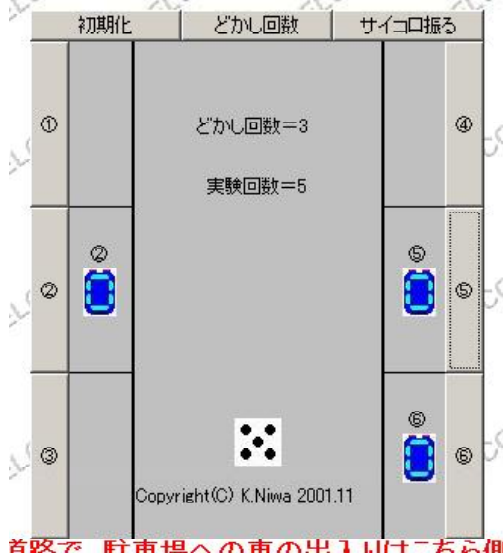


【駐車場モデルの実験】

1. はじめに



[J a v a アプレット]



[J a v a アプリケーション]

【問 題】

1階が左右それぞれ3台ずつ計6台の自動車を停めることのできる駐車場をもったビルがあります。駐車場の出入り口は、道路に面していて、そこからしか車の出入りができません。また、駐車場は左右に区切られているので、駐車場の中での車の左右の移動はできません。例えば、駐車場左側一番奥に車1が停まっていて、駐車場左側一番道路側に車2が停まっているとします。車1が駐車場を出るのに、車2が邪魔になるのでどけてもらわなければなりません。また、車が入るときにも同様なことが起こります。このビルの住人が6台の車をこの駐車場から出入りさせるとき、駐車位置を指定して固定して使ったほうが良いのか、それとも駐車位置を固定せずに奥から詰めて駐車して使ったほうが良いのかを考えてください。

【実験の仕方】

1. 駐車位置を固定して使用する場合

まず、[サイコロ振る]をクリックします。
 例えば、サイコロの目が3で、駐車場③が空きの場合は [③] ボタンをクリックして、駐車場③に車③を停めます。
 サイコロの目が3で、駐車場③に車③が停まっている場合は [③] ボタンをクリックして、駐車場③から車③を出します。
 駐車場③から車③を出すには [③] ボタンを6回クリックします。④→⑤→⑥→①→②→空白(車が出た状態を表す)と変わるので、空白の状態にしてください。
 駐車している他の車が邪魔で、どけてもらわなければ車の出入りができないときは、[どかし回数]をクリックしてカウントします。(ただし、2台の車をどける場合も、どかし回数は1とします。)
 そして、これを何度も繰り返します。

2. 駐車位置を固定せずに使用する場合

まず、[サイコロ振る]をクリックします。
 例えば、サイコロの目が3で、車③が停まっていない場合は、駐車場の奥から詰めて空いているところに車③を停めます。
 例えば、左一番奥の駐車場①に車③を停めるには、[①] ボタンを3回クリックします。①→②→③(車③が停まった状態を表す)と変わるので③の状態にしてください。これで、駐車場①に車③が停まったこととなります。
 サイコロの目が3で、駐車場①に車③が停まっている場合は、駐車場①から車③を出します。
 駐車場①から車③を出すには [①] ボタンを4回クリックします。④→⑤→⑥→空白(車が出た状態)と変わるので、空白の状態にしてください。
 駐車している他の車が邪魔で、どけてもらわなければ車の出入りができないときは、[どかし回数]

をクリックしてカウントします。(ただし、2台の車をどける場合も、どかし回数は1とします。)そして、これを何度も繰り返します。
シミュレーションソフト「駐車場モデルの実験」を使って、上の問題を考えてみてください。

2. Java アプレット

(1) Java プログラムリスト

```

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                                                                           //
//                                                                                                           //
//                                                                                                           //
//                                                                                                           //
//                                                                                                           //
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

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

public class Tyuusya extends Applet{

    Button[] myBtnN;           //初期化ボタン、どかし回数ボタン、サイコロを振るボタン
    Button[] myBtnW;           //①ボタン、②ボタン、③ボタン
    Button[] myBtnE;           //④ボタン、⑤ボタン、⑥ボタン
    Panel myPanelN;           //初期化・どかし回数・サイコロを振るボタン用パネル
    Panel myPanelW;           //①・②・③ボタン用パネル
    Panel myPanelE;           //④・⑤・⑥ボタン用パネル
    int imgNo;
    Image[] img;               //自動車、サイコロ画像
    MediaTracker myMt;         //myMt をメディアトラッカー型で宣言する
    int kk=0;                  //どかし回数
    int ct=0;                  //実験回数
    int k;                     //カウンター
    int count;                 //カウンター
    int flagN=0,flagW=999,flagE=999;
    int flag1=0,flag2=0,flag3=0,flag4=0,flag5=0,flag6=0;
    long t;                    //時間かせぎ
    int r=1;                   //乱数

    //public void init() の開始*****
    public void init () {
        setBackground (Color.lightGray);           //背景色の設定

        //サイコロ画像と車画像の準備
        imgNo=0;
        img=new Image [7];
        img [0]=getImage (getDocumentBase (), "car.jpg");           //自動車画像
        img [1]=getImage (getDocumentBase (), "sai1.jpg");           //サイコロの目 1 画像
        img [2]=getImage (getDocumentBase (), "sai2.jpg");           //サイコロの目 2 画像
        img [3]=getImage (getDocumentBase (), "sai3.jpg");           //サイコロの目 3 画像
        img [4]=getImage (getDocumentBase (), "sai4.jpg");           //サイコロの目 4 画像
        img [5]=getImage (getDocumentBase (), "sai5.jpg");           //サイコロの目 5 画像
        img [6]=getImage (getDocumentBase (), "sai6.jpg");           //サイコロの目 6 画像

        myMt=new MediaTracker (this);           //メディアトラッカーの実体化
        for (count=0;count<=6;count++) {
            myMt.addImage (img [count],0);           //メディアトラッカーに画像を追加
        }
        try {
            myMt.waitForID (0);
        }
    }
}

```

```

        catch (InterruptedException e) {
        }

//North ボタンの準備
myBtnN=new Button[3]; //ボタンの実体化
myBtnN[0]=new Button("初期化");
myBtnN[1]=new Button("どかし回数");
myBtnN[2]=new Button("サイコロ振る");
myPanelN=new Panel(); //パネルの実体化
myPanelN.setLayout(new GridLayout(1,3)); //パネルをグリッドレイアウトにする
for (count=0;count<=2;count++){
    myPanelN.add(myBtnN[count]); //パネルにボタンを貼り付ける
}

//West ボタンの準備
myBtnW=new Button[3]; //ボタンの実体化
myBtnW[0]=new Button("①");
myBtnW[1]=new Button("②");
myBtnW[2]=new Button("③");
myPanelW=new Panel(); //パネルの実体化
myPanelW.setLayout(new GridLayout(3,1)); //パネルをグリッドレイアウトにする
for (count=0;count<=2;count++){
    myPanelW.add(myBtnW[count]); //パネルにボタンを貼り付ける
}

//East ボタンの準備
myBtnE=new Button[3]; //ボタンの実体化
myBtnE[0]=new Button("④");
myBtnE[1]=new Button("⑤");
myBtnE[2]=new Button("⑥");
myPanelE=new Panel(); //パネルの実体化
myPanelE.setLayout(new GridLayout(3,1)); //パネルをグリッドレイアウトにする
for (count=0;count<=2;count++){
    myPanelE.add(myBtnE[count]); //パネルにボタンを貼り付ける
}

//全体をボーダーレイアウトに設定する
setLayout(new BorderLayout()); //全体をボーダーレイアウトにする
add("North",myPanelN); //パネルを北に貼り付ける
add("West",myPanelW); //パネルを西に貼り付ける
add("East",myPanelE); //パネルを東に貼り付ける

//North ボタンのイベント処理
myBtnN[0].addActionListener(new ActionListener() { //初期化
    public void actionPerformed(ActionEvent e) {
        flagN=0;flagW=999;flagE=999;
        flag1=0;flag2=0;flag3=0;flag4=0;flag5=0;flag6=0;
        kk=0; //どかし回数の初期化
        ct=0; //実験回数の初期化
        r=1; //サイコロの目の初期化
        repaint();//paint()メソッドを呼び出す
    }
});

myBtnN[1].addActionListener(new ActionListener() { //どかし回数
    public void actionPerformed(ActionEvent e) {
        kk=kk+1;
        flagN=1;
        repaint();//paint()メソッドを呼び出す
    }
});

myBtnN[2].addActionListener(new ActionListener() { //サイコロを振る
    public void actionPerformed(ActionEvent e) {
        ct=ct+1;

```

```

        flagN=2;
        repaint();
    }
});

//West ボタンのイベント処理
myBtnW[0].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=0;
        flagN=99;
        repaint();
    }
});
myBtnW[1].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=1;
        flagN=99;
        repaint();
    }
});
myBtnW[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=2;
        flagN=99;
        repaint();
    }
});

//East ボタンのイベント処理
myBtnE[0].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=0;
        flagN=99;
        repaint();
    }
});
myBtnE[1].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=1;
        flagN=99;
        repaint();
    }
});
myBtnE[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=2;
        flagN=99;
        repaint();
    }
});
});

}

//public void paint(Graphics g) の開始*****
public void paint(Graphics g) {
    if (myMt.checkID(0)) {

        //駐車場描写
        g.drawLine(67,0,67,350);g.drawLine(68,0,68,350);
        g.drawLine(233,0,233,350);g.drawLine(232,0,232,350);
        g.drawLine(0,130-1,67,130-1);g.drawLine(0,240-1,67,240-1);
        g.drawLine(233,130-1,300,130-1);g.drawLine(233,240-1,300,240-1);
    }
}

```

```

//駐車場番号表示
/*g.drawString("①",40,40);
g.drawString("②",40,150);
g.drawString("③",40,260);
g.drawString("④",253-3,40);
g.drawString("⑤",253-3,150);
g.drawString("⑥",253-3,260);*/

g.drawString("どかし回数="+kk,110,80);
g.drawString("実験回数="+ct,120-3,120);

//North パネル
//初期状態または初期化ボタンを押したとき
if (flagN==0) {
    g.clearRect(0,0,300,350); //全体クリア

    //駐車場描写
    g.drawLine(67,0,67,350);g.drawLine(68,0,68,350);
    g.drawLine(233,0,233,350);g.drawLine(232,0,232,350);
    g.drawLine(0,130-1,67,130-1);g.drawLine(0,240-1,67,240-1);
    g.drawLine(233,130-1,300,130-1);g.drawLine(233,240-1,300,240-1);

    g.drawString("どかし回数="+kk,110,80);
    g.drawString("実験回数="+ct,120-3,120);

    //駐車場番号表示
    /*g.drawString("①",40,40+10);
    g.drawString("②",40,150+10);
    g.drawString("③",40,260+10);
    g.drawString("④",253-3,40+10);
    g.drawString("⑤",253-3,150+10);
    g.drawString("⑥",253-3,260+10);*/

    //サイコロ表示
    g.drawImage(img[1],135,270,this);
}

//どかし回数ボタンを押したとき
else if (flagN==1) {
    switch (r) {
        case 1:
            g.drawImage(img[1],135,270,this);
            break;
        case 2:
            g.drawImage(img[2],135,270,this);
            break;
        case 3:
            g.drawImage(img[3],135,270,this);
            break;
        case 4:
            g.drawImage(img[4],135,270,this);
            break;
        case 5:
            g.drawImage(img[5],135,270,this);
            break;
        case 6:
            g.drawImage(img[6],135,270,this);
            break;
    }
}

//サイコロ振るボタンを押したとき
else if (flagN==2) {
    g.drawString("Copyright(C) K.Niwa 2001.11",80-10,325);
}

```

```

//自動車の表示・消去
if (flag1!=0) {
    g.drawImage(img[0],40-5,60,this); //自動車①の表示
}
else if (flag1==0) {
    g.clearRect(40-5,60,22,32); //自動車①の消去
}
if (flag2!=0) {
    g.drawImage(img[0],40-5,170,this); //自動車②の表示
}
else if (flag2==0) {
    g.clearRect(40-5,170,22,32); //自動車②の消去
}
if (flag3!=0) {
    g.drawImage(img[0],40-5,280,this); //自動車③の表示
}
else if (flag3==0) {
    g.clearRect(40-5,280,22,32); //自動車③の消去
}
if (flag4!=0) {
    g.drawImage(img[0],238+5+3,60,this); //自動車④の表示
}
else if (flag4==0) {
    g.clearRect(238+5+3,60,22,32); //自動車④の消去
}
if (flag5!=0) {
    g.drawImage(img[0],238+5+3,170,this); //自動車⑤の表示
}
else if (flag5==0) {
    g.clearRect(238+5+3,170,22,32); //自動車⑤の消去
}
if (flag6!=0) {
    g.drawImage(img[0],238+5+3,280,this); //自動車⑥の表示
}
else if (flag6==0) {
    g.clearRect(238+5+3,280,22,32); //自動車⑥の消去
}
}

```

//番号の表示

```

//駐車場 1 の番号の表示
if (flag1==1) {
    g.drawString("①",40,40+15);
}
else if (flag1==2) {
    g.drawString("②",40,40+15);
}
else if (flag1==3) {
    g.drawString("③",40,40+15);
}
else if (flag1==4) {
    g.drawString("④",40,40+15);
}
else if (flag1==5) {
    g.drawString("⑤",40,40+15);
}
else if (flag1==6) {
    g.drawString("⑥",40,40+15);
}

//駐車場 2 の番号の表示
if (flag2==1) {
    g.drawString("②",40,150+15);
}

```

```

else if (flag2==2) {
    g.drawString("③",40,150+15);
}
else if (flag2==3) {
    g.drawString("④",40,150+15);
}
else if (flag2==4) {
    g.drawString("⑤",40,150+15);
}
else if (flag2==5) {
    g.drawString("⑥",40,150+15);
}
else if (flag2==6) {
    g.drawString("①",40,150+15);
}
//駐車場 3 の番号の表示
if (flag3==1) {
    g.drawString("③",40,260+15);
}
else if (flag3==2) {
    g.drawString("④",40,260+15);
}
else if (flag3==3) {
    g.drawString("⑤",40,260+15);
}
else if (flag3==4) {
    g.drawString("⑥",40,260+15);
}
else if (flag3==5) {
    g.drawString("①",40,260+15);
}
else if (flag3==6) {
    g.drawString("②",40,260+15);
}
//駐車場 4 の番号の表示
if (flag4==1) {
    g.drawString("④",250,40+15);
}
else if (flag4==2) {
    g.drawString("⑤",250,40+15);
}
else if (flag4==3) {
    g.drawString("⑥",250,40+15);
}
else if (flag4==4) {
    g.drawString("①",250,40+15);
}
else if (flag4==5) {
    g.drawString("②",250,40+15);
}
else if (flag4==6) {
    g.drawString("③",250,40+15);
}
//駐車場 5 の番号の表示
if (flag5==1) {
    g.drawString("⑤",250,150+15);
}
else if (flag5==2) {
    g.drawString("⑥",250,150+15);
}
else if (flag5==3) {
    g.drawString("①",250,150+15);
}
}

```

```

else if (flag5==4) {
    g.drawString ("②",250,150+15);
}
else if (flag5==5) {
    g.drawString ("③",250,150+15);
}
else if (flag5==6) {
    g.drawString ("④",250,150+15);
}
//駐車場 6 の番号の表示
if (flag6==1) {
    g.drawString ("⑥",250,260+15);
}
else if (flag6==2) {
    g.drawString ("①",250,260+15);
}
else if (flag6==3) {
    g.drawString ("②",250,260+15);
}
else if (flag6==4) {
    g.drawString ("③",250,260+15);
}
else if (flag6==5) {
    g.drawString ("④",250,260+15);
}
else if (flag6==6) {
    g.drawString ("⑤",250,260+15);
}

//サイコロパラパラ表示
for (k=1;k<=10;k++) {
    for (count=1;count<=6;count++) {
        g.drawImage (img [count],135,270,this);
        for (t=1;t<=500000;t++) {} //時間かせぎ
    }
}

//サイコロの目の判断
r=(int) (6*Math.random ()+1);
switch (r) {
    case 1:
        g.drawImage (img [1],135,270,this);
        break;
    case 2:
        g.drawImage (img [2],135,270,this);
        break;
    case 3:
        g.drawImage (img [3],135,270,this);
        break;
    case 4:
        g.drawImage (img [4],135,270,this);
        break;
    case 5:
        g.drawImage (img [5],135,270,this);
        break;
    case 6:
        g.drawImage (img [6],135,270,this);
        break;
}
} //else if (flagN==2)

```



```

//West パネル
//①ボタンを押したとき
if (flagW==0) {
    flag1=flag1+1;
    if (flag1==7) {
        flag1=0;
    }
    flagW=99;
}
//②ボタンを押したとき
else if (flagW==1) {
    flag2=flag2+1;
    if (flag2==7) {
        flag2=0;
    }
    flagW=99;
}
//③ボタンを押したとき
else if (flagW==2) {
    flag3=flag3+1;
    if (flag3==7) {
        flag3=0;
    }
    flagW=99;
}
}
//East パネル
//④ボタンを押したとき
if (flagE==0) {
    flag4=flag4+1;
    if (flag4==7) {
        flag4=0;
    }
    flagE=99;
}
//⑤ボタンを押したとき
else if (flagE==1) {
    flag5=flag5+1;
    if (flag5==7) {
        flag5=0;
    }
    flagE=99;
}
//⑥ボタンを押したとき
else if (flagE==2) {
    flag6=flag6+1;
    if (flag6==7) {
        flag6=0;
    }
    flagE=99;
}
}
//自動車の表示・消去
if (flag1!=0) {
    g.drawImage(img[0],40-5,60,this); //自動車①の表示
}
else if (flag1==0) {
    g.clearRect(40-5,60,22,32); //自動車①の消去
}
if (flag2!=0) {
    g.drawImage(img[0],40-5,170,this); //自動車②の表示
}
else if (flag2==0) {
    g.clearRect(40-5,170,22,32); //自動車②の消去
}
}

```

```

if (flag3!=0) {
    g.drawImage(img[0],40-5,280,this); //自動車③の表示
}
else if (flag3==0) {
    g.clearRect(40-5,280,22,32); //自動車③の消去
}
if (flag4!=0) {
    g.drawImage(img[0],238+5+3,60,this); //自動車④の表示
}
else if (flag4==0) {
    g.clearRect(238+5+3,60,22,32); //自動車④の消去
}
if (flag5!=0) {
    g.drawImage(img[0],238+5+3,170,this); //自動車⑤の表示
}
else if (flag5==0) {
    g.clearRect(238+5+3,170,22,32); //自動車⑤の消去
}
if (flag6!=0) {
    g.drawImage(img[0],238+5+3,280,this); //自動車⑥の表示
}
else if (flag6==0) {
    g.clearRect(238+5+3,280,22,32); //自動車⑥の消去
}

//番号の表示
//駐車場 1 の番号の表示
if (flag1==1) {
    g.drawString("①",40,40+15);
}
else if (flag1==2) {
    g.drawString("②",40,40+15);
}
else if (flag1==3) {
    g.drawString("③",40,40+15);
}
else if (flag1==4) {
    g.drawString("④",40,40+15);
}
else if (flag1==5) {
    g.drawString("⑤",40,40+15);
}
else if (flag1==6) {
    g.drawString("⑥",40,40+15);
}
//駐車場 2 の番号の表示
if (flag2==1) {
    g.drawString("②",40,150+15);
}
else if (flag2==2) {
    g.drawString("③",40,150+15);
}
else if (flag2==3) {
    g.drawString("④",40,150+15);
}
else if (flag2==4) {
    g.drawString("⑤",40,150+15);
}
else if (flag2==5) {
    g.drawString("⑥",40,150+15);
}
else if (flag2==6) {
    g.drawString("①",40,150+15);
}
}

```

```

//駐車場 3 の番号の表示
if (flag3==1) {
    g.drawString ("③",40,260+15);
}
else if (flag3==2) {
    g.drawString ("④",40,260+15);
}
else if (flag3==3) {
    g.drawString ("⑤",40,260+15);
}
else if (flag3==4) {
    g.drawString ("⑥",40,260+15);
}
else if (flag3==5) {
    g.drawString ("①",40,260+15);
}
else if (flag3==6) {
    g.drawString ("②",40,260+15);
}
//駐車場 4 の番号の表示
if (flag4==1) {
    g.drawString ("④",250,40+15);
}
else if (flag4==2) {
    g.drawString ("⑤",250,40+15);
}
else if (flag4==3) {
    g.drawString ("⑥",250,40+15);
}
else if (flag4==4) {
    g.drawString ("①",250,40+15);
}
else if (flag4==5) {
    g.drawString ("②",250,40+15);
}
else if (flag4==6) {
    g.drawString ("③",250,40+15);
}
//駐車場 5 の番号の表示
if (flag5==1) {
    g.drawString ("⑤",250,150+15);
}
else if (flag5==2) {
    g.drawString ("⑥",250,150+15);
}
else if (flag5==3) {
    g.drawString ("①",250,150+15);
}
else if (flag5==4) {
    g.drawString ("②",250,150+15);
}
else if (flag5==5) {
    g.drawString ("③",250,150+15);
}
else if (flag5==6) {
    g.drawString ("④",250,150+15);
}
//駐車場 6 の番号の表示
if (flag6==1) {
    g.drawString ("⑥",250,260+15);
}
else if (flag6==2) {
    g.drawString ("①",250,260+15);
}
}

```

```

else if (flag6==3) {
    g.drawString ("②",250,260+15);
}
else if (flag6==4) {
    g.drawString ("③",250,260+15);
}
else if (flag6==5) {
    g.drawString ("④",250,260+15);
}
else if (flag6==6) {
    g.drawString ("⑤",250,260+15);
}

//サイコロの表示
switch (r) {
case 1:
    g.drawImage (img [1],135,270,this);
    break;
case 2:
    g.drawImage (img [2],135,270,this);
    break;
case 3:
    g.drawImage (img [3],135,270,this);
    break;
case 4:
    g.drawImage (img [4],135,270,this);
    break;
case 5:
    g.drawImage (img [5],135,270,this);
    break;
case 6:
    g.drawImage (img [6],135,270,this);
    break;
}
g.drawString ("Copyright (C) K.Niwa 2001.11",80-10,325); //作者の表示
}
}
}

```

(2) HTMLリスト

```

<HTML>
<HEAD>
< !-----
                「駐車場モデルの実験」
                Copyright (C) K.Niwa 2001.11.14
                ----->
</HEAD>
<BODY>
    <CENTER>
        <B>「駐車場モデルの実験」</B>
        <BR><BR>
        <APPLET CODE="Tyuusya.class" WIDTH="300" HEIGHT="350"></APPLET>
        <BR><BR>
    </CENTER>
</BODY>
</HTML>

```



```

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

//North ボタンの準備
myBtnN=new Button [3];           //ボタンの実態化
myBtnN[0]=new Button ("初期化");
myBtnN[1]=new Button ("どかし回数");
myBtnN[2]=new Button ("サイコロ振る");

myPanelN=new Panel ();           //パネルの実体化
myPanelN.setLayout (new GridLayout (1,3)); //パネルをグリッドレイアウトにする
for (count=0;count<=2;count++) {
    myPanelN.add (myBtnN[count]); //パネルにボタンを貼り付ける
}

//West ボタンの準備
myBtnW=new Button [3];           //ボタンの実態化
myBtnW[0]=new Button ("①");
myBtnW[1]=new Button ("②");
myBtnW[2]=new Button ("③");

myPanelW=new Panel ();           //パネルの実体化
myPanelW.setLayout (new GridLayout (3,1)); //パネルをグリッドレイアウトにする
for (count=0;count<=2;count++) {
    myPanelW.add (myBtnW[count]); //パネルにボタンを貼り付ける
}

//East ボタンの準備
myBtnE=new Button [3];
myBtnE[0]=new Button ("④");
myBtnE[1]=new Button ("⑤");
myBtnE[2]=new Button ("⑥");
myPanelE=new Panel ();
myPanelE.setLayout (new GridLayout (3,1));

for (count=0;count<=2;count++) {
    myPanelE.add (myBtnE[count]);
}

//全体をボーダーレイアウトに設定する
setLayout (new BorderLayout ());
add ("North",myPanelN);
add ("West",myPanelW);
add ("East",myPanelE);

//North ボタンのイベント処理
myBtnN[0].addActionListener (new ActionListener () { //初期化
    public void actionPerformed (ActionEvent e) {
        flagN=0;flagW=999;flagE=999;
        flag1=0;flag2=0;flag3=0;flag4=0;flag5=0;flag6=0;
        kk=0; //どかし回数の初期化
        ct=0; //実験回数の初期化
        r=1; //サイコロの目の初期化
        repaint ();
    }
});
myBtnN[1].addActionListener (new ActionListener () { //どかし回数
    public void actionPerformed (ActionEvent e) {
        kk=kk+1;
        flagN=1;
        repaint ();
    }
});

```

```

    }
});
myBtnN[2].addActionListener(new ActionListener() { //サイコロを振る
    public void actionPerformed(ActionEvent e) {
        ct=ct+1;
        flagN=2;
        repaint();
    }
});

//West ボタンのイベント処理
myBtnW[0].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=0;
        flagN=99;
        repaint();
    }
});
myBtnW[1].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=1;
        flagN=99;
        repaint();
    }
});
myBtnW[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagW=2;
        flagN=99;
        repaint();
    }
});

//East ボタンのイベント処理
myBtnE[0].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=0;
        flagN=99;
        repaint();
    }
});
myBtnE[1].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=1;
        flagN=99;
        repaint();
    }
});
myBtnE[2].addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        flagE=2;
        flagN=99;
        repaint();
    }
});
} //public FTyusya ()

//public void paint(Graphics g) の開始*****
public void paint(Graphics g) {

    if (myMt.checkID(0)) {
        //駐車場描写
        g.drawLine(67,0+20,67,350+20);g.drawLine(68,0+20,68,350+20);
        g.drawLine(233,0+20,233,350+20);g.drawLine(232,0+20,232,350+20);
    }
}

```

```

g.drawLine(0,130-1+20,67,130-1+20);g.drawLine(0,240-1+20,67,240-1+20);
g.drawLine(233,130-1+20,300,130-1+20);g.drawLine
(233,240-1+20,300,240-1+20);

//駐車場番号表示
/*g.drawString("①",40,40+20);
g.drawString("②",40,150+20);
g.drawString("③",40,260+20);
g.drawString("④",253-3,40+20);
g.drawString("⑤",253-3,150+20);
g.drawString("⑥",253-3,260+20);*/

g.drawString("どかし回数="+kk,110,80+20);
g.drawString("実験回数="+ct,120-3,120+20);

//North パネル
//初期状態または初期化ボタンを押したとき
if (flagN==0) {
g.clearRect(0,0,300,350+20); //全体クリア

//駐車場描写
g.drawLine(67,0+20,67,350+20);g.drawLine(68,0+20,68,350+20);
g.drawLine(233,0+20,233,350+20);g.drawLine(232,0+20,232,350+20);
g.drawLine(0,130-1+20,67,130-1+20);g.drawLine
(0,240-1+20,67,240-1+20);
g.drawLine(233,130-1+20,300,130-1+20);g.drawLine
(233,240-1+20,300,240-1+20);
g.drawString("どかし回数="+kk,110,80+20);
g.drawString("実験回数="+ct,120-3,120+20);

//駐車場番号表示
/*g.drawString("①",40,40+10+20);
g.drawString("②",40,150+10+20);
g.drawString("③",40,260+10+20);
g.drawString("④",253-3,40+10+20);
g.drawString("⑤",253-3,150+10+20);
g.drawString("⑥",253-3,260+10+20);*/

//サイコロ表示
g.drawImage(img[1],135,270+20,this);
}
//どかし回数ボタンを押したとき
else if (flagN==1) {
switch (r) {
case 1:
g.drawImage(img[1],135,270+20,this);
break;
case 2:
g.drawImage(img[2],135,270+20,this);
break;
case 3:
g.drawImage(img[3],135,270+20,this);
break;
case 4:
g.drawImage(img[4],135,270+20,this);
break;
case 5:
g.drawImage(img[5],135,270+20,this);
break;
case 6:
g.drawImage(img[6],135,270+20,this);
break;
}
}
}

```



```

//サイコロ振るボタンを押したとき
else if (flagN==2) {
    g.drawString("Copyright(C) K.Niwa 2002.08",80-10,325+20);

//自動車の表示・消去
if (flag1!=0) {
    g.drawImage(img[0],40-5,60+20,this); //自動車①の表示
}
else if (flag1==0) {
    g.clearRect(40-5,60+20,22,32+20); //自動車①の消去
}
if (flag2!=0) {
    g.drawImage(img[0],40-5,170+20,this); //自動車②の表示
}
else if (flag2==0) {
    g.clearRect(40-5,170+20,22,32+20); //自動車②の消去
}
if (flag3!=0) {
    g.drawImage(img[0],40-5,280+20,this); //自動車③の表示
}
else if (flag3==0) {
    g.clearRect(40-5,280+20,22,32+20); //自動車③の消去
}
if (flag4!=0) {
    g.drawImage(img[0],238+5+3-3,60+20,this); //自動車④の表示
}
else if (flag4==0) {
    g.clearRect(238+5+3-3,60+20,22,32+20); //自動車④の消去
}
if (flag5!=0)
{
    g.drawImage(img[0],238+5+3-3,170+20,this); //自動車⑤の表示
}
else if (flag5==0) {
    g.clearRect(238+5+3-3,170+20,22,32+20); //自動車⑤の消去
}
if (flag6!=0) {
    g.drawImage(img[0],238+5+3-3,280+20,this); //自動車⑥の表示
}
else if (flag6==0) {
    g.clearRect(238+5+3-3,280+20,22,32+20); //自動車⑥の消去
}

//番号の表示
//駐車場1の番号の表示
if (flag1==1) {
    g.drawString("①",40,40+15+20);
}
else if (flag1==2) {
    g.drawString("②",40,40+15+20);
}
else if (flag1==3) {
    g.drawString("③",40,40+15+20);
}
else if (flag1==4) {
    g.drawString("④",40,40+15+20);
}
else if (flag1==5) {
    g.drawString("⑤",40,40+15+20);
}
else if (flag1==6) {
    g.drawString("⑥",40,40+15+20);
}
}

```

```

//駐車場 2 の番号の表示
if (flag2==1) {
    g.drawString ("②",40,150+15+20);
}
else if (flag2==2) {
    g.drawString ("③",40,150+15+20);
}
else if (flag2==3) {
    g.drawString ("④",40,150+15+20);
}
else if (flag2==4) {
    g.drawString ("⑤",40,150+15+20);
}
else if (flag2==5) {
    g.drawString ("⑥",40,150+15+20);
}
else if (flag2==6) {
    g.drawString ("①",40,150+15+20);
}
//駐車場 3 の番号の表示
if (flag3==1) {
    g.drawString ("③",40,260+15+20);
}
else if (flag3==2) {
    g.drawString ("④",40,260+15+20);
}
else if (flag3==3) {
    g.drawString ("⑤",40,260+15+20);
}
else if (flag3==4) {
    g.drawString ("⑥",40,260+15+20);
}
else if (flag3==5) {
    g.drawString ("①",40,260+15+20);
}
else if (flag3==6) {
    g.drawString ("②",40,260+15+20);
}
//駐車場 4 の番号の表
if (flag4==1) {
    g.drawString ("④",250-3,40+15+20);
}
else if (flag4==2) {
    g.drawString ("⑤",250-3,40+15+20);
}
else if (flag4==3) {
    g.drawString ("⑥",250-3,40+15+20);
}
else if (flag4==4) {
    g.drawString ("①",250-3,40+15+20);
}
else if (flag4==5) {
    g.drawString ("②",250-3,40+15+20);
}
else if (flag4==6) {
    g.drawString ("③",250-3,40+15+20);
}
//駐車場 5 の番号の表示
if (flag5==1) {
    g.drawString ("⑤",250-3,150+15+20);
}
else if (flag5==2) {
    g.drawString ("⑥",250-3,150+15+20);
}
}

```

```

else if (flag5==3) {
    g.drawString ("①",250-3,150+15+20);
}
else if (flag5==4) {
    g.drawString ("②",250-3,150+15+20);
}
else if (flag5==5) {
    g.drawString ("③",250-3,150+15+20);
}
else if (flag5==6) {
    g.drawString ("④",250-3,150+15+20);
}
//駐車場 6 の番号の表示
if (flag6==1) {
    g.drawString ("⑥",250-3,260+15+20);
}
else if (flag6==2) {
    g.drawString ("①",250-3,260+15+20);
}
else if (flag6==3) {
    g.drawString ("②",250-3,260+15+20);
}
else if (flag6==4) {
    g.drawString ("③",250-3,260+15+20);
}
else if (flag6==5) {
    g.drawString ("④",250-3,260+15+20);
}
else if (flag6==6) {
    g.drawString ("⑤",250-3,260+15+20);
}

//サイコロパラパラ表示
for (k=1;k<=10;k++) {
    for (count=1;count<=6;count++) {
        g.drawImage (img [count],135,270+20,this);
        for (t=1;t<=500000;t++) {} //時間かせぎ
    }
}

//サイコロの目の判断
r=(int) (6*Math.random ()+1);
switch (r) {
    case 1:
        g.drawImage (img [1],135,270+20,this);
        break;
    case 2:
        g.drawImage (img [2],135,270+20,this);
        break;
    case 3:
        g.drawImage (img [3],135,270+20,this);
        break;
    case 4:
        g.drawImage (img [4],135,270+20,this);
        break;
    case 5:
        g.drawImage (img [5],135,270+20,this);
        break;
    case 6:
        g.drawImage (img [6],135,270+20,this);
        break;
}
} //else if (flagN==2)

```

```

//West パネル
//①ボタンを押したとき
if (flagW==0) {
    flag1=flag1+1;
    if (flag1==7) {
        flag1=0;
    }
    flagW=99;
}
//②ボタンを押したとき
else if (flagW==1) {
    flag2=flag2+1;
    if (flag2==7) {
        flag2=0;
    }
    flagW=99;
}
//③ボタンを押したとき
else if (flagW==2) {
    flag3=flag3+1;
    if (flag3==7) {
        flag3=0;
    }
    flagW=99;
}
}

//East パネル
//④ボタンを押したとき-
if (flagE==0) {
    flag4=flag4+1;
    if (flag4==7) {
        flag4=0;
    }
    flagE=99;
}
//⑤ボタンを押したとき
else if (flagE==1)
{
    flag5=flag5+1;
    if (flag5==7) {
        flag5=0;
    }
    flagE=99;
}
//⑥ボタンを押したとき
else if (flagE==2) {
    flag6=flag6+1;
    if (flag6==7) {
        flag6=0;
    }
    flagE=99;
}
}

//自動車の表示・消去
if (flag1!=0) {
    g.drawImage (img[0],40-5,60+20,this);           //自動車①の表示
}
else if (flag1==0) {
    g.clearRect (40-5,60+20,22,32+20);           //自動車①の消去
}
if (flag2!=0) {
    g.drawImage (img[0],40-5,170+20,this);       //自動車②の表示
}
}

```

```

else if (flag2==0) {
    g.clearRect(40-5,170+20,22,32+20); //自動車②の消去
}
if (flag3!=0) {
    g.drawImage(img[0],40-5,280+20,this); //自動車③の表示
}
else if (flag3==0) {
    g.clearRect(40-5,280+20,22,32+20); //自動車③の消去
}
if (flag4!=0) {
    g.drawImage(img[0],238+5+3-3,60+20,this); //自動車④の表示
}
else if (flag4==0) {
    g.clearRect(238+5+3-3,60+20,22,32+20); //自動車④の消去
}
if (flag5!=0) {
    g.drawImage(img[0],238+5+3-3,170+20,this); //自動車⑤の表示
}
else if (flag5==0) {
    g.clearRect(238+5+3-3,170+20,22,32+20); //自動車⑤の消去
}
if (flag6!=0) {
    g.drawImage(img[0],238+5+3-3,280+20,this); //自動車⑥の表示
}
else if (flag6==0) {
    g.clearRect(238+5+3-3,280+20,22,32+20); //自動車⑥の消去
}

//番号の表示
//駐車場 1 の番号の表示
if (flag1==1) {
    g.drawString("①",40,40+15+20);
}
else if (flag1==2) {
    g.drawString("②",40,40+15+20);
}
else if (flag1==3) {
    g.drawString("③",40,40+15+20);
}
else if (flag1==4) {
    g.drawString("④",40,40+15+20);
}
else if (flag1==5) {
    g.drawString("⑤",40,40+15+20);
}
else if (flag1==6) {
    g.drawString("⑥",40,40+15+20);
}
//駐車場 2 の番号の表示
if (flag2==1) {
    g.drawString("②",40,150+15+20);
}
else if (flag2==2) {
    g.drawString("③",40,150+15+20);
}
else if (flag2==3) {
    g.drawString("④",40,150+15+20);
}
else if (flag2==4) {
    g.drawString("⑤",40,150+15+20);
}
else if (flag2==5) {
    g.drawString("⑥",40,150+15+20);
}
}

```

```

else if (flag2==6) {
    g.drawString ("①",40,150+15+20);
}
//駐車場 3 の番号の表示
if (flag3==1) {
    g.drawString ("③",40,260+15+20);
}
else if (flag3==2) {
    g.drawString ("④",40,260+15+20);
}
else if (flag3==3) {
    g.drawString ("⑤",40,260+15+20);
}
else if (flag3==4) {
    g.drawString ("⑥",40,260+15+20);
}
else if (flag3==5) {
    g.drawString ("①",40,260+15+20);
}
else if (flag3==6) {
    g.drawString ("②",40,260+15+20);
}
//駐車場 4 の番号の表示
if (flag4==1) {
    g.drawString ("④",250-3,40+15+20);
}
else if (flag4==2) {
    g.drawString ("⑤",250-3,40+15+20);
}
else if (flag4==3) {
    g.drawString ("⑥",250-3,40+15+20);
}
else if (flag4==4) {
    g.drawString ("①",250-3,40+15+20);
}
else if (flag4==5) {
    g.drawString ("②",250-3,40+15+20);
}
else if (flag4==6) {
    g.drawString ("③",250-3,40+15+20);
}
//駐車場 5 の番号の表示
if (flag5==1) {
    g.drawString ("⑤",250-3,150+15+20);
}
else if (flag5==2) {
    g.drawString ("⑥",250-3,150+15+20);
}
else if (flag5==3) {
    g.drawString ("①",250-3,150+15+20);
}
else if (flag5==4) {
    g.drawString ("②",250-3,150+15+20);
}
else if (flag5==5) {
    g.drawString ("③",250-3,150+15+20);
}
else if (flag5==6) {
    g.drawString ("④",250-3,150+15+20);
}
//駐車場 6 の番号の表示
if (flag6==1) {
    g.drawString ("⑥",250-3,260+15+20);
}

```

```

else if (flag6==2) {
    g.drawString ("①",250-3,260+15+20);
}
else if (flag6==3) {
    g.drawString ("②",250-3,260+15+20);
}
else if (flag6==4) {
    g.drawString ("③",250-3,260+15+20);
}
else if (flag6==5) {
    g.drawString ("④",250-3,260+15+20);
}
else if (flag6==6) {
    g.drawString ("⑤",250-3,260+15+20);
}

//サイコロの表示
switch (r) {
case 1:
    g.drawImage (img [1],135,270+20,this);
    break;
case 2:
    g.drawImage (img [2],135,270+20,this);
    break;
case 3:
    g.drawImage (img [3],135,270+20,this);
    break;
case 4:
    g.drawImage (img [4],135,270+20,this);
    break;
case 5:
    g.drawImage (img [5],135,270+20,this);
    break;
case 6:
    g.drawImage (img [6],135,270+20,this);
    break;
} //switch (r) {

//作者の表示
g.drawString ("Copyright (C) K.Niwa 2002.08",80-10,325+20);

} //if (myMt.checkID (0))

} //public void paint (Graphics g)

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

} //public class FTyuusya extends Frame

```