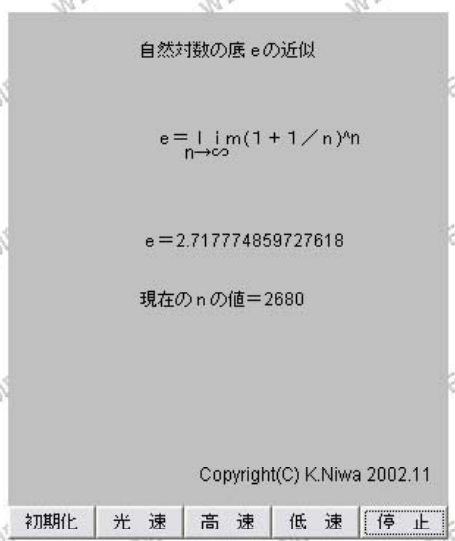
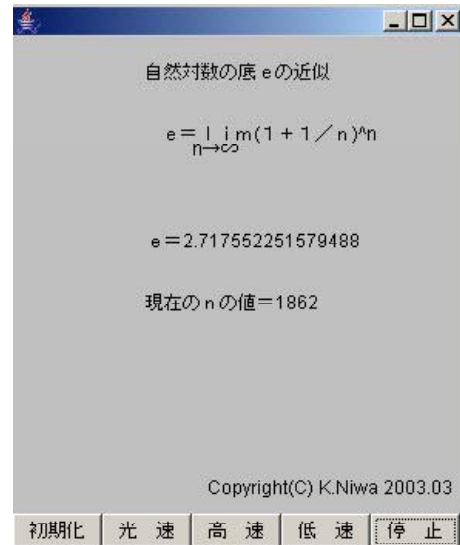


【自然対数の底 e】



【J a v a アプレット】



【J a v a アプリケーション】

1. はじめに

対数は、17世紀にネイピアやビュルギといった数学者たちが生み出した関数である。

円周率 π と自然対数の底 e とは密接な関係があり、どちらも無理数で超越数（整数係数の代数方程式の解にならない実数）である。

1737年、オイラーは、 e が無理数であることを示した。

1873年、フランスの数学者エルミートは、 e が超越数であることを証明した。

1882年、ドイツ人数学者リンデマンが、エルミートの方針に従って、 π が超越数であることを示し、二千年来の超難問である「円積問題」を解いた。

エルミートは、 π が超越数であることの証明は e が超越数であることの証明に比べて非常に困難であると考えていたのである。

そのとき、エルミートは60歳、リンデマンは30歳であった。

また、美しい数学の公式の中でも、特に美しいものとして「人類の至宝（しほう）」とまで言われている次の式をオイラーは見つけている。

$$e^{i\pi} + 1 = 0$$

e 、 i 、 π 、 1 、 0 といった数学の重要な数が全て登場し、しかもコンパクトにまとめられているが、この式の意味は未だによく分かっていないのである。

【円積問題について】

作図問題のうち最も古く最もよく知られているのは、定木（目盛りのない、直線を引くためのもの）とコンパスによる平面図形の作図で、そのうちのギリシアの三大作図不可能問題の一つに「円積問題」がある。

- (1) 与えられた円と等しい面積をもつ正方形を作ること（円積問題）
- (2) 与えられた立方体の体積の2倍に等しい体積をもつ立方体を作ること（立方体倍積問題または Delos の問題）
- (3) 与えられた角を3等分すること（角の3等分問題）

シミュレーションソフト「自然対数の底 e 」を動かして、自然対数 e の近似値が求まる様子を観察してみてください。


```

myBtn[1].addActionListener(new ActionListener() { //光速ボタンの定義
    public void actionPerformed(ActionEvent e) {
        flag=1;
        Speed=20; //速度
        repaint();
    }
});

myBtn[2].addActionListener(new ActionListener() { //高速ボタンの定義
    public void actionPerformed(ActionEvent e) {
        flag=2;
        Speed=100; //速度
        repaint();
    }
});

myBtn[3].addActionListener(new ActionListener() { //低速ボタンの定義
    public void actionPerformed(ActionEvent e) {
        flag=3;
        Speed=200; //速度
        repaint();
    }
});

myBtn[4].addActionListener(new ActionListener() { //停止ボタンの定義
    public void actionPerformed(ActionEvent e) {
        flag=4;
        repaint();
    }
});

} //public void init()

/***** public void start()メソッド *****/
//
// アプレットの実行を開始するとき呼ばれる
//
public void start() {
    if (myTh==null) {
        myTh=new Thread(this); //スレッドの実体化
        myTh.start(); //スレッドを開始する
    }
}

/***** public void run()メソッド *****/
//
// スレッドの実行中に呼ばれる
//
public void run() {
    while (true) {
        try {
            myTh.sleep(Speed);
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2 || flag==3) {
            repaint();
        }
    }
}

```

```

/***** public void paint(Graphics g)メソッド *****/
// アプレットの開始時に呼ばれる
// repaint()メソッドで呼ばれる
// update()メソッドで呼ばれる
//
public void paint(Graphics g) {
//初期状態または初期化ボタンを押したときのイベント処理
if(flag==0) {
n=1; // e = (1 + 1/n)^n の n の初期化
tte=0; // e の近似値の初期化
s=1;
k=1;
g.drawString("自然対数の底 e の近似",90,30);
//g.drawString(" (オイラー数) ",115,50);
g.drawString(" e = l i m ( 1 + 1 / n ) ^ n ",100,90);
g.drawString(" n → ∞ ",120,100);
g.drawString(" e = ",90,160);
g.drawString("現在の n の値=",90,200); //現在の n の値を表示
g.drawString("Copyright (C) K.Niwa 2002.11",130,320); //作者表示
} //if(flag==0)

//光速または高速または低速ボタンを押したときのイベント処理
else if (flag==1 || flag==2 || flag==3) {
if (flag==1) {
d=1; //光速
}
else if (flag==2) {
d=1; //高速
}
else if (flag==3) {
d=1; //低速
}

if (n<999999999) {
n=n+d;
}
else {
flag=4; //停止させる
}

s=1;
for (count=1;count<=n;count++) {
k=(double) 1+(double) 1/(double) n;
s=s*k;
}

tte=s;

g.drawString("自然対数の底 e の近似",90,30);
//g.drawString(" (オイラー数) ",115,50);
g.drawString(" e = l i m ( 1 + 1 / n ) ^ n ",100,90);
g.drawString(" n → ∞ ",120,100);
g.drawString(" e = "+(double) tte,90,160);
g.drawString("現在の n の値="+n,90,200); //現在の n の値を表示
g.drawString("Copyright (C) K.Niwa 2002.11",130,320); //作者表示

} //else if (flag==1 || flag==2 || flag==3)

//停止ボタンを押したときのイベント処理
if(flag==4) {
g.drawString("自然対数の底 e の近似",90,30);
//g.drawString(" (オイラー数) ",115,50);
g.drawString(" e = l i m ( 1 + 1 / n ) ^ n ",100,90);
g.drawString(" n → ∞ ",120,100);
g.drawString(" e = "+(double) tte,90,160);
}
}

```

```

        g.drawString("現在の n の値="+n,90,200); //現在の n の値を表示
        g.drawString("Copyright (C) K.Niwa 2002.11",130,320); //作者表示
    } //if(flag==4)
    } //public void paint(Graphics g)
} //public class Ste extends Applet implements Runnable

```

(2) HTML リスト

```

<HTML>
  <HEAD>
  < !-----
                                「自然対数の底 e の近似」
                                Copyright (C) K.Niwa 2002.11.03
----->
  </HEAD>
  <BODY>
    <CENTER>
      <B> 「自然対数の底 e の近似」 </B>
      <BR><BR>
      <APPLET CODE="Ste.class" WIDTH="300" HEIGHT="360"></APPLET>
    </CENTER>
  </BODY>
</HTML>

```

3. J a v a アプリケーション・プログラムリスト

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                                                               //
//              「 F 自然対数の底 e 」                                                       //
//            Copyright (C) K.Niwa 2003.03.22                                             //
//            ( J a v a アプリケーション )                                                 //
//                                                                                               //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//ライブラリーのクラスの読み込み
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;

/***** public class FSte extends Frame implements Runnable *****/
public class FSte extends Frame implements Runnable {
    Thread myTh;                                //スレッドをたえるようにする
    Button[] myBtn;                              //スレッド型で宣言する
    Panel myPanel;                              //ボタン型配列で宣言する
    //パネル型で宣言する
    int flag=0;
    int Speed=100;                               //速度
    int ct=0;                                    //実験回数
    int count;                                  //ループカウンタ
    int d;                                       //速度の刻み幅に使用
    long n;                                     // e = (1 + 1/n)^n
    double tte;                                 // e の近似値
    double s;                                  // e = (1 + 1/n)^n を求めるのに使用
    double k;                                  // e = (1 + 1/n)^n を求めるのに使用
}

```

```

/***** フレームとイベントの定義 *****/
// ブラウザがアプレットを読み込んだときに呼ばれる
// イベントの監視を行い、ボタンが選択されたとき等にも呼ばれる
// 背景色の設定やボタンの定義や設置等に用いられる
//
public FSte() {
    setSize(300,360);
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });

    setBackground(Color.lightGray);
    myTh=null; //スレッドの初期化
    if (myTh==null) {
        myTh=new Thread(this); //スレッドの実体化
        myTh.start(); //スレッドを開始する
    }
    myBtn=new Button[5]; //ボタンの実体化
    myBtn[0]=new Button("初期化");
    myBtn[1]=new Button("光 速");
    myBtn[2]=new Button("高 速");
    myBtn[3]=new Button("低 速");
    myBtn[4]=new Button("停 止");
    myPanel=new Panel(); //パネルの実体化
    myPanel.setLayout(new GridLayout(1,5));
    for (count=0;count<=4;count++) {
        myPanel.add(myBtn[count]); //パネルにボタンを貼り付ける
    }
    setLayout(new BorderLayout()); //全体をボーダーレイアウトにする
    add("South",myPanel); //パネルを南に貼り付ける

    myBtn[0].addActionListener(new ActionListener() { //初期化ボタンの定義
        public void actionPerformed(ActionEvent e) {
            flag=0;
            repaint();
        }
    });

    myBtn[1].addActionListener(new ActionListener() { //光速ボタンの定義
        public void actionPerformed(ActionEvent e) {
            flag=1;
            Speed=20; //速度
            repaint();
        }
    });

    myBtn[2].addActionListener(new ActionListener() { //高速ボタンの定義
        public void actionPerformed(ActionEvent e) {
            flag=2;
            Speed=100; //速度
            repaint();
        }
    });

    myBtn[3].addActionListener(new ActionListener() { //低速ボタンの定義
        public void actionPerformed(ActionEvent e) {
            flag=3;
            Speed=200; //速度
            repaint();
        }
    });
}

```

```

        myBtn[4].addActionListener(new ActionListener() { //停止ボタンの定義
            public void actionPerformed(ActionEvent e) {
                flag=4;
                repaint();
            }
        });
    } //public FSte()

    /******* public void run()メソッド ***** */
    // スレッドの実行中に呼ばれる
    //
    public void run() {
        while (true) {
            try {
                myTh.sleep(Speed);
            }
            catch (InterruptedException e) {}
            if (flag==1 || flag==2 || flag==3) {
                repaint();
            }
        }
    }

    /******* public void paint(Graphics g)メソッド ***** */
    // アプレットの開始時に呼ばれる
    // repaint()メソッドで呼ばれる
    // update()メソッドで呼ばれる
    //
    public void paint(Graphics g) {
        //初期状態または初期化ボタンを押したときのイベント処理
        if(flag==0) {

            n=1; //  $e = (1 + 1/n)^n$  の n の初期化
            tte=0; // e の近似値の初期化
            s=1;
            k=1;

            g.drawString("自然対数の底 e の近似",90,30+20);
            //g.drawString(" (オイラー数) ",115,50);
            g.drawString("e = l i m ( 1 + 1 / n ) ^ n",100,90);
            g.drawString("n → ∞",120,100);
            g.drawString("e =",90,160);
            g.drawString("現在の n の値=",90,200); //現在の n の値を表示
            g.drawString("Copyright (C) K.Niwa 2003.03",130,320); //作者表示

        } //if(flag==0)

        //高速ボタンまたは低速ボタンを押したときのイベント処理
        else if (flag==1 || flag==2 || flag==3) {
            if (flag==1) {
                d=1; //光速
            }
            else if (flag==2) {
                d=1; //高速
            }
            else if (flag==3) {
                d=1; //低速
            }

            if (n<999999999) {
                n=n+d;
            }
        }
    }

```

```

else {
    flag=4;          //停止させる
}

s=1;
for (count=1;count<=n;count++) {
    k=(double) 1+(double) 1/(double) n;
    s=s*k;
}

tte=s;

g.drawString("自然対数の底 e の近似",90,30+20);
//g.drawString(" (オイラー数) ",115,50);
g.drawString(" e = l i m ( 1 + 1 / n ) ^ n ",100,90);
g.drawString(" n → ∞ ",120,100);
g.drawString(" e = "+(double) tte,90,160);
g.drawString("現在の n の値="+n,90,200); //現在の n の値を表示
g.drawString("Copyright (C) K.Niwa 2003.03",130,320); //作者表示

} //else if (flag==1 || flag==2 || flag==3)

//停止ボタンを押したときのイベント処理
if(flag==4) {
    g.drawString("自然対数の底 e の近似",90,30+20);
    //g.drawString(" (オイラー数) ",115,50);
    g.drawString(" e = l i m ( 1 + 1 / n ) ^ n ",100,90);
    g.drawString(" n → ∞ ",120,100);
    g.drawString(" e = "+(double) tte,90,160);
    g.drawString("現在の n の値="+n,90,200); //現在の n の値を表示
    g.drawString("Copyright (C) K.Niwa 2003.03",130,320); //作者表示

} //if(flag==4)

} //public void paint(Graphics g)

/***** public static void main(String[] args) *****/
public static void main(String[] args) {
    Frame w=new FSte();
    w.show();
} //public static void main(String[] args)

} //public class Ste extends Applet implements Runnable

```