**Java-8 – Predefined functional interfaces – Predicate Joining**

* **Predicate Joining:**

If you want to joining multiple predicates into a single result, we can use predicate joining.

P1: Given number is greater than 10?

P2: Is even number?

If you want to check given number is not greater than 10, then we can use negate().

P1.negate();

If you want to check the given number is greater than 10 and it is an even number, then we can use and().

P1.and(p2)

If you want to check either one of the condition, then we can use or().

P1.or(p2);

All the above methods are defined under Predicate interface as default methods.

* **Example:**

class PredicateJoiningDemo{

public static void main(String[] args){

int[] x = {0,5,10,15,20,25,30};

Predicate<Integer> p1= I -> I > 10;

Predicate<Integer> p2 = I -> I % 2 == 0;

System.out.println(“The numbers greater than 10 are:”);

m1(p1, x);

System.out.println(“The even numbers are:”);

m1(p2, x);

System.out.println(“The numbers not greater than 10 are:”);

m1(p1.negate(), x);

System.out.println(“The numbers greater than 10 and even are:”);

m1(p1.and(p2), x);

System.out.println(“The numbers greater than 10 or even are:”);

m1(p1.or(p2), x);

}

public static void m1(Predicate<Integer> p, int[] x){

for(int x1: x){

if(p.test(x1))

System.out.println(x1);

}

}

}