Ques - 1

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

Question:

Write a Java program that defines a method testPredicate to test a given number against a provided predicate. The program should include three predicates: isEven, isGreaterThanTen, and isPrime, each with different conditions. Your task is to implement the testPredicate method such that it accepts an integer and a predicate as parameters, and returns true if the given number satisfies the predicate, otherwise false.

Instructions:

Define a method testPredicate in the CalculationLogic class that accepts two parameters: an integer number and a predicate predicate of type Predicate<Integer>. This method should return a boolean value indicating whether the given number satisfies the provided predicate.

Define three predicates:

isEven: Checks if a given number is even.

isGreaterThanTen: Checks if a given number is greater than 10.

isPrime: Checks if a given number is prime.

Implement the testPredicate method such that it tests the given number against the provided predicate and returns the result.

In the main method:

Define a sample integer number (e.g., 13).

Test each predicate using the testPredicate method and print the result indicating whether the number satisfies each predicate.

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

**public** **class** CalculationLogic

{

**public** **boolean** testPredicate(**int** number,Predicate<Integer> p)

{

**return** p.test(number);

}

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

{

Predicate<Integer> isEven= a-> a%2==0;

Predicate<Integer> isGreater= a-> a>10;

Predicate<Integer> isPrime= a->

{

**int** c=0;

**for**(**int** i=1;i<=a;i++)

{

**if**(a%i==0)c++;

}

**if**(c==2)**return** **true**;

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

};

//--------------------------------------

CalculationLogic cl=**new** CalculationLogic();

**int** num=13;

System.***out***.println(num+" is ever --->"+cl.testPredicate(num, isEven));

System.***out***.println(num+" is Greater than "+10+" :"+cl.testPredicate(num, isGreater));

System.***out***.println(num+" is prime --->"+cl.testPredicate(num, isPrime));

}

}

**Output:-**

13 is ever --->false

13 is Greater than 10 :true

13 is prime --->true

Ques - 2

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

Question:

Create a Java program that defines a method modifyValue to perform modifications on an integer value using a provided consumer. The program should include three consumers: doubleValue, incrementBy, and squareValue, each with different modification operations. Your task is to implement the modifyValue method such that it accepts an integer value and a consumer as parameters, and applies the modification specified by the consumer to the value.

Instructions:

Define a method modifyValue in the ValueModifier class that accepts two parameters: an integer value value and a consumer consumer of type Consumer<Integer>. This method should apply the operation specified by the consumer to the value.

Define three consumers:

doubleValue: Doubles the value.

incrementBy: Increments the value by a given value increment.

squareValue: Squares the value.

Implement the modifyValue method such that it applies the operation specified by the consumer to the value.

In the main method:

Define a sample integer value (e.g., 5).

Test each consumer using the modifyValue method and print the modified value after each operation.

Example Output:

Original value: 5

After doubling the value: 10

After incrementing the value by 3: 8

After squaring the value: 64

Ques - 3

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

Question:

Create a Java program that defines a method applyFunction to apply a function to a given integer value. The program should include three functions: add, multiply, and subtract, each with different operations. Your task is to implement the applyFunction method such that it accepts an integer value and a function as parameters, and applies the operation specified by the function to the value.

Instructions:

Define a method applyFunction in the FunctionApplier class that accepts two parameters: an integer value value and a function function of type Function<Integer, Integer>. This method should apply the operation specified by the function to the value and return the result.

Define three functions:

add: Adds a given value increment to the input value.

multiply: Multiplies the input value by a given value factor.

subtract: Subtracts a given value decrement from the input value.

Implement the applyFunction method such that it applies the operation specified by the function to the value and returns the result.

In the main method:

Define a sample integer value (e.g., 10).

Test each function using the applyFunction method and print the result after each operation.

Example Output:

Original value: 10

After adding 5: 15

After multiplying by 2: 20

After subtracting 3: 7

Ques - 4

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

Question:

Create a Java program that demonstrates the usage of the Supplier interface to generate random numbers within a specified range. Your task is to implement a method generateRandomNumber in the NumberGenerator class that generates a random integer between a given minimum and maximum value (inclusive).

Instructions:

Define a method generateRandomNumber in the NumberGenerator class that accepts two parameters: min and max of type int, representing the minimum and maximum values for the range.

Implement the generateRandomNumber method such that it generates and returns a random integer between the specified min and max values (inclusive).

In the main method:

Prompt the user to enter the minimum and maximum values for the range.

Use the generateRandomNumber method to generate a random number within the specified range.

Print the generated random number.

Example Output:

Enter the minimum value: 5

Enter the maximum value: 10

Random number generated: 7