```
import java.util.Scanner;
public class CRC {
public static int n;
public static void main(String[] args)
{
Scanner sc=new Scanner(System.in);
CRC crc=new CRC();
String copy,rec,code,zero="0000000000000000";
System.out.println("enter the dataword to be sent");
code=sc.nextLine();     // 1 0 0 1
n=code.length();
System.out.println("The dataword length is "+n);
copy=code;
code+=zero;          //1001 0000000000000000
System.out.println("The modified dataword is "+code);
code=crc.divide(code);
System.out.println("dataword="+copy);
copy=copy.substring(0,n)+code.substring(n);
System.out.print("CRC=");
System.out.println(code.substring(n));

System.out.println("transmitted frame is="+copy);
System.out.println("enter received data:");
rec=sc.nextLine();
if(zero.equals(crc.divide(rec).substring(n)))
System.out.println("correct bits received");
else
System.out.println("received frame contains one or more error");
sc.close();
}
public String divide(String s)
{
String div="10001000000100001";    // x^16+x^12+x^5+1     crc 16 bits
int i,j;
char x;
for(i=0;i<n;i++)          // outer loop
{
   x=s.charAt(i);
for(j=0;j<17;j++)         // inner loop
{
         if(x=='1')
{
         if(s.charAt(i+j)!=div.charAt(j))
       s=s.substring(0,i+j)+"1"+s.substring(i+j+1);      // 2 bits are same  result is 0 otherwise
                                                                  result is 1
 else
         s=s.substring(0,i+j)+"0"+s.substring(i+j+1);
}
}
}
return s;       // remainder
}
}
```

This Java program simulates a Cyclic Redundancy Check (CRC) for error detection in data transmission. The key components of the code include:

Input Handling:

It reads the dataword (the original message) from the user.

It appends a string of zeros (0000000000000000) to the dataword to create a codeword of appropriate length for CRC computation.

Division (CRC Calculation):

The divide() method uses a predefined divisor (10001000000100001), which is the CRC-16-IBM polynomial.

It performs bitwise XOR division between the codeword and divisor and modifies the codeword with the remainder.

Transmission:

After CRC calculation, it prints the CRC and the transmitted frame (original dataword with CRC appended).

Error Detection:

The program reads the received frame and performs the same CRC division.

If the remainder is all zeros, the frame is correct; otherwise, it has one or more errors.

Key Considerations:

Polynomials in CRC:

The div string is a generator polynomial. The specific value here corresponds to the CRC-16-IBM polynomial.

Bitwise XOR:

The division process essentially mimics a bitwise XOR, replacing parts of the codeword based on the division results.

Explanation for the program:

1. Scanner sc=new Scanner(System.in);

This creates an instance of the Scanner class, which allows the program to take user input from the console (standard input).

The Scanner object sc will be used to read strings or numbers typed by the user during the execution of the program. In this case, it is used to get the dataword and received frame as inputs.

2. CRC crc=new CRC();

This creates an instance of the CRC class, enabling you to call the non-static methods of the class, such as divide().

In this case, the CRC object crc will be used to invoke the method crc.divide() to perform the division operation required for CRC calculation.