

1.

**package** lambdaExpressions;

@FunctionalInterface

**interface** Arithmetic {

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

}

**public** **class** LambdaArtimeticOperations {

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

Arithmetic addition = (**int** a, **int** b) -> (a + b);

System.***out***.println("Addition = " + addition.operation(5, 6));

Arithmetic subtraction = (**int** a, **int** b) -> (a - b);

System.***out***.println("Subtraction = " + subtraction.operation(5, 3));

Arithmetic multiplication = (**int** a, **int** b) -> (a \* b);

System.***out***.println("Multiplication = " + multiplication.operation(4, 6));

Arithmetic division = (**int** a, **int** b) -> (a / b);

System.***out***.println("Division = " + division.operation(12, 6));

}

}

2.

package lambdaExpressions;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Stream;

class Product{

int id;

String name;

float price;

String status;

public Product(int id, String name, float price, String status) {

super();

this.id = id;

this.name = name;

this.price = price;

this.status = status;

}

}

public class LambdaExample2{

public static void main(String[] args) {

List<Product> list=new ArrayList<Product>();

list.add(new Product(1,"Samsung A5",7000f,"accepted"));

list.add(new Product(3,"Iphone 6S",60000f,"accepted"));

list.add(new Product(2,"Sony Xperia",15000f,"completed"));

list.add(new Product(4,"Nokia Lumia",8000f,"accepted"));

list.add(new Product(5,"Redmi4 ",26000f,"accepted"));

list.add(new Product(6,"Lenevo Vibe",17000f,"completed"));

Stream<Product> filtered\_data = list.stream().filter(p -> p.price > 10000);

filtered\_data.forEach(product -> System.out.println(product.name+": "+product.price));

Stream<Product> filtered\_data1 = list.stream().filter(c -> c.status ="accepted");

filtered\_data1.forEach(product -> System.out.println(product.name+": "+product.status));

}

}

3.consumer

**package** lambdaExpressions;

**import** java.util.ArrayList;

**import** java.util.List;

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

**public** **class** ConsumerInterfaceExample {

**static** **void** addList(List<Integer> list){

**int** result = list.stream().mapToInt(Integer::intValue).sum();

System.***out***.println("Sum of list values: "+result);

}

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

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

list.add(10);

list.add(20);

list.add(30);

list.add(40);

Consumer<List<Integer>> consumer = ConsumerInterfaceExample::*addList*;

consumer.accept(list);

}

}

3.functional

package lambdaExpressions;

import java.util.function.Function;

import java.util.List;

import java.util.ArrayList;

public class FunctionalInterfaceExample {

static Integer addList(List<Integer> list){

return list.stream()

.mapToInt(Integer::intValue)

.sum();

}

public static void main(String[] args) {

// Creating a list and adding values

List<Integer> list = new ArrayList<Integer>();

list.add(10);

list.add(20);

list.add(30);

list.add(40);

// Referring addList() method

Function<List<Integer>, Integer> fun = FunctionalInterfaceExample::addList;

// Calling Function interface method

int result = fun.apply(list);

System.out.println("Sum of list values: "+result);

}

}

3.predicate

**package** lambdaExpressions;

**import** java.util.function.Predicate;

**public** **class** PredicateInterfaceExample {

**static** Boolean checkAge(**int** age){

**if**(age>17)

**return** **true**;

**else** **return** **false**;

}

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

Predicate<Integer> predicate = PredicateInterfaceExample::*checkAge*;

**boolean** result = predicate.test(25);

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

}

}

4.

**package** lambdaExpressions;

**import** java.util.\*;

**public** **class** LambdaExampe4 {

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

{

ArrayList<String> Numbers = **new** ArrayList<String>();

Numbers.add("tfgcf");

Numbers.add("vhgchg");

Numbers.add("jvhgg");

Numbers.add("hjhmvg");

**for** (String i : Numbers) {

**if**(i.length()%2!=0 );

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

}

}

}

5.

package lambdaExpressions;

import java.util.Arrays;

import java.util.Objects;

import java.util.stream.Collectors;

public class LambdaExample5 {

public static void main(String[] args) {

String[] strArray = { "Red", "green", "Blue", "Yellow", "violet", "magenta", "pURPLE" };

System.out.println(Arrays.stream(strArray).filter(Objects::nonNull).map(s->s.charAt(0)).collect(Collectors.toList()));

}

}