2. Prime Number

Problem: Write a Java program to check if a given number is prime.

Test Cases:

Input: 29

Output: true

Input: 15

Output:false  
  
EXPLANATION  
1. The PrimeNumber class contains a method DisplayPrime(int iNum) that checks if the given number iNum is prime or not.

2. The DisplayPrime method initializes a variable iCount to 0, and then iterates from 1 to iNum/2 to check if iNum is divisible by any number in that range.

3. If iNum is divisible by any number, the iCount is incremented. After the loop, if iCount is 1, it means iNum is prime, and the method returns true. Otherwise, it returns false.

4. The Question2 class contains the main method, which prompts the user to enter a number, creates a PrimeNumber object, and calls the DisplayPrime method to check if the input number is prime.

5. The result of the DisplayPrime method (either true or false) is stored in the bRet variable and then printed to the console.

The Mermaid flow diagram provides a detailed visual representation of the code's logic and flow. It includes the following key components:

1. The PrimeNumber class and its DisplayPrime method, which is responsible for the prime number checking logic.

2. The Question2 class and its main method, which handles the user input, creates the PrimeNumber object, and calls the DisplayPrime method to check the primality of the input number.3. The flow of the DisplayPrime method, including the initialization of iCount, the loop to check divisibility, and the final decision to return true or false based on the iCount value.4. The flow of the main method, including prompting the user for input, creating the PrimeNumber object, calling the DisplayPrime method, and displaying the result.



