)

{

**this**.id = id;

}

**public** **int** getAmt()

{

**return** amt;

}

**public** **void** setAmt(**int** amt)

{

**this**.amt = amt;

}

**public** String getStatus()

{

**return** status;

}

**public** **void** setStatus(String status)

{

**this**.status = status;

}

}

-----------------------------------------------

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.stream.Stream;

**public** **class** order

{

**public** **static** **void** main(String arg[])

{

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

list.add(**new** lambdaorder(1, 20000, "completed"));

list.add(**new** lambdaorder(2, 15000, "completed"));

list.add(**new** lambdaorder(3, 30000, "accepted"));

list.add(**new** lambdaorder(4, 1000, "completed"));

list.add(**new** lambdaorder(5, 40000, "accepted"));

list.add(**new** lambdaorder(6, 35000, "completed"));

System.***out***.println("\n order having price more than 10000");

Stream<lambdaorder>data= list.stream().filter(p -> p.amt>10000);

data.forEach(lambdaorder -> System.***out***.println(lambdaorder.amt +" "+ lambdaorder.status));

System.***out***.println("\n accepted orders");

Stream<lambdaorder>data1= list.stream().filter(p -> p.getStatus().matches("accepted"));

data1.forEach(lambdaorder -> System.***out***.println(lambdaorder.amt +" "+ lambdaorder.id));

System.***out***.println("\n completed order");

Stream<lambdaorder>data2= list.stream().filter(p -> p.getStatus().matches("completed"));

data2.forEach(lambdaorder -> System.***out***.println(lambdaorder.amt +" "+ lambdaorder.id));

}

}

3)

Predicate

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** predicate

{

**public** **static** **void** main(String arg[])

{

List<Integer> list = Arrays.*asList*(1,2,3,4);

list.stream().filter(t -> t%2 == 0).forEach(t -> System.***out***.println("even value is : " +t));

}

}

Consumer

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.function.Consumer;

**public** **class** consumer

{

**public** **static** **void** main(String arg[])

{

Consumer<Integer> c1 = (t) -> System.***out***.println("number is " +t);

List<Integer> list = Arrays.*asList*(1,2,3,4);

list.stream().forEach(c1);

}

}

Supplier

**import** java.util.function.Supplier;

**public** **class** supplier

{

**public** **static** **void** main(String arg[])

{

Supplier<String> s1 = () -> "nikita";

System.***out***.println(s1.get());

}

}

4)

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** oddlength

{

**public** **static** **void** main(String arg[])

{

List<String> l1 = Arrays.*asList*("nikita", "kamal", "mithi", "pankaj");

l1.removeIf(t -> t.length()%2 != 0);

System.***out***.println("the name with even length are : " +l1);

}

}

5)

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** lambda5

{

**public** **static** **void** main(String arg[])

{

List<String> l1 = Arrays.*asList*("nikita", "pankaj", "kamal", "mithi");

StringBuilder str = **new** StringBuilder();

forEach(l1, t -> str.append(t.charAt(0)));

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

**public** **void** forEach(List<String> l1, Consumer<String> consumer)

{

**for**(String a : l1)

{

consumer.accept(a);

}

}

}

}

6)

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** lambda6

{

**public** **static** **void** main(String arg[])

{

List<String> l1 = Arrays.*asList*("nikita", "pankaj", "kamal", "mithi");

System.***out***.println(l1);

l1.replaceAll(t -> t.toUpperCase());

System.***out***.println(l1);

}

}

7)

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.Map.Entry;

**public** **class** lambda7

{

**public** **static** **void** main(String[] args)

{

Map<Integer, String> m = **new** HashMap<>();

m.put(1, "niki");

m.put(2, "mithi");

m.put(3, "pankaj");

m.put(4, "kamal");

StringBuilder str = **new** StringBuilder();

**for**(Map.Entry map:m.entrySet());

{

str.append(map.getKey());

str.append(map.getValue());

}

System.***out***.println("\n" +str+ "\n");

}

}

8)

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.function.Consumer;

**public** **class** lambda8 **extends** Thread

{

**public** **void** run()

{

System.***out***.println(" it is running...");

}

**public** **static** **void** main(String arg[])

{

lambda8 l = **new** lambda8();

l.start();

List<Integer> l1 = Arrays.*asList*(10,20,30,40);

Consumer<List<Integer>> print = list -> list.stream().forEach(a -> System.***out***.println(a));

System.***out***.println(l1);

}

}