Package com.kevin

Class FibonnaciRecorder

java.lang.Object
    com.kevin.FibonnaciRecorder

```
public class FibonnaciRecorder
extends java.lang.Object
```

Author:

Kevin Lai

Constructor Summary

Constructors

Constructor	Description
`FibonnaciRecorder()`	

Method Summary

All Methods Static Methods Concrete Methods

Modifier and Type	Method	Description
`static void`	`createChart(java.util.ArrayList<java.lang.Long> iterativeTimes, java.util.ArrayList<java.lang.Long> recursiveTimes)`	Using this method exports a line chart to a .png file with results from the sequences.
`static int`	`fibonacciIterative(int n)`	This is the iterative implementation of the Fibonacci sequence.
`static int`	`fibonacciRecursive(int n)`	This is the recursive

Modifier and Type	Method	Description
		implementation of the Fibonacci sequence.
`static void`	`main(java.lang.String[] args)`	The main method stores data from the Fibonacci methods as well as the times needed to execute them.
`static void`	`recordData(java.util.ArrayList<java.lang.Long> iterativeTimes, java.util.ArrayList<java.lang.Long> recursiveTimes, java.util.ArrayList<java.lang.Integer> results)`	This method exports a text file with results from the sequence, with the Fibonacci sequence itself as refrence.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructor Details

FibonnaciRecorder

```
public FibonnaciRecorder()
```

Method Details

main

```
public static void main(java.lang.String[] args)
```

The main method stores data from the Fibonacci methods as well as the times needed to execute them. It also will store the actual Fibonacci sequence to pass to the `recordData (ArrayList, ArrayList, ArrayList)` method.

createChart

```
public static void createChart(
java.util.ArrayList<java.lang.Long> iterativeTimes,
java.util.ArrayList<java.lang.Long> recursiveTimes)
```

Using this method exports a line chart to a .png file with results from the sequences. The x axis represents the number of terms used, and the y axis represents time taken to execute the method in nanoseconds.

Parameters:

`iterativeTimes` - The time data for the iterative method.

`recursiveTimes` - The time data for the recursive method.

recordData

```
public static void recordData(
java.util.ArrayList<java.lang.Long> iterativeTimes,
java.util.ArrayList<java.lang.Long> recursiveTimes,
java.util.ArrayList<java.lang.Integer> results)
```

This method exports a text file with results from the sequence, with the Fibonacci sequence itself as refrence.

Parameters:

`iterativeTimes` - The time data for the iterative method.

`recursiveTimes` - The time data for the recursive method.

`results` - The Fibonacci sequence, recorded by using the iterative method.

fibonacciIterative

```
public static int fibonacciIterative(int n)
```

This is the iterative implementation of the Fibonacci sequence.

Parameters:

`n` - The number of terms to be used.

Returns:

Returns the nth number in the sequence.

fibonacciRecursive

```
public static int fibonacciRecursive(int n)
```

This is the recursive implementation of the Fibonacci sequence.

Parameters:

`n` - The number of terms to be used.

Returns:

Returns the nth number in the sequence.