For: Melissa Bakke

Assignment: Exercise 22.8 All prime numbers up to 1,000,000

|  |
| --- |
| **Screenshot(s)** |
|  |
|  |

|  |
| --- |
| **Code** |
| *import java.io.RandomAccessFile;*  */\*\**  *\* Class: Exercise 228*  *\* Developer: Melissa Bakke*  *\* Date: 03/02/2017*  *\* Purpose: Program that finds all prime numbers up to 1,000,000.*  *\*/*  *public class Exercise228 {*  *final static int ARRAY\_SIZE = 10000;*    */\*\**  *\* @param args the command line arguments*  *\*/*  *public static void main(String[] args) throws Exception {*  *//TestFileClass.printFile();*  *// declare variables*  *final long N = 1000000; // how many numbers are tested for primeness*  *long[] primeNumbers = new long[ARRAY\_SIZE];*  *int squareRoot = 1;*  *long number; // the number tested for primeness*    *// check file for starting point, if empty start at 1*  *// if not empty, start at last entry*  *RandomAccessFile inout = new RandomAccessFile("PrimeNumbers.dat", "rw");*  *if (inout.length() == 0) {*  *number = 1;*  *}*  *else {*  *inout.seek(inout.length() - 8); // find last number in file*  *number = inout.readLong(); // read last number*  *}*    *// repeatedly find prime numbers*  *newNumber:while (number <= N) {*  *// check to see if number++ is prime*  *number++;*    *inout.seek(0);*    *if ((squareRoot \* squareRoot) < number) {*  *squareRoot++;*  *}*    *while (inout.getFilePointer() < inout.length()) {*  *int size = readNextBatch(primeNumbers, inout);*  *for (int k = 0; k < size && primeNumbers[k] <= squareRoot; k++) {*  *if (number % primeNumbers[k] == 0) { // this number is not prime*  *continue newNumber;*  *} // end if*  *} // end for*  *}// end while*    *// Append a new prime number to the end of the file*  *inout.seek(inout.length());*  *inout.writeLong(number);*    *} // end newNumber while loop*  *} // end of main*    *public static int readNextBatch(long[] primeNumbers, RandomAccessFile inout) {*  *int size = 0;*  *try {*  *while(inout.getFilePointer() < inout.length() && size < ARRAY\_SIZE) {*  *primeNumbers[size++] = inout.readLong();*  *}*  *}*  *catch (Exception ex) {*  *ex.printStackTrace();*  *}*  *return size;*  *} // end method*    *}* |
|  |