

```
////////////////////////////////////
//                                //
//          「Amidakuji」        //
//    Copyright (C) Sohun 9.11.2022    //
//                                //
////////////////////////////////////
```

```
//----- 各 パ ッ ケ ー ジ か ら ク ラ ス を 呼 び 込 む
```

```
-----
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;
```

```
//-----
```

```
public class MyAmidakujiEngj extends Frame{
    Button[] myBtn;           //ボタン型で宣言する
    Panel myPanel;          //パネル型で宣言する
    int px=71,py=10+20;     //当たり位置の x 座標と y
    座標に使用
    int xx,i,j;             //あみだの描写に使用に使用
    int flag=0;             // [開始] ボタンをクリック
    する前(0)か後(1)かに使用
    int flag1;              //あみだ横線の左端が縦線
    の奇数本目(1)か偶数本目(2)かに使用
    int flag2=0;           //既に、同じ位置にあみだ横
    線を引いている(1)か否(0)かに使用
    int flag3=0;
    int flag4=0;
    int ct=0;              //あみだ横線本数に使用
    int[] x=new int[51];   //あみだ横線の左端の x 座
    標に使用
    int[] y=new int[51];   //あみだ横線の左端の y 座
    標に使用
```

```

int[] yy=new int[21][11]; //あみだ横線の左端のソート後の y
座標に使用

long t; //時間稼ぎに使用
int tt; //ソートに使用
int r; //乱数
double r1,r2; //乱数
Image myImg; //イメージ型で宣言する
MediaTracker myMt; //メディアトラッカー型で
宣言する
int count; //ループカウンター
int b1=0,b2=0,b3=0,b4=0,b5=0,b6=0,b7=0,b8=0,b9=0,b10=0; //あみだ抽選最終位置のカ
ウントに使用
int k1=0,k2=0,k3=0,k4=0,k5=0,k6=0,k7=0,k8=0,k9=0,k10=0; //あみだ横線の両
端の y 座標の設定 yy[k*][j]に使用
int mini,kx,p; //あみだ抽選に使用
int caunt;
int N; //時間稼ぎ

//----- フ レ ー ム と イ ベ ン ト の 定 義
-----

public MyAmidakujiEngj() {

setSize(300-20,360+20); //フレームの大きさ
addWindowListener(new WindowAdapter(){ //フレームの定義
public void windowClosing(WindowEvent e){
System.exit(0);
}
});

myImg=Toolkit.getDefaultToolkit().getImage("tama.jpg"); //当たり位置の画像を読み
込む

myMt=new MediaTracker(this); //メディアトラッカーの実
体化

```

```

myMt.addImage(myImg,0); //メディアトラッ
カーに画像を貼り付ける

try //例外処理
{
myMt.waitForID(0);
}
catch(InterruptedException e)
{
}

myBtn=new Button[4]; //ボタンの実体化
myBtn[0]=new Button("Init");
myBtn[1]=new Button("High");
myBtn[2]=new Button("Medium");
myBtn[3]=new Button("Low");

myPanel=new Panel(); //パネルの実体化
myPanel.setLayout(new GridLayout(1,4));
for (count=0;count<=3;count++)
{
myPanel.add(myBtn[count]); //パネルにボタンを貼り付ける
}
setLayout(new BorderLayout());
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
public void actionPerformed(ActionEvent e){
flag=0;
ct=0; //あみだ横線本数
flag4=0; //あみだ抽選が一番下まで届いたか(99)否か(0)。
b1=0;b2=0;b3=0;b4=0;b5=0;b6=0;b7=0;b8=0;b9=0;b10=0;//あみだ抽選最終位置のカウン
トの初期化
k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;//あみだ横線の両端の y 座標の
設定 yy[k*]j]の初期化

```

```
caunt=1;
```

```
//あみだ横線の左端の座標の初期化
```

```
for (i=0;i<=50;i++)  
{  
x[i]=0;y[i]=0;  
}
```

```
//ソート後のあみだ横線の左端の y 座標の初期化
```

```
for (j=1;j<=10;j++)  
{  
for (i=1;i<=20;i++)  
{  
yy[i][j]=0;  
}  
}
```

```
repaint();  
}  
});
```

```
myBtn[1].addActionListener(new ActionListener(){ //高速アミダ ボタン
```

```
public void actionPerformed(ActionEvent e){
```

```
flag=1;
```

```
N=10000000;
```

```
ct=0; //あみだ横線本数
```

```
flag4=0;
```

```
b1=0;b2=0;b3=0;b4=0;b5=0;b6=0;b7=0;b8=0;b9=0;b10=0;//あみだ抽選最終位置のカウン  
トの初期化
```

```
k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;//あみだ横線の両端の y 座標の  
設定 yy[k*][j]の初期化
```

```
caunt=1;
```

```
//あみだ横線の左端の座標の初期化
```

```
for (i=0;i<=50;i++)
```

```
{  
x[i]=0;y[i]=0;  
}
```

//ソート後のあみだ横線の左端の y 座標の初期化

```
for (j=1;j<=10;j++)  
{  
for (i=1;i<=20;i++)  
{  
yy[i][j]=0;  
}  
}
```

```
repaint();  
}  
});
```

```
myBtn[2].addActionListener(new ActionListener(){ //中速アミダ ボタン
```

```
public void actionPerformed(ActionEvent e){
```

```
flag=2;
```

```
N=50000000;
```

```
ct=0; //あみだ横線本数
```

```
flag4=0;
```

```
b1=0;b2=0;b3=0;b4=0;b5=0;b6=0;b7=0;b8=0;b9=0;b10=0;//あみだ抽選最終位置のカウン  
トの初期化
```

```
k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;//あみだ横線の両端の y 座標の  
設定 yy[k*][j]の初期化
```

```
caunt=1;
```

//あみだ横線の左端の座標の初期化

```
for (i=0;i<=50;i++)  
{  
x[i]=0;y[i]=0;  
}
```

//ソート後のあみだ横線の左端の y 座標の初期化

```
for (j=1;j<=10;j++)
```

```
{
```

```
for (i=1;i<=20;i++)
```

```
{
```

```
yy[i][j]=0;
```

```
}
```

```
}
```

```
repaint();
```

```
}
```

```
});
```

```
myBtn[3].addActionListener(new ActionListener(){ //低速アミダ ボタン
```

```
public void actionPerformed(ActionEvent e){
```

```
flag=3;
```

```
N=90000000;
```

```
ct=0; //あみだ横線本数
```

```
flag4=0;
```

```
b1=0;b2=0;b3=0;b4=0;b5=0;b6=0;b7=0;b8=0;b9=0;b10=0;//あみだ抽選最終位置のカウン  
トの初期化
```

```
k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;//あみだ横線の両端の y 座標の  
設定 yy[k*][j]の初期化
```

```
caunt=1;
```

```
//あみだ横線の左端の座標の初期化
```

```
for (i=0;i<=50;i++)
```

```
{
```

```
x[i]=0;y[i]=0;
```

```
}
```

```
//ソート後のあみだ横線の左端の y 座標の初期化
```

```
for (j=1;j<=10;j++)
```

```
{
```

```
for (i=1;i<=20;i++)
```

```
{  
yy[i][j]=0;  
}  
}
```

```
repaint();  
}  
});
```

```
}//public FAmida()
```

```
//----- public void paint(Graphics g) の 開 始
```

```
-----  
public void paint(Graphics g)  
{  
if (myMt.checkID(0))  
{  
// [ 初 期 化 ] ボ タ ン を 押 し た と き  
*****  
if (flag==0)  
{  
g.clearRect(0,0,275,360+20); //全体クリア  
g.setColor(Color.black);  
g.drawString("Sohun 9.2022",110+70,340); //作者表示  
  
for (xx=25;xx<=250;xx=xx+25)  
{  
g.drawLine(xx,20+20,xx,135+20); //あみだ縦線の描写  
}  
  
g.drawImage(myImg,px,py,this); //当たり位置の描写  
g.drawLine(0,300+20,300-20,300+20); //仕切り線の描写  
} //if (flag==0)
```

```

// [ * 速 ア ミ タ ` ] ボ タ ン を 押 し た と き
*****

else if (flag==1 || flag==2 || flag==3)
{
g.setColor(Color.black);
g.drawString("Sohun 9.2022",110+70,340); //作者表示

k1=0;k2=0;k3=0;k4=0;k5=0;k6=0;k7=0;k8=0;k9=0;k10=0;//あみだ横線の両端の y 座標の
設定 yy[k*][j]の初期化
ct=0; //あみだ横線本数
flag=1; // [開始] ボタンをクリックする前(0)か後(1)か。
flag4=0;//

//あみだ横線の左端の座標の初期化
for (i=0;i<=50;i++)
{
x[i]=0;y[i]=0;
}

//ソート後のあみだ横線の左端の y 座標の初期化
for (j=1;j<=10;j++)
{
for (i=1;i<=20;i++)
{
yy[i][j]=0;
}
}

for (xx=25;xx<=250;xx=xx+25)
{
g.drawLine(xx,20+20,xx,135+20); //あみだ縦線の描写
}

g.drawImage(myImg,px,py,this); //当たりの位置
g.drawLine(0,300+20,300-20,300+20);//仕切り線の描写

```



```
//----- 左 から 右 へ あ み だ 横 線 の 描 写 開 始 ( 5 0 本 )
```

```
-----  
for (i=1;i<=50;i++)
```

```
{
```

```
//あみだ横線の左端の x 座標を乱数で選ぶ
```

```
r1=9*Math.random();
```

```
r=(int)r1;
```

```
//-----
```

```
switch (r)
```

```
{
```

```
case 0:
```

```
x[i]=25;
```

```
flag1=1; //あみだ横線の左端が縦線の奇数本目 flag1=1
```

```
break;
```

```
case 1:
```

```
x[i]=50;
```

```
flag1=2; //あみだ横線の左端が縦線の偶数本目 flag1=2
```

```
break;
```

```
case 2:
```

```
x[i]=75;
```

```
flag1=1;
```

```
break;
```

```
case 3:
```

```
x[i]=100;
```

```
flag1=2;
```

```
break;
```

```
case 4:
```

```
x[i]=125;
```

```
flag1=1;
```

```
break;
```

```
case 5:
```

```
x[i]=150;
```

```
flag1=2;
```

```
break;
```

```

case 6:
x[i]=175;
flag1=1;
break;
case 7:
x[i]=200;
flag1=2;
break;
case 8:
x[i]=225;
flag1=1;
break;
} //switch (r)
//-----

//あみだ横線の左端の y 座標を乱数で選ぶ
r2=Math.random();
//-----

switch (flag1)
{
//あみだ横線の左端の縦線が奇数番目のとき-----
case 1:
if (r2<0.1)
{
y[i]=30+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;//既に、同じ位置にあみだ横線を引いているとき
}
}
}
else if (r2<0.2)
{

```

```
y[i]=40+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
else if (r2<0.3)
{
y[i]=50+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.4)
{
y[i]=60+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.5)
{
y[i]=70+20;
for(j=1;j<i;j++)
{
```

```
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.6)
{
y[i]=80+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.7)
{
y[i]=90+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.8)
{
y[i]=100+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
```

```

}
}
}
else if (r2<0.9)
{
y[i]=110+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<1)
{
y[i]=120+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
break;

//-----
//あみだ横線の左端の縦線が偶数番目のとき
case 2:
if (r2<0.1)
{
y[i]=35+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])

```

```
{
flag2=1;
}
}
}
else if (r2<0.2)
{
y[i]=45+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.3)
{
y[i]=55+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.4)
{
y[i]=65+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
```

```
}  
}  
else if (r2<0.5)  
{  
y[i]=75+20;  
for(j=1;j<i;j++)  
{  
if(x[j]==x[i] && y[j]==y[i])  
{  
flag2=1;  
}  
}  
}  
else if (r2<0.6)  
{  
y[i]=85+20;  
for(j=1;j<i;j++)  
{  
if(x[j]==x[i] && y[j]==y[i])  
{  
flag2=1;  
}  
}  
}  
else if (r2<0.7)  
{  
y[i]=95+20;  
for(j=1;j<i;j++)  
{  
if(x[j]==x[i] && y[j]==y[i])  
{  
flag2=1;  
}  
}  
}  
else if (r2<0.8)
```

```
{
y[i]=105+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<0.9)
{
y[i]=115+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
else if (r2<1)
{
y[i]=125+20;
for(j=1;j<i;j++)
{
if(x[j]==x[i] && y[j]==y[i])
{
flag2=1;
}
}
}
break;
} //switch (flag1)
//-----
```



```

if (flag2==1)    //既に、同じあみだ横線があったとき-----
{
i=i-1;    //あみだ横線の左端座標 x[i],y[i]をもう一度探す
flag2=0;
}
else if (flag2==0) //同じあみだ横線がなかったとき-----
{
ct=ct+1; //あみだ横線本数-----
//g.setColor(Color.red);
g.drawLine(x[i],y[i],x[i]+25,y[i]); //左から右へ横線を引く
//g.setColor(Color.black);

//描写済みのあみだ横線本数の表示-----
//g.clearRect(10,310,100,20);
//g.drawString("あみだ横線本数 "+ct,10,320);
for (t=1;t<=100000;t++){           //時間かせぎ
}
}
//----- あみだ横線の描写終了 ( 5 0 本 )
-----

```

```

//あみだ横線の両端の y 座標の設定 yy[i][j]の開始-----
for (i=1;i<=50;i++)
{
if (x[i]==25)
{
k1=k1+1;
yy[k1][1]=y[i];
k2=k2+1;
yy[k2][2]=y[i];
}
else if (x[i]==50)

```

```
{
k2=k2+1;
yy[k2][2]=y[i];
k3=k3+1;
yy[k3][3]=y[i];
}
else if (x[i]==75)
{
k3=k3+1;
yy[k3][3]=y[i];
k4=k4+1;
yy[k4][4]=y[i];
}
else if (x[i]==100)
{
k4=k4+1;
yy[k4][4]=y[i];
k5=k5+1;
yy[k5][5]=y[i];
}
else if (x[i]==125)
{
k5=k5+1;
yy[k5][5]=y[i];
k6=k6+1;
yy[k6][6]=y[i];
}
else if (x[i]==150)
{
k6=k6+1;
yy[k6][6]=y[i];
k7=k7+1;
yy[k7][7]=y[i];
}
else if (x[i]==175)
{
```

```

k7=k7+1;
yy[k7][7]=y[i];
k8=k8+1;
yy[k8][8]=y[i];
}
else if (x[i]==200)
{
k8=k8+1;
yy[k8][8]=y[i];
k9=k9+1;
yy[k9][9]=y[i];
}
else if (x[i]==225)
{
k9=k9+1;
yy[k9][9]=y[i];
k10=k10+1;
yy[k10][10]=y[i];
}
} //あみだ横線の両端の y 座標の設定 yy[i][j]の終了-----

```

```

//あみだ横線の両端の y 座標のソート（昇順）開始-----

```

```

for (j=1;j<=19;j++)
{
for (i=j+1;i<=20;i++)
{
if (yy[i][1]<yy[j][1])
{
tt=yy[i][1];
yy[i][1]=yy[j][1];
yy[j][1]=tt;
}
}
}

```

```
}
for (j=1;j<=19;j++)
{
for (i=j+1;i<=20;i++)
{
if (yy[i][2]<yy[j][2])
{
tt=yy[i][2];
yy[i][2]=yy[j][2];
yy[j][2]=tt;
}
}
}
for (j=1;j<=19;j++)
{
for (i=j+1;i<=20;i++)
{
if (yy[i][3]<yy[j][3])
{
tt=yy[i][3];
yy[i][3]=yy[j][3];
yy[j][3]=tt;
}
}
}
for (j=1;j<=19;j++)
{
for (i=j+1;i<=20;i++)
{
if (yy[i][4]<yy[j][4])
{
tt=yy[i][4];
yy[i][4]=yy[j][4];
yy[j][4]=tt;
}
}
}
}
```

```
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][5]<yy[j][5])  
{  
tt=yy[i][5];  
yy[i][5]=yy[j][5];  
yy[j][5]=tt;  
}  
}  
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][6]<yy[j][6])  
{  
tt=yy[i][6];  
yy[i][6]=yy[j][6];  
yy[j][6]=tt;  
}  
}  
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][7]<yy[j][7])  
{  
tt=yy[i][7];  
yy[i][7]=yy[j][7];  
yy[j][7]=tt;  
}  
}  
}
```

```
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][8]<yy[j][8])  
{  
tt=yy[i][8];  
yy[i][8]=yy[j][8];  
yy[j][8]=tt;  
}  
}  
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][9]<yy[j][9])  
{  
tt=yy[i][9];  
yy[i][9]=yy[j][9];  
yy[j][9]=tt;  
}  
}  
}  
for (j=1;j<=19;j++)  
{  
for (i=j+1;i<=20;i++)  
{  
if (yy[i][10]<yy[j][10])  
{  
tt=yy[i][10];  
yy[i][10]=yy[j][10];  
yy[j][10]=tt;  
}  
}  
}
```

```
}//あみだ横線の両端の y 座標のソート（昇順）終了-----
```

```
//-----あみだ抽選開始-----
```

```
mini=20+20;kx=75;p=3;//現時点での x 座標 kx , 現時点での縦線の位置 p
```

```
g.setColor(Color.red);
```

```
do{
```

```
  i=1;flag3=0;
```

```
  do{
```

```
    if (p%2==1)//縦線が奇数番目のとき-----
```

```
    {
```

```
      if(yy[i][p]>mini)
```

```
      {
```

```
        g.drawLine(kx,mini,kx,yy[i][p]);
```

```
        if (yy[i][p]%10== 0)
```

```
        {
```

```
          g.drawLine(kx,yy[i][p],kx+25,yy[i][p]);
```

```
          mini=yy[i][p];
```

```
          kx=kx+25;
```

```
          p=p+1;
```

```
        }
```

```
      else if (yy[i][p]%10!=0)
```

```
      {
```

```
        g.drawLine(kx,yy[i][p],kx-25,yy[i][p]);
```

```
        mini=yy[i][p];
```

```
        kx=kx-25;
```

```
        p=p-1;
```

```
      }
```

```
      flag3=99;
```

```
    }
```

```
  else
```

```
  {
```

```
    i=i+1;
```

```
if (i>20)
{
g.drawLine(kx,mini,kx,135+20);
g.clearRect(10,310+20,100,20);
g.setColor(Color.black);
g.drawString("Number of experiments "+caunt,10+10,320+20);

//----- 抽選位置の判断の開始 -----
switch (kx)
{
case 25:
b1=b1+1;
break;

case 50:
b2=b2+1;
break;
case 75:
b3=b3+1;
break;
case 100:
b4=b4+1;
break;
case 125:
b5=b5+1;
break;
case 150:
b6=b6+1;
break;
case 175:
b7=b7+1;
break;
case 200:
b8=b8+1;
break;
case 225:
```



```

b9=b9+1;
break;
case 250:
b10=b10+1;
break;
}
//----- 抽選位置の判断の終了 -----

//棒グラフの描写 -----
g.drawRect(25-5,20+300-10*b1,10,10*b1);
g.drawRect(50-5,20+300-10*b2,10,10*b2);
g.drawRect(75-5,20+300-10*b3,10,10*b3);
g.drawRect(100-5,20+300-10*b4,10,10*b4);
g.drawRect(125-5,20+300-10*b5,10,10*b5);
g.drawRect(150-5,20+300-10*b6,10,10*b6);
g.drawRect(175-5,20+300-10*b7,10,10*b7);
g.drawRect(200-5,20+300-10*b8,10,10*b8);
g.drawRect(225-5,20+300-10*b9,10,10*b9);
g.drawRect(250-5,20+300-10*b10,10,10*b10);

for (t=1;t<=N;t++){ //時間稼ぎ+++++

if (caunt<81) //実験回数
{
caunt++;
repaint();
}
flag4=99;
}
}

else if (p%2==0)//縦線が奇数番目のとき-----
{
if(yy[i][p]>mini)
{
g.drawLine(kx,mini,kx,yy[i][p]);

```

```

if (yy[i][p]%10!= 0)
{
g.drawLine(kx,yy[i][p],kx+25,yy[i][p]);
mini=yy[i][p];
kx=kx+25;
p=p+1;
}
else if (yy[i][p]%10==0)
{
g.drawLine(kx,yy[i][p],kx-25,yy[i][p]);
mini=yy[i][p];
kx=kx-25;
p=p-1;
}
flag3=99;
}
else
{
i=i+1;
if (i>20)
{
g.drawLine(kx,mini,kx,135+20);
g.clearRect(10,310+20,100,20);
//g.clearRect(10+10,310+10+20,100,20);
g.setColor(Color.black);
g.drawString("Number of experiments "+caunt,10+10,320+20);

//----- 抽選位置の判断の開始 -----
switch (kx)
{
case 25:
b1=b1+1;
break;
case 50:
b2=b2+1;
break;

```

```
case 75:
b3=b3+1;
break;
case 100:
b4=b4+1;
break;
case 125:
b5=b5+1;
break;
case 150:
b6=b6+1;
break;
case 175:
b7=b7+1;
break;
case 200:
b8=b8+1;
break;
case 225:
b9=b9+1;
break;
case 250:
b10=b10+1;
break;
}
//----- 抽選位置の判断の終了 -----
```

```
//棒グラフの描写
```

```
g.drawRect(25-5,20+300-10*b1,10,10*b1);
g.drawRect(50-5,20+300-10*b2,10,10*b2);
g.drawRect(75-5,20+300-10*b3,10,10*b3);
g.drawRect(100-5,20+300-10*b4,10,10*b4);
g.drawRect(125-5,20+300-10*b5,10,10*b5);
g.drawRect(150-5,20+300-10*b6,10,10*b6);
g.drawRect(175-5,20+300-10*b7,10,10*b7);
g.drawRect(200-5,20+300-10*b8,10,10*b8);
```

