

L12

FibonacciTest.java

```
import java.util.*;

public class FibonacciTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter an integer or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q"))
                done = true;
            else {
                int myLong = Integer.parseInt(answer);
                long startTime = System.currentTimeMillis();
                long fibN = fibonacci(myLong);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime) / 1000.0;
                System.out.print("Fibonacci(" + myLong + ") = " + fibN);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static long fibonacci(long n) {
        if (n <= 2) {
            if (n == 0)
                return (0);
            else
                return (1);
        } else
```

```
        return (fibonacci(n - 1) + fibonacci(n - 2));
    }
}
```

ExponentTest.java

```
import java.util.Scanner;

public class ExponentTest {
    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter an integer a then n to compute a^n or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q"))
                done = true;
            else {
                int a = Integer.parseInt(answer);
                System.out.print("Enter an integer n: ");
                answer = myScanner.nextLine();
                int n = Integer.parseInt(answer);
                long startTime = System.currentTimeMillis();
                long exp = exponent(a, n);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime) / 1000.0;
                System.out.print("exponent(" + a + ", " + n + ") = " + exp);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static int exponent(int a, int n) {
        if (n > 1) {
            return a * exponent(a, n - 1);
        } else {
            return a;
        }
    }
}
```

```
}  
}  
}
```

Reverse.java

```
import java.util.Scanner;  
  
public class Reverse {  
    public static void main(String[] args) {  
        Scanner myScanner = new Scanner(System.in);  
        boolean done = false;  
        while (!done) {  
            System.out.print("Enter a string to reverse or Q to quit: ");  
            String answer = myScanner.nextLine();  
            if (answer.equalsIgnoreCase("q"))  
                done = true;  
            else {  
                long startTime = System.currentTimeMillis();  
                String rev = reverse(answer);  
                long endTime = System.currentTimeMillis();  
                double elapsedTime = (endTime - startTime) / 1000.0;  
                System.out.print("reverse(" + answer + ") = " + rev);  
                System.out.println(" took " + elapsedTime + " seconds");  
            }  
        }  
        System.out.println("Goodbye!");  
    }  
  
    public static String reverse(String s) {  
        if (s.length() > 0)  
            return s.charAt(s.length() - 1) + reverse(s.substring(0, s.length() - 1));  
        else  
            return "";  
    }  
}
```

Revision #1

Created 24 April 2025 22:21:37 by Brandon Duke

Updated 24 April 2025 22:21:42 by Brandon Duke