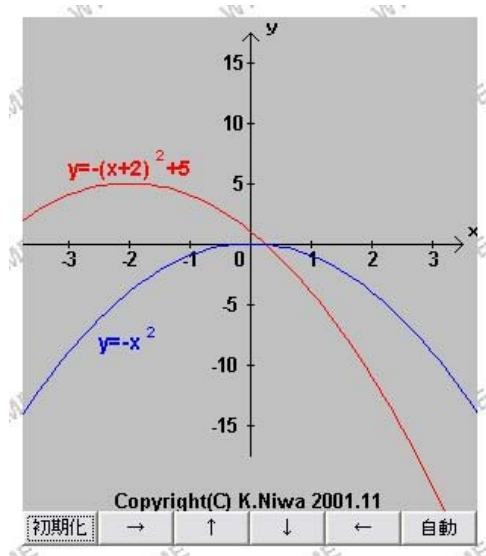
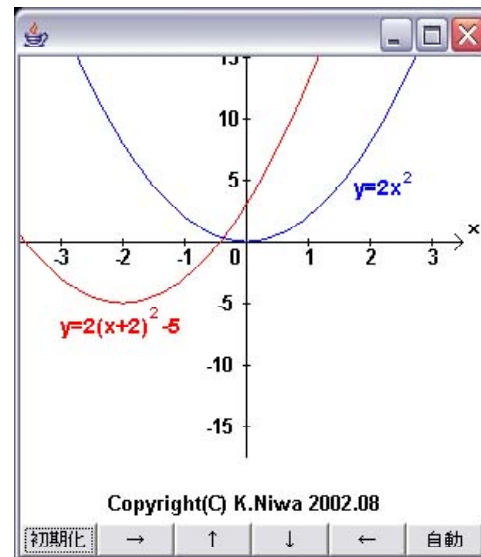


【2次関数のグラフの平行移動】



【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

$y = a x^2$ のグラフと $y = a (x - b)^2 + c$ のグラフは、位置は違うけれど、形も広がりも全く同じです。

$y = a (x - b)^2 + c$ のグラフは、 $y = a x^2$ のグラフを x 軸方向に(右方向に) $+ b$ 、y 軸方向に(上方向に) $+ c$ だけ平行移動したものです。

シミュレーションソフト「2次関数のグラフの平行移動」を使って、 $y = a (x - b)^2 + c$ のグラフと、 $y = a x^2$ のグラフの位置関係を観察してみてください。

2. J a v a アプレット

(1) J a v a プログラムリスト

```

////////////////////////////////////
//
//                               「2次関数の平行移動2」
//                               y=- (x+2)^2+5
//                               Copyright (C) K.Niwa 2001.11.17
//                               (J a v a アプレット)
//
////////////////////////////////////

```

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;

```

```

public class Kansuu2 extends Applet{
    Button[] myBtn; //ボタン型として宣言する
    Panel myPanel; //パネル型として宣言する
    int flag=0; //変数宣言と初期化
    double x,y; //グラフの座標
    int px,py,oldpx,oldpy; //グラフの座標
    int N; //ゴマの個数
    int count,i; //ループカウンタ
}

```

```
int fg;
double a=0,b=0;
int t;
```

```
//グラフ描写に使用したフラグ
//グラフの頂点の座標
//時間かせぎ
```

```
/**public void init()   init メソッド (void は値を返さないメソッドの意)*****
public void init() {
    setBackground(Color.lightGray);           //背景色をグレーにする
    myBtn=new Button[6];                       //ボタンの実体化
    myBtn[0]=new Button("初期化");
    myBtn[1]=new Button("→");
    myBtn[2]=new Button("↑");
    myBtn[3]=new Button("↓");
    myBtn[4]=new Button("←");
    myBtn[5]=new Button("自動");

    myPanel=new Panel();                       //パネルの実体化
    myPanel.setLayout(new GridLayout(1,6));   //パネルをグリッドレイアウトにする
    for (count=0;count<=5;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

//初期化ボタンを押したときのイベント処理
    myBtn[0].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            a=0;
            b=0;
            repaint();
        }
    });

//→ボタンを押したときのイベント処理
    myBtn[1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            a=a+0.1;
            repaint();
        }
    });

//↑ボタンを押したときのイベント処理
    myBtn[2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            b=b+0.5;
            repaint();
        }
    });

//↓ボタンを押したときのイベント処理
    myBtn[3].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            b=b-0.5;
            repaint();
        }
    });
};
```

```

//←ボタンを押したときのイベント処理
myBtn[4].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=0;
        a=a-0.1;
        repaint();
    }
});

//自動ボタンを押したときのイベント処理
myBtn[5].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flag=1;
        a=0;
        b=0;
        repaint();
    }
});

} //public void init()

****public void paint(Graphics g)****paint メソッド *****
public void paint(Graphics g) //paint メソッド {

//手動のとき
if (flag==0) {
    g.clearRect(0,0,300,350); //全体のクリア
    Font f0=new Font((g.getFont()).getName(),Font.BOLD,13);
    g.setFont(f0);

//座標軸
g.drawLine(150,10,150,290); //y 軸
g.drawLine(150,10,150-5,10+5);
g.drawLine(150,10,150+5,10+5);
g.drawString(" y ",155+3,10);
for (i=0;i<=120;i=i+40) {
    g.drawLine(150-2,150-i,150+2,150-i); //y 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine(150-2,150+i,150+2,150+i); //y 軸メモリ
}
g.drawString(" 5",150-15-3+3,110+5); //y 軸メモリ値
g.drawString("10",150-15-3,70+5);
g.drawString("15",150-15-3,30+5);
g.drawString(" -5",150-20-6+3,190+5);
g.drawString("-10",150-20-6,230+5);
g.drawString("-15",150-20-6,270+5); //y 軸メモリ値
g.drawLine(0,150,290,150); //x 軸
g.drawLine(290,150,290-5,150-5);
g.drawLine(290,150,290-5,150+5);
g.drawString(" x ",290,145);
for (i=0;i<=120;i=i+40) {
    g.drawLine(150+i,150-2,150+i,150+2); //x 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine(150-i,150-2,150-i,150+2); //x 軸メモリ
}
g.drawString("1",190-2,165); //x 軸メモリ値
g.drawString("2",230-2,165);
g.drawString("3",270-2,165);
g.drawString("0",150-10,165);
g.drawString("-1",110-5,165);
g.drawString("-2",70-5,165);
g.drawString("-3",30-5,165); //x 軸メモリ値

```

```

//グラフ描写
//y=-(x+2)^2+5
g.setColor(Color.red); //赤色にする

Font f1=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f1);
g.drawString("y=-(x+2)",30,105);

Font f2=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f2);
g.drawString("2",88,95);

Font f3=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f3);
g.drawString("+5",95,105);

for (x=-4;x<=4.1;x=x+0.1) {
    y=-1*(x+2)*(x+2)+5;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else{
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.blue); //青色にする

Font f4=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f4);
g.drawString("y=-x",50,220);

Font f5=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f5);
g.drawString("2",82,212);

for (x=-4;x<=4.1;x=x+0.1) { //
    y=-1*(x-a)*(x-a)+b;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else{
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.black); //黒色にする

Font f6=new Font((g.getFont()).getName(),Font.BOLD,13);
g.setFont(f6);
g.drawString("Copyright (C) K.Niwa 2001.11",60,325); //作者名
} //if (flag==0)

```

```

//自動のとき
else if (flag==1) {
// x 軸方向の移動
for (a=0;a>=-2;a=a-0.01) {
g.clearRect(0,0,300,350);

//座標軸
g.drawLine(150,10,150,290); // y 軸
g.drawLine(150,10,150-5,10+5);
g.drawLine(150,10,150+5,10+5);
g.drawString(" y ",155+3,10);
for (i=0;i<=120;i=i+40) {
g.drawLine(150-2,150-i,150+2,150-i); // y 軸メモリ
}
for (i=0;i<=120;i=i+40) {
g.drawLine(150-2,150+i,150+2,150+i); // y 軸メモリ
}
g.drawString(" 5",150-15-3+3,110+5); // y 軸メモリ値
g.drawString("10",150-15-3,70+5);
g.drawString("15",150-15-3,30+5);
g.drawString(" -5",150-20-6+3,190+5);
g.drawString("-10",150-20-6,230+5);
g.drawString("-15",150-20-6,270+5); // y 軸メモリ値

g.drawLine(0,150,290,150); // x 軸
g.drawLine(290,150,290-5,150-5);
g.drawLine(290,150,290-5,150+5);
g.drawString(" x ",290,145);
for (i=0;i<=120;i=i+40) {
g.drawLine(150+i,150-2,150+i,150+2); // x 軸メモリ
}
for (i=0;i<=120;i=i+40) {
g.drawLine(150-i,150-2,150-i,150+2); // x 軸メモリ
}
g.drawString("1",190-2,165); // x 軸メモリ値
g.drawString("2",230-2,165);
g.drawString("3",270-2,165);
g.drawString("0",150-10,165);
g.drawString("-1",110-5,165);
g.drawString("-2",70-5,165);
g.drawString("-3",30-5,165); // x 軸メモリ値

//グラフ描写
//y=-(x+2)^2+5
g.setColor(Color.red); //赤にする

Font f7=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f7);
g.drawString("y=-(x+2)",30,105);

Font f8=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f8);
g.drawString("2",90-2,95);

Font f9=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f9);
g.drawString("+5",100-5,105);

```

```

for (x=-4;x<=4.1;x=x+0.1) {
    y=-1*(x+2)*(x+2)+5;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else
    {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

//y=-x^2
g.setColor(Color.blue);

Font f10=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f10);
g.drawString("y=-x",50,220);

Font f11=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f11);
g.drawString("2",82,212);

for (x=-4;x<=4.1;x=x+0.1) {
    y=-1*(x-a)*(x-a)+b;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.black);
Font f12=new Font((g.getFont()).getName(),Font.BOLD,13);
g.setFont(f12);
g.drawString("Copyright(C) K.Niwa 2001.11",60,325); //作者名

for (t=1;t<100000;t++) {} //時間かせぎ

} //for (a=-2;a<=0;a=a+0.01)

// y 軸方向の移動
for (b=0;b<=5;b=b+0.02) {
    g.clearRect(0,0,300,350);

    //座標軸
    g.drawLine(150,10,150,290); // y 軸
    g.drawLine(150,10,150-5,10+5);
    g.drawLine(150,10,150+5,10+5); //
    g.drawString(" y ",155+3,10);
    for (i=0;i<=120;i=i+40) {
        g.drawLine(150-2,150-i,150+2,150-i); // y 軸メモリ
    }
}

```

```

for (i=0;i<=120;i=i+40) {
    g.drawLine (150-2,150+i,150+2,150+i); // y 軸メモリ
}
g.drawString (" 5",150-15-3+3,110+5); // y 軸メモリ値
g.drawString ("10",150-15-3,70+5);
g.drawString ("15",150-15-3,30+5);
g.drawString (" -5",150-20-6+3,190+5);
g.drawString ("-10",150-20-6,230+5);
g.drawString ("-15",150-20-6,270+5); // y 軸メモリ値

g.drawLine (0,150,290,150);
g.drawLine (290,150,290-5,150-5);
g.drawLine (290,150,290-5,150+5); // x 軸
g.drawString (" x ",290,145);
for (i=0;i<=120;i=i+40) {
    g.drawLine (150+i,150-2,150+i,150+2); // x 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine (150-i,150-2,150-i,150+2); // x 軸メモリ
}
g.drawString ("1",190-2,165); // x 軸メモリ値
g.drawString ("2",230-2,165);
g.drawString ("3",270-2,165);
g.drawString ("0",150-10,165);
g.drawString ("-1",110-5,165);
g.drawString ("-2",70-5,165);
g.drawString ("-3",30-5,165); // x 軸メモリ値

```

//グラフ描写

```

//y=-(x+2)^2+5
g.setColor (Color.red); //赤色にする

Font f13=new Font ((g.getFont()).getName(),Font.BOLD,14);
g.setFont (f13);
g.drawString ("y=-(x+2)",30,105);

Font f14=new Font ((g.getFont()).getName(),Font.BOLD,10);
g.setFont (f14);
g.drawString ("2",90-2,95);

Font f15=new Font ((g.getFont()).getName(),Font.BOLD,14);
g.setFont (f15);
g.drawString ("+5",100-5,105);

for (x=-4;x<=4.1;x=x+0.1) {
    y=-1*(x+2)*(x+2)+5;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine (px,py,px,py);
    }
    else {
        g.drawLine (oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

```

```

//y=-x^2
g.setColor(Color.blue);

Font f16=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f16);
g.drawString("y=-x",50,220);

Font f17=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f17);
g.drawString("2",82,212);

for (x=-4;x<=4.1;x=x+0.1) {
    y=-1*(x-a)*(x-a)+b;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.black);
Font f18=new Font((g.getFont()).getName(),Font.BOLD,13);
g.setFont(f18);
g.drawString("Copyright (C) K.Niwa 2001.11",60,325); //作者名

for (t=1;t<1000000;t++) {} //時間かせぎ

} //for (b=-5;b<=0;b=b+0.02)
} //else if (flag==1)
} //public void paint(Graphics g) //paint メソッド
} //public class Kansuu extends Applet

```

(2) HTML リスト

```

<HTML>
  <HEAD>
  < !-----
                                2 次関数の平行移動 2
                                y=- (x-2)^2+5
                                Copyright (C) K.Niwa 2001.11.17
  ----->
  </HEAD>
  <BODY>
    <CENTER>
      <B> 2 次関数の平行移動 2 </B>
      <BR><BR>
      <APPLET CODE="Kansuu2.class" WIDTH="300" HEIGHT="350"></APPLET>
      <BR><BR>
      <B>※ 赤いグラフは、青いグラフを x 軸方向に - 2、y 軸方向に + 5、平
                                行移動したものである。</B>
      <BR>
      <B>※ この 2 つの放物線は、位置は違うけれども形も広がりも全く同じで
                                あることを確認してください... </B>
    </CENTER>
  </BODY>
</HTML>

```



```

    });
//→ボタンを押したときのイベント処理
    myBtn[1].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            a=a+0.1;
            repaint();
        }
    });
//↑ボタンを押したときのイベント処理
    myBtn[2].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            b=b+0.5;
            repaint();
        }
    });
//↓ボタンを押したときのイベント処理
    myBtn[3].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            b=b-0.5;
            repaint();
        }
    });
//←ボタンを押したときのイベント処理
    myBtn[4].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=0;
            a=a-0.1;
            repaint();
        }
    });
//自動ボタンを押したときのイベント処理
    myBtn[5].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            flag=1;
            a=0;
            b=0;
            repaint();
        }
    });
} //public FKansuu1 ()

/**** public void paint(Graphics g) メソッド ****
public void paint(Graphics g) //paint メソッド {
//手動のとき
    if (flag==0) {
        g.clearRect(0,0,300,360); //全体のクリア
        Font f0=new Font((g.getFont()).getName(),Font.BOLD,13);
        g.setFont(f0);

        //座標軸
        g.drawLine(150,10,150,290); //y 軸
        g.drawLine(150,10,150-5,10+5);
        g.drawLine(150,10,150+5,10+5);
        g.drawString("y",155+3,10);

```

```

for (i=0;i<=120;i=i+40) {
    g.drawLine (150-2,150-i,150+2,150-i); // y 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine (150-2,150+i,150+2,150+i); // y 軸メモリ
}
g.drawString (" 5",150-15-3+3,110+5); // y 軸メモリ値
g.drawString ("10",150-15-3,70+5);
g.drawString ("15",150-15-3,30+5);
g.drawString (" -5",150-20-6+3,190+5);
g.drawString ("-10",150-20-6,230+5);
g.drawString ("-15",150-20-6,270+5); // y 軸メモリ値

g.drawLine (0,150,290,150); // x 軸
g.drawLine (290,150,290-5,150-5);
g.drawLine (290,150,290-5,150+5);
g.drawString (" x ",290,145);
for (i=0;i<=120;i=i+40) {
    g.drawLine (150+i,150-2,150+i,150+2); // x 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine (150-i,150-2,150-i,150+2); // x 軸メモリ
}
g.drawString ("1",190-2,165); // x 軸メモリ値
g.drawString ("2",230-2,165);
g.drawString ("3",270-2,165);
g.drawString ("0",150-10,165);
g.drawString ("-1",110-5,165);
g.drawString ("-2",70-5,165);
g.drawString ("-3",30-5,165); // x 軸メモリ値

```

//グラフ描写

```

//y=2(x+2)^2-5
g.setColor (Color.red); //グラフの色を赤にする

Font f1=new Font ((g.getFont ()).getName (),Font.BOLD,14);
g.setFont (f1);
g.drawString ("y=2(x+2)",30,210);

Font f2=new Font ((g.getFont ()).getName (),Font.BOLD,10);
g.setFont (f2);
g.drawString ("2",90-2,200);

Font f3=new Font ((g.getFont ()).getName (),Font.BOLD,14);
g.setFont (f3);
g.drawString ("-5",100-5,210);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x+2)*(x+2)-5;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine (px,py,px,py);
    }
    else {
        g.drawLine (oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

```

```

//y=2x^2
g.setColor(Color.blue); //グラフの色を青にする

Font f4=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f4);
g.drawString("y=2x",220,120);

Font f5=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f5);
g.drawString("2",250+2,110+2);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x-a)*(x-a)+b;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.black);

Font f6=new Font((g.getFont()).getName(),Font.BOLD,13);
g.setFont(f6);
g.drawString("Copyright (C) K.Niwa 2002.08",60,325); //作者名

} //if (flag==0)

//自動のとき
else if (flag==1) {
// x 軸方向の移動
for (a=0;a>=-2;a=a-0.01) {
    g.clearRect(0,0,300,300);

//座標軸
g.drawLine(150,10,150,290); // y 軸
g.drawLine(150,10,150-5,10+5);
g.drawLine(150,10,150+5,10+5);
g.drawString(" y ",155+3,10);
for (i=0;i<=120;i=i+40) {
    g.drawLine(150-2,150-i,150+2,150-i); // y 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine(150-2,150+i,150+2,150+i); // y 軸メモリ
}
g.drawString(" 5",150-15-3+3,110+5); // y 軸メモリ値
g.drawString(" 10",150-15-3,70+5);
g.drawString(" 15",150-15-3,30+5);
g.drawString(" -5",150-20-6+3,190+5);
g.drawString(" -10",150-20-6,230+5);
g.drawString(" -15",150-20-6,270+5); // y 軸メモリ値

g.drawLine(0,150,290,150); // x 軸
g.drawLine(290,150,290-5,150-5);
g.drawLine(290,150,290-5,150+5);
g.drawString(" x ",290,145);
}
}

```

```

for (i=0;i<=120;i=i+40) {
    g.drawLine (150+i,150-2,150+i,150+2); // x 軸メモリ
}
for (i=0;i<=120;i=i+40) {
    g.drawLine (150-i,150-2,150-i,150+2); // x 軸メモリ
}
g.drawString ("1",190-2,165); // x 軸メモリ値
g.drawString ("2",230-2,165);
g.drawString ("3",270-2,165);
g.drawString ("0",150-10,165);
g.drawString ("-1",110-5,165);
g.drawString ("-2",70-5,165);
g.drawString ("-3",30-5,165); // x 軸メモリ値

//グラフ描写
//y=2(x+2)^2-5
g.setColor (Color.red); //グラフを赤色にする

Font f7=new Font ((g.getFont()).getName(),Font.BOLD,14);
g.setFont (f7);
g.drawString ("y=2(x+2)",30,210);

Font f8=new Font ((g.getFont()).getName(),Font.BOLD,10);
g.setFont (f8);
g.drawString ("2",90-2,200);

Font f9=new Font ((g.getFont()).getName(),Font.BOLD,14);
g.setFont (f9);
g.drawString ("-5",100-5,210);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x+2)*(x+2)-5;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine (px,py,px,py);
    }
    else {
        g.drawLine (oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

//y=2x^2
g.setColor (Color.blue); //グラフを青色にする

Font f10=new Font ((g.getFont()).getName(),Font.BOLD,14);
g.setFont (f10);
g.drawString ("y=2x",220,120);

Font f11=new Font ((g.getFont()).getName(),Font.BOLD,10);
g.setFont (f11);
g.drawString ("2",250+2,110+2);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x-a)*(x-a)+b;
    px=(int) (150+40*x);
    py=(int) (150-8*y);
    if (fg==0) {
        g.drawLine (px,py,px,py);
    }
}

```

```

        else {
            g.drawLine(oldpx,oldpy,px,py);
        }
        oldpx=px;oldpy=py;
        fg++;
    }
    fg=0;

    g.setColor(Color.black);
    Font f12=new Font((g.getFont()).getName(),Font.BOLD,13);
    g.setFont(f12);
    g.drawString("Copyright(C) K.Niwa 2002.08",60,325); //作者名

    for (t=1;t<1000000;t++) {} //時間かせぎ

} //for (a=0;a>=-2;a=a-0.01)

// y 軸方向の移動
for (b=0;b>=-5;b=b-0.02) {
    g.clearRect(0,0,300,300);

    //座標軸
    g.drawLine(150,10,150,290); // y 軸
    g.drawLine(150,10,150-5,10+5);
    g.drawLine(150,10,150+5,10+5);
    g.drawString(" y ",155+3,10);
    for (i=0;i<=120;i=i+40) {
        g.drawLine(150-2,150-i,150+2,150-i); // y 軸メモリ
    }
    for (i=0;i<=120;i=i+40) {
        g.drawLine(150-2,150+i,150+2,150+i); // y 軸メモリ
    }
    g.drawString(" 5",150-15-3+3,110+5); // y 軸メモリ値
    g.drawString("10",150-15-3,70+5);
    g.drawString("15",150-15-3,30+5);
    g.drawString(" -5",150-20-6+3,190+5);
    g.drawString("-10",150-20-6,230+5);
    g.drawString("-15",150-20-6,270+5); // y 軸メモリ値

    g.drawLine(0,150,290,150); // x 軸
    g.drawLine(290,150,290-5,150-5);
    g.drawLine(290,150,290-5,150+5);
    g.drawString(" x ",290,145);
    for (i=0;i<=120;i=i+40) {
        g.drawLine(150+i,150-2,150+i,150+2); // x 軸メモリ
    }
    for (i=0;i<=120;i=i+40) {
        g.drawLine(150-i,150-2,150-i,150+2); // x 軸メモリ
    }
    g.drawString("1",190-2,165); // x 軸メモリ値
    g.drawString("2",230-2,165);
    g.drawString("3",270-2,165);
    g.drawString("0",150-10,165);
    g.drawString("-1",110-5,165);
    g.drawString("-2",70-5,165);
    g.drawString("-3",30-5,165); // x 軸メモリ値

    //グラフ描写
    //y=2(x+2)^2-5
    g.setColor(Color.red); //グラフを赤色にする

    Font f13=new Font((g.getFont()).getName(),Font.BOLD,14);
    g.setFont(f13);
    g.drawString("y=2(x+2)",30,210);

```

```

Font f14=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f14);
g.drawString("2",90-2,200);

Font f15=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f15);
g.drawString("-5",100-5,210);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x+2)*(x+2)-5;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

//y=2x^2
g.setColor(Color.blue); //グラフを赤色にする

Font f16=new Font((g.getFont()).getName(),Font.BOLD,14);
g.setFont(f16);
g.drawString("y=2x",220,120);

Font f17=new Font((g.getFont()).getName(),Font.BOLD,10);
g.setFont(f17);
g.drawString("2",250+2,110+2);

for (x=-4;x<=4.1;x=x+0.1) {
    y=2*(x-a)*(x-a)+b;
    px=(int)(150+40*x);
    py=(int)(150-8*y);
    if (fg==0) {
        g.drawLine(px,py,px,py);
    }
    else {
        g.drawLine(oldpx,oldpy,px,py);
    }
    oldpx=px;oldpy=py;
    fg++;
}
fg=0;

g.setColor(Color.black);
Font f18=new Font((g.getFont()).getName(),Font.BOLD,13);
g.setFont(f18);
g.drawString("Copyright(C) K.Niwa 2002.08",60,325); //作者名

for (t=1;t<100000;t++) {} //時間かせぎ

} //for (b=0;b>=-5;b=b-0.02)

} //else if (flag==1)

} //public void paint(Graphics g) //paint メソッド

```

```
//***** public static void main メソッド *****  
public static void main(String[] args) {  
    Frame w=new FKansuu1 ();  
    w.show ();  
} //public static void main(String[] args)  
  
} //public class FKansuu1 extends Frame
```