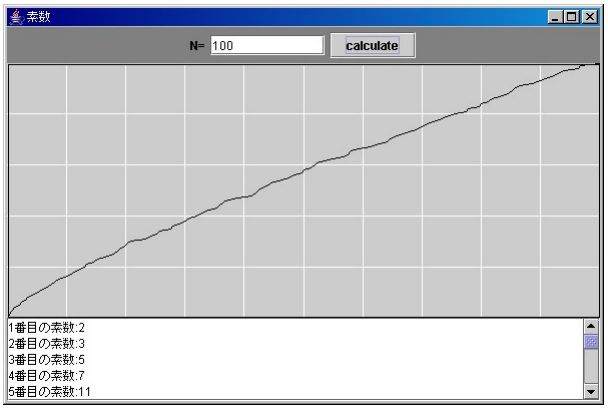
2005-7-2




```
package jp.ac.kindai.prime;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Prime extends JFrame implements
    ActionListener,Runnable{
        Thread th = null;
        public static int N;  // 求める素数の個数
        public static int[] primeN;
            // primeNum[i]:i 番目の素数（クラス配列）
        public static JFrame windowFrame;
        // ウインドウ（クラス変数）
        public static JPanel graphPanel;              // グラフ表示用パネル（クラス変数）
        public static JTextArea txtAdispPrimes;       // 素数を表示するテキストエリア（クラス変数）
        public static JTextField txtF_N;              // 求める素数の個数を入力するテキストフィールド

        public static void main( String[] args ) { Prime prime = new Prime(); }

        /** 素数を N 個求める */
        static void getPrimes(int n){
                                //--- n 番目の素数の次に大きい素数:n+1 番目の素数を求める
                for( int i=1; i < n ; i++) { primeN[i+1] = nextPrime( primeN[i] ); }
        }

        /**  テキストエリア：txtAdispPrimes 素数を表示 */
        static void showPrimes(int n){
                for( int i=1; i <= n ; i++) { txtAdispPrimes.append( i + "番目の素数:" + primeN[i] +"¥n" ); }
        }

        /** パネル：graphPanel に素数分布グラフを表示 */
        static void showGraph(int n) {
                Graphics g = graphPanel.getGraphics();
                int H = graphPanel.getHeight(); int W = graphPanel.getWidth();
                        g.clearRect(0,0,W,H);
                double dx = (1.0*W)/(1.0*primeN[n]); double dy = (1.0*H)/(1.0*n);

                // 目盛
                g.setColor(Color.white);
                double x = 0;      double y = H;      double x_delta = dx*primeN[n]/10.0;
                // 縦線
                for(int i=1; i < 10 ; i++){
                        x = x + x_delta;
                        g.drawLine((int)x, (int)y, (int)x, (int)(y-H+2));
                }
                // 横線
                x=0 ;     y = H;    double y_delta = 1.0*H/5;
                for(int i=1; i < 5 ; i++){
                        y = y - y_delta;   g.drawLine((int)x, (int)y, (int)(x+W), (int)y);
                }
                // 外枠
                g.setColor(Color.black);     g.drawRect(1,1,W-2,H-2);
                // グラフ
                double x0=0;       double y0=H;       double x1=0;       double y1=H;
                for(int i=1; i <= n ; i++){
                        x1=1.0*dx*primeN[i]; y1=y0-dy;
                        g.drawLine((int)x0, (int)y0, (int)x1, (int)y1);         x0=x1; y0=y1;
                }
        }

        //* nextPrime(int p)：素数 p の次に大きい素数を求め返す */
        static int nextPrime( int p ){
                int m = p+1; while( !isPrime(m) ){ m++; }
```

```
            return m;
        }

        //* isPrime(int n) : n が素数ならば true, そうでなければ false を戻す */
        static boolean isPrime( int n ){
            // n=p*q (p <= q) とすると, q=n/p >= p だから
            // p を素数として小さい数から代入し, n/p >= p が成り立つ範囲で割り切れるか調べれば十分
            for(int j = 1; n/primeN[j] >= primeN[j] ; j++){
                int p = primeN[j];
                // n が素数 p で割り切れたら n は素数ではないので false を戻す
                if( Math.IEEEremainder(n,p) == 0 ){ return false; }
            }
            return true;
        }

        //constractor of Prime
        public Prime(){
            JPanel control_panel = new JPanel(); //パネル：ラベル, ボタン, テキストフィールド
                control_panel.setBackground(Color.gray);
                JLabel labelN = new JLabel("N=");
                txtF_N = new JTextField("100",10);
                JButton btnDo = new JButton("calculate");
                btnDo.addActionListener(this);
              control_panel.add(labelN);  control_panel.add(txtF_N);  control_panel.add(btnDo);

            graphPanel = new JPanel();  //パネル：グラフ描画エリア
                graphPanel.setBackground(Color.lightGray);

            txtAdispPrimes = new JTextArea(5,50);          //テキストエリア
            JScrollPane scroll = new JScrollPane(txtAdispPrimes);
            scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);

        //画面デザイン：北（ラベル, テキストフィールド, ボタン), 中央（グラフ), 南（テキストエリア）
            windowFrame = new JFrame("素数");

                Container container = windowFrame.getContentPane();
                    container.setLayout(new BorderLayout());
                        container.add(control_panel,BorderLayout.NORTH);
                        container.add(graphPanel,BorderLayout.CENTER);
                        container.add(scroll,BorderLayout.SOUTH);
                    windowFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                windowFrame.setBounds(200,100,600,400);        // ウインドウの表示位置とサイズ
                    windowFrame.setVisible(true);
        }

        //  java.awt.event.ActionListener
        public void actionPerformed(ActionEvent arg0) {
            N = Integer.parseInt(txtF_N.getText());  // N を取得し, 小さい方から N 個素数を求め,
            if( th == null){ th = new Thread(this);   th.start(); }
        }

        public void run(){
            primeN = new int[N+1];      //素数を格納する配列
            primeN[1] = 2;              //1 番目の素数は 2
            getPrimes(N);
            showPrimes(N);     // １）テキストエリアに表示,
            showGraph(N);      // ２）素数の分布のグラフを表示
            th = null;
        }

}
```