```java

//   TestFib.java
import java.util.*;

public class TestFib
{
    public static void main(String[] args)
    {
    int n, fib1, fib2;
     Scanner conIn = new Scanner(System.in);

    System.out.println ();
    System.out.print ("Please enter a positive integer: ");

    if (conIn.hasNextInt())
      n = conIn.nextInt();
    else
        {
          System.out.println("Error: you must enter an integer.");
          System.out.println("Terminating program.");
          return;
        }
    System.out.println();

    fib2 = Fib.fib2(n);
    System.out.println("Fib2(" + n + ") is " + fib2);

    fib1 = Fib.fib1(n);
    System.out.println("Fib1(" + n + ") is " + fib1);

      System.out.println("Fib2(14) took " + nanoTime1() / 1000000.0 +
      " seconds to complete.");

      System.out.println("Fib1(14) method took " + nanoTime1() / 1000000.0 +
      " seconds to complete.");


    }

    // Determines duration of time to execute code.
    // Returns the current value of the most precise available system timer, in nanoseconds.
    public static long nanoTime1() {
        long startTime = System.nanoTime();

        for(int x = 0; x < 15; x++) {
            System.out.println(Fib.fib1(x));
        }

        long elapsedTime = System.nanoTime() - startTime;
        return elapsedTime;
    }

    // Determines duration of time to execute code.
    // Returns the current value of the most precise available system timer, in nanoseconds.
    public static long nanoTime2() {
        long startTime = System.nanoTime();

```

```
         for(int y = 0; y < 15; y++) { System.out.println(Fib.fib2(y)); }

        long elapsedTime = System.nanoTime() - startTime;
        return elapsedTime;
    }
}
```