## Java 8

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

**public** **class** J01\_ListIterate {

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

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

l.add(0);

l.add(10);

l.add(20);

l.add(5);

l.add(15);

l.add(25);

List<Integer> l1 = l.stream().filter(i->i%2==1).collect(Collectors.*toList*());

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

// for(Integer l1 :l) {

// if(l1%2==0) {

// System.out.println(l1);

// }

// }

}

}

//O/p - [5, 15, 25]

// .stream() -> Converts the List<Integer>(l) into a Stream<Integer>.

Streams are used to process collections efficiently.

// .filter(i -> i % 2 == 1) -> The filter() method is used to filter elements from the stream.

// i % 2 == 1 checks if a number is odd (i % 2 gives remainder 1 for odd numbers).

// So, this step keeps only the odd numbers: 5, 15, and 25.

// .collect(Collectors.toList()) -> Converts the filtered stream back into a List<Integer> (l1).

import java.util.\*;

import java.util.stream.Collectors;

//BY Java 8

class MultiplicationElement {

public static void main(String args[]) {

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

list.add(0);

list.add(10);

list.add(20);

list.add(5);

list.add(15);

list.add(25);

List<Integer> l=list.stream().map(i->i\*2).collect(Collectors.toList());

System.out.println(l);

}

}

// O/p - [0, 20, 40, 10, 30, 50]

public class J02\_Multiplecation {

public static void main(String args[]) {

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

// list.add(0);

// list.add(10);

// list.add(20);

// list.add(5);

// list.add(15);

// list.add(25);

//

// List<Integer> list1 = new ArrayList<>();

// for(Integer l : list) {

// list1.add(l\*2);

// }

//

// System.out.println(list);

// System.out.println(list1);

}

}

**.filter(i -> i.length() >= 9)**

* **filter()** is used to **select elements that satisfy a condition**.
* The condition **i.length() >= 9** checks if the string has **9 or more characters**.
* Only the strings with **9 or more** characters remain in the stream.

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

**public** **class** J04\_CountMethod {

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

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

list.add("Pavan");

list.add("Ramkrishanna");

list.add("Ramanuj");

list.add("Nagarjuna");

list.add("Chandrashekhar");

Long l = list.stream().filter(i->i.length()>=9).count();

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

}

}

// o/p

// 3

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

**import** java.util.stream.Collectors;

**class** Employee {

String name;

**int** salary;

**public** Employee(String name, **int** salary) {

**super**();

**this**.name = name;

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee [name=" + name + ", salary=" + salary + "]";

}

}

**public** **class** J05\_FilterEmployeeBySalary {

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

List<Employee> employees = Arrays.*asList*(

**new** Employee("Rama", 500),

**new** Employee("Aman" ,300),

**new** Employee("Jhon", 200)

);

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

// salary >=300

List<Employee> highsalary = employees.stream().filter(emp-> emp.salary >=300).collect(Collectors.*toList*());

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

}

}

// o/p -

// [Employee [name=Rama, salary=500], Employee [name=Aman, salary=300], Employee [name=Jhon, salary=200]]

// [Employee [name=Rama, salary=500], Employee [name=Aman, salary=300]]

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