```
/*
  A basic extension of the java.applet.Applet class
 */

import java.awt.*;
import java.applet.*;

public class LiftControl extends Applet
{
  public void init()
  {
    // Take out this line if you don't use symantec.itools.net.RelativeURL or sy
mantec.itools.awt.util.StatusScroller
    // symantec.itools.lang.Context.setApplet(this);

    // This code is automatically generated by Visual Cafe when you add
    // components to the visual environment. It instantiates and initializes
    // the components. To modify the code, only use code syntax that matches
    // what Visual Cafe can generate, or Visual Cafe may be unable to back
    // parse your Java file into its visual environment.
    //{{INIT_CONTROLS
    setLayout(null);
    setSize(826,606);
    Upanel = new java.awt.Panel();
    Upanel.setLayout(new GridLayout(6,1,0,0));
    Upanel.setBounds(216,72,70,495);
    add(Upanel);
    U6 = new java.awt.Button();
    U6.setBounds(0,0,70,82);
    U6.setBackground(new Color(12632256));
    Upanel.add(U6);
    U5 = new java.awt.Button();
    U5.setLabel("昇");
    U5.setBounds(0,82,70,82);
    U5.setBackground(new Color(12632256));
    Upanel.add(U5);
    U4 = new java.awt.Button();
    U4.setLabel("昇");
    U4.setBounds(0,164,70,82);
    U4.setBackground(new Color(12632256));
    Upanel.add(U4);
    U3 = new java.awt.Button();
    U3.setLabel("昇");
    U3.setBounds(0,246,70,82);
    U3.setBackground(new Color(12632256));
    Upanel.add(U3);
    U2 = new java.awt.Button();
    U2.setLabel("昇");
    U2.setBounds(0,328,70,82);
    U2.setBackground(new Color(12632256));
    Upanel.add(U2);
    U1 = new java.awt.Button();
    U1.setLabel("昇");
    U1.setBounds(0,410,70,82);
    U1.setBackground(new Color(12632256));
    Upanel.add(U1);
    label7 = new java.awt.Label("昇ボタン",Label.CENTER);
    label7.setBounds(204,12,94,29);
    add(label7);
    Dpanel = new java.awt.Panel();
    Dpanel.setLayout(new GridLayout(6,1,0,0));
    Dpanel.setBounds(312,72,81,496);
    add(Dpanel);
    D6 = new java.awt.Button();
    D6.setLabel("降");
    D6.setBounds(0,0,81,82);
    D6.setBackground(new Color(12632256));
    Dpanel.add(D6);
    D5 = new java.awt.Button();
    D5.setLabel("降");
    D5.setBounds(0,82,81,82);
    D5.setBackground(new Color(12632256));
    Dpanel.add(D5);
    D4 = new java.awt.Button();
    D4.setLabel("降");
    D4.setBounds(0,164,81,82);
    D4.setBackground(new Color(12632256));
    Dpanel.add(D4);
    D3 = new java.awt.Button();
    D3.setLabel("降");
    D3.setBounds(0,246,81,82);
    D3.setBackground(new Color(12632256));
    Dpanel.add(D3);
```

```
D2 = new java.awt.Button();
D2.setLabel("降");
D2.setBounds(0,328,81,82);
D2.setBackground(new Color(12632256));
Dpanel.add(D2);
D1 = new java.awt.Button();
D1.setBounds(0,410,81,82);
D1.setBackground(new Color(12632256));
Dpanel.add(D1);
label1 = new java.awt.Label("降ボタン",Label.CENTER);
label1.setBounds(312,12,84,36);
add(label1);
Epanel = new java.awt.Panel();
Epanel.setLayout(new GridLayout(6,1,0,0));
Epanel.setBounds(456,72,55,178);
add(Epanel);
E6 = new java.awt.Button();
E6.setLabel("６階");
E6.setBounds(0,0,55,29);
E6.setBackground(new Color(12632256));
Epanel.add(E6);
E5 = new java.awt.Button();
E5.setLabel("５階");
E5.setBounds(0,29,55,29);
E5.setBackground(new Color(12632256));
Epanel.add(E5);
E4 = new java.awt.Button();
E4.setLabel("４階");
E4.setBounds(0,58,55,29);
E4.setBackground(new Color(12632256));
Epanel.add(E4);
E3 = new java.awt.Button();
E3.setLabel("３階");
E3.setBounds(0,87,55,29);
E3.setBackground(new Color(12632256));
Epanel.add(E3);
E2 = new java.awt.Button();
E2.setLabel("２階");
E2.setBounds(0,116,55,29);
E2.setBackground(new Color(12632256));
Epanel.add(E2);
E1 = new java.awt.Button();
E1.setLabel("１階");
E1.setBounds(0,145,55,29);
E1.setBackground(new Color(12632256));
Epanel.add(E1);
label2 = new java.awt.Label("エレベータ内ボタン");
label2.setBounds(444,12,108,32);
add(label2);
label3 = new java.awt.Label("エレベータ",Label.CENTER);
label3.setBounds(84,24,85,26);
add(label3);
Finish = new java.awt.Button();
Finish.setLabel("終了");
Finish.setBounds(456,480,77,41); // Finish.setBounds(636,516,77,41);
Finish.setBackground(new Color(12632256));
// add(Finish);
Start = new java.awt.Button();
Start.setLabel("開始");
Start.setBounds(456,420,73,41); // Start.setBounds(636,456,73,41);
Start.setBackground(new Color(12632256));
add(Start);
Liftpanel = new java.awt.Panel();
Liftpanel.setLayout(new GridLayout(6,1,0,0));
Liftpanel.setBounds(72,70,92,490);
Liftpanel.setBackground(new Color(12632256));
add(Liftpanel);
sixthF = new java.awt.Label("６階");
sixthF.setBounds(0,0,92,81);
Liftpanel.add(sixthF);
fifthF = new java.awt.Label("５階");
fifthF.setBounds(0,81,92,81);
Liftpanel.add(fifthF);
fourthF = new java.awt.Label("４階");
fourthF.setBounds(0,162,92,81);
Liftpanel.add(fourthF);
thirdF = new java.awt.Label("３階");
thirdF.setBounds(0,243,92,81);
Liftpanel.add(thirdF);
secondF = new java.awt.Label("２階");
secondF.setBounds(0,324,92,81);
Liftpanel.add(secondF);
firstF = new java.awt.Label("１階");
firstF.setBounds(0,405,92,81);
Liftpanel.add(firstF);
//}}
```

```
    //{{REGISTER_LISTENERS
    SymMouse aSymMouse = new SymMouse();
    Start.addMouseListener(aSymMouse);
    Finish.addMouseListener(aSymMouse);
    E1.addMouseListener(aSymMouse);
    E2.addMouseListener(aSymMouse);
    E3.addMouseListener(aSymMouse);
    E4.addMouseListener(aSymMouse);
    E5.addMouseListener(aSymMouse);
    E6.addMouseListener(aSymMouse);
    D2.addMouseListener(aSymMouse);
    D3.addMouseListener(aSymMouse);
    D4.addMouseListener(aSymMouse);
    D5.addMouseListener(aSymMouse);
    D6.addMouseListener(aSymMouse);
    U1.addMouseListener(aSymMouse);
    U2.addMouseListener(aSymMouse);
    U3.addMouseListener(aSymMouse);
    U4.addMouseListener(aSymMouse);
    U5.addMouseListener(aSymMouse);
    //}}
  }

  //{{DECLARE_CONTROLS
  java.awt.Panel Upanel;
  java.awt.Button U6;
  java.awt.Button U5;
  java.awt.Button U4;
  java.awt.Button U3;
  java.awt.Button U2;
  java.awt.Button U1;
  java.awt.Label label7;
  java.awt.Panel Dpanel;
  java.awt.Button D6;
  java.awt.Button D5;
  java.awt.Button D4;
  java.awt.Button D3;
  java.awt.Button D2;
  java.awt.Button D1;
  java.awt.Label label1;
  java.awt.Panel Epanel;
  java.awt.Button E6;
  java.awt.Button E5;
  java.awt.Button E4;
  java.awt.Button E3;
  java.awt.Button E2;
  java.awt.Button E1;
  java.awt.Label label2;
  java.awt.Label label3;
  java.awt.Button Finish;
  java.awt.Button Start;
  java.awt.Panel Liftpanel;
  java.awt.Label sixthF;
  java.awt.Label fifthF;
  java.awt.Label fourthF;
  java.awt.Label thirdF;
  java.awt.Label secondF;
  java.awt.Label firstF;
  //}}

  class SymMouse extends java.awt.event.MouseAdapter
  {
    public void mousePressed(java.awt.event.MouseEvent event)
    {
      Object object = event.getSource();
      if (object == Start)
        Start_MousePressed(event);
      else if (object == Finish)
        Finish_MousePressed(event);
      else if (object == E1)
        E1_MousePressed(event);
      else if (object == E2)
        E2_MousePressed(event);
      else if (object == E3)
        E3_MousePressed(event);
      else if (object == E4)
        E4_MousePressed(event);
      else if (object == E5)
        E5_MousePressed(event);
      else if (object == E6)
        E6_MousePressed(event);
      else if (object == D2)
        D2_MousePressed(event);
      else if (object == D3)
        D3_MousePressed(event);
      else if (object == D4)
        D4_MousePressed(event);
      else if (object == D5)
```

```
      D5_MousePressed(event);
    else if (object == D6)
      D6_MousePressed(event);
    else if (object == U1)
      U1_MousePressed(event);
    else if (object == U2)
      U2_MousePressed(event);
    else if (object == U3)
      U3_MousePressed(event);
    else if (object == U4)
      U4_MousePressed(event);
    else if (object == U5)
      U5_MousePressed(event);
  }
}

void Start_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.
      WaitThread wth = new WaitThread(1000); //WaitThread
      wth.start();

  //{{CONNECTION
  // Request the focus
  Start.requestFocus();
  //}}
}

void Finish_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.

  //{{CONNECTION
  // Hide the Applet
    try {
        System.exit(0);
    } catch (SecurityException e) {}
  //System.exit(0); //setVisible(false);
  //}}
}

void E1_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.
  int i = 1;
  //E1.setBackground(new Color(16762880));
  liftrequestCommon(i);
  //{{CONNECTION
  // Show the Applet
  setVisible(true);
  //}}
}

void E2_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.
  int i = 2;
  //E2.setBackground(new Color(16762880));
  liftrequestCommon(i);
  //{{CONNECTION
  // Show the Applet
  setVisible(true);
  //}}
}

void E3_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.
  int i = 3;
  //E3.setBackground(new Color(16762880));
  liftrequestCommon(i);
  //{{CONNECTION
  // Show the Applet
  setVisible(true);
  //}}
}

void E4_MousePressed(java.awt.event.MouseEvent event)
{
  // to do: code goes here.
  int i = 4;
  //E4.setBackground(new Color(16762880));
  liftrequestCommon(i);
  //{{CONNECTION
  // Show the Applet
  setVisible(true);
```

```
    //}}
  }

  void E5_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 5;
    //E5.setBackground(new Color(16762880));
    liftrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void E6_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 6;
    //E6.setBackground(new Color(16762880));
    liftrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void liftrequestCommon(int f){
    ED.liftreq.LiftRequestAdd(f);

    ED.liftbtn.ItsButtonLightOn(f);
  }

  void D2_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 2;
    //D2.setBackground(new Color(16762880));
    downrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void D3_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 3;
    //D3.setBackground(new Color(16762880));
    downrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void D4_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 4;
    //D4.setBackground(new Color(16762880));
    downrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void D5_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 5;
    //D5.setBackground(new Color(16762880));
    downrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void D6_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 6;
    //D6.setBackground(new Color(16762880));
```

```
    downrequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void downrequestCommon(int f){
    ED.downreq.DownRequestAdd(f);
    ED.downbtn.ItsButtonLightOn(f);
  }

  void U1_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 1;
    //U1.setBackground(new Color(16762880));
    uprequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void U2_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 2;
    //U2.setBackground(new Color(16762880));
    uprequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void U3_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 3;
    //U3.setBackground(new Color(16762880));
    uprequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void U4_MousePressed(java.awt.event.MouseEvent event)
  {
    // to do: code goes here.
    int i = 4;
    //U4.setBackground(new Color(16762880));
    uprequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void U5_MousePressed(java.awt.event.MouseEvent event)
  {
    {
    // to do: code goes here.
    int i = 5;
    //U5.setBackground(new Color(16762880));
    uprequestCommon(i);
    //{{CONNECTION
    // Show the Applet
    setVisible(true);
    //}}
  }

  void uprequestCommon(int f){
    ED.upreq.UpRequestAdd(f);
    ED.upbtn.ItsButtonLightOn(f);
  }
}

//基本データとオブジェクト
class ED
{
    public static String signal;            //信号
    public static String act;               //リフトの状態
    public static String cid;               //COMKIND
                        //"leaveUp"|"arriveUp"|"leaveDown"|"arriveDown"|"noOpera
tion"
```

```
    public static int groundFloorNo = 1;
    public static int topFloorNo = 6;
    public static int liftid = 1;
    public static int cf;                    //current Floor
    //オブジェクトの生成
    public static Lift lift = new Lift();
  public static ButtonPanel liftbtn = new ButtonPanel();  //リフトボタンパネルの
インスタンス
  public static ButtonPanel upbtn = new ButtonPanel();    //上昇用フロアボタンパ
ネルのインスタンス
  public static ButtonPanel downbtn = new ButtonPanel();  //下降用フロアボタンパ
ネルのインスタンス
    public static LiftReqSet liftreq = new LiftReqSet();    //リフトボタンからの
要求
    public static UpRequest upreq = new UpRequest();        //フロアからの上昇要
求
    public static DownRequest downreq = new DownRequest();  //フロアからの下降要
求
}

//フロアからの上昇要求
class UpRequest
{
  //初期値はすべてfalseにセットされる => Init処理不要
  static boolean upreq[] = new boolean[ED.topFloorNo - ED.groundFloorNo + 1];

  void UpRequestAdd(int i){
          upreq[i - 1] = true;
  }

  void UpRequestRemove(int i){
          upreq[i - 1] = false;
  }

  boolean UpRequestFromCurrentFlr(int i){
          if (upreq[i - 1] == true) return true;
          else return false;
  }

  boolean UpRequestFromAboveFlr(int i){
          int j = i;
          while (j <= ED.topFloorNo - ED.groundFloorNo - 1){
              if (upreq[j] == true) return true;
              j++;
          }
          return false;
  }

  boolean UpRequestFromBelowFlr(int i){
          int j = i - 2;                              //current floor No - 2 最
上階からの要求なし
          while (j >= ED.groundFloorNo - 1){
              if (upreq[j] == true) return true;
              j--;
          }
          return false;
  }

}

//フロアからの下降要求
class DownRequest
{
  //初期値はすべてfalseにセットされる => Init処理不要
  static boolean downreq[] = new boolean[ED.topFloorNo - ED.groundFloorNo + 1];

  void DownRequestAdd(int i){
          downreq[i - 1] = true;
  }

  void DownRequestRemove(int i){
          downreq[i - 1] = false;
  }

  boolean DownRequestFromCurrentFlr(int i){
          if (downreq[i - 1] == true) return true;
          else return false;
  }

  boolean DownRequestFromAboveFlr(int i){
          int j = i;
          while (j <= ED.topFloorNo - ED.groundFloorNo){
              if (downreq[j] == true) return true;
              j++;
          }
          return false;
```

```
  }

  boolean DownRequestFromBelowFlr(int i){
          int j = i - 2;                  //current floor No - 1;最下位階からの
要求なし
          while (j >= ED.groundFloorNo - 1){
              if (downreq[j] == true) return true;
              j--;
          }
          return false;
  }
}

//リフト行き先要求
class LiftReqSet
{
  static boolean lftreq[] = new boolean[ED.topFloorNo - ED.groundFloorNo + 1];

  static {
      for (int i = ED.groundFloorNo - 1; i < ED.topFloorNo - ED.groundFloorNo; i
++)
          {lftreq[i] = false;}
  }

  void LiftRequestAdd(int i){
          lftreq[i - 1] = true;
  }

  void LiftRequestRemove(int i){
          lftreq[i - 1] = false;
  }

  boolean LiftRequestFromCurrentFlr(int i){
          if (lftreq[i - 1] == true) return true;
          else return false;
  }

  boolean LiftRequestFromAboveFlr(int i){
          int j = i;                      //current floor No + 1
          while (j <= ED.topFloorNo - ED.groundFloorNo){
              if (lftreq[j] == true) return true;
              j++;
          }
          return false;
  }

  boolean LiftRequestFromBelowFlr(int i){
          int j = i - 2;                  //current floor No - 1
          while (j >= ED.groundFloorNo - 1){
              if (lftreq[j] == true) return true;
              j--;
          }
          return false;
  }
}

//ボタン
class button
{
      boolean isLighted;
      public void ButtonLightOn(){
          isLighted = true;
      }

      public void ButtonLightOff(){
          isLighted = false;
      }

      public boolean ButtonIsDark(){
          if (isLighted == false) return true;
          else return false;
      }

      public boolean ButtonIsBright(){
          if (isLighted == true) return true;
          else return false;
      }
}

//ボタン配列（上昇用フロアボタン／下降用フロアボタン／リフトボタン）
class ButtonPanel
{
      button btnpanel[] = new button[ED.topFloorNo - ED.groundFloorNo + 1];

      ButtonPanel()    //初期化＝配列要素のオブジェクト化
      {
```

```
            for (int j = ED.groundFloorNo - 1; j < ED.topFloorNo; j++){
                btnpanel[j] = new button();
            }
        }

        void ItsButtonLightOn(int f){
            btnpanel[f - 1].ButtonLightOn();
        }

        void ItsButtonLightOff(int f){
            btnpanel[f - 1].ButtonLightOff();
        }

        boolean ItsButtonIsDark(int f){
            return btnpanel[f - 1].ButtonIsDark();
        }

        boolean ItsButtonIsBright(int f){
            return btnpanel[f - 1].ButtonIsBright();
        }
}

//リスト
class Lift
{
        //初期化
        Lift(){
          ED.signal = "canLeaveSig";
          ED.act = "waiting";
          ED.cf = ED.groundFloorNo;
        }

        //上昇指令
        void LiftCommandLeavingUp(){
            if ((ED.act.equals("waiting"))||
                (ED.act.equals("waitingUp"))||
                (ED.act.equals("waitingDown"))&&
                (ED.cf < ED.topFloorNo)){
                ED.cid = "leaveUp";
                ED.act = "movingUp";
                //扉を閉める
                switch (ED.cf) {
                    case 1: System.out.println("1F close");break;
                    case 2: System.out.println("2F close");break;
                    case 3: System.out.println("3F close");break;
                    case 4: System.out.println("4F close");break;
                    case 5: System.out.println("5F close");break;
                    case 6: System.out.println("6F close");break;
                }
                ED.cf++;
                //信号は到着可能を示す
                ED.signal = "arrivingSig";
            }
        }

        //上昇継続
        void LiftContinueLeavingUp(){
            if (ED.act.equals("movingUp") && ED.cf < ED.topFloorNo){
                ED.cid = "noOperation";
                switch (ED.cf) {
                    case 1: System.out.println("1F go through");break;
                    case 2: System.out.println("2F go through");break;
                    case 3: System.out.println("3F go through");break;
                    case 4: System.out.println("4F go through");break;
                    case 5: System.out.println("5F go through");break;
                    case 6: System.out.println("6F go through");break;
                }
                ED.cf++;
            }
        }

        //上昇到着指令
        void LiftCommandArrivingUp(){
            if (ED.act.equals("movingUp")){
                ED.cid = "arriveUp";
                ED.act = "waitingUp";
                //信号は離床可能状態を示す
                ED.signal = "canLeaveSig";
                //? 扉を開く
                switch (ED.cf) {
                    case 1: System.out.println("1F open");break;
                    case 2: System.out.println("2F open");break;
                    case 3: System.out.println("3F open");break;
                    case 4: System.out.println("4F open");break;
                    case 5: System.out.println("5F open");break;
                    case 6: System.out.println("6F open");break;
```

```
            }
        }
    }

    //下降指令
    void LiftCommandLeavingDown(){
        if ((ED.act.equals("waiting") || ED.act.equals("waitingDown") || ED.ac
t.equals("waitingUp")) &&
            ED.cf > ED.groundFloorNo){
            ED.cid = "leaveDown";
            ED.act = "movingDown";
            //扉を閉じる
            try {
            switch (ED.cf) {
                case 1: System.out.println("1F close");break;
                case 2: System.out.println("2F close");break;
                case 3: System.out.println("3F close");break;
                case 4: System.out.println("4F close");break;
                case 5: System.out.println("5F close");break;
                case 6: System.out.println("6F close");break;
            }
            }catch(NullPointerException e){System.out.println("error: "+e);}
            ED.cf--;
            //信号は到着可能を示す
            ED.signal = "arrivingSig";
        }
    }

    //下降継続
    void LiftContinueLeavingDown(){
        if (ED.act.equals("movingDown") && ED.cf > ED.groundFloorNo){
            ED.cid = "noOperation";
            switch (ED.cf) {
                case 1: System.out.println("1F go through");break;
                case 2: System.out.println("2F go through");break;
                case 3: System.out.println("3F go through");break;
                case 4: System.out.println("4F go through");break;
                case 5: System.out.println("5F go through");break;
                case 6: System.out.println("6F go through");break;
            }
            ED.cf--;
        }
    }

    //下降到着指令
    void LiftCommandArrivingDown(){
        if (ED.act.equals("movingDown")){
            ED.cid = "arriveDown";
            ED.act = "waitingDown";
            //信号は離床可能状態を示す
            ED.signal = "canLeaveSig";
        }
            //扉を開く
            switch (ED.cf) {
                case 1: System.out.println("1F open");break;
                case 2: System.out.println("2F open");break;
                case 3: System.out.println("3F open");break;
                case 4: System.out.println("4F open");break;
                case 5: System.out.println("5F open");break;
                case 6: System.out.println("6F open");break;
            }
    }

    //待機状態になる
    void LiftWait(){
        if (ED.act.equals("waitingUp") || ED.act.equals("waitingDown")){
            ED.cid = "noOperation";
            ED.act = "waiting";
        }
    }
}

//本体
class WaitThread extends Thread
{
    int delay;
    WaitThread(int d){
        delay = d;
    }

    public void run(){
        while (true){
            try {Thread.sleep(delay);
            }catch (InterruptedException e){}
            LiftRunInSys(ED.signal);
        }
```

```
}

void LiftRunInSys(String signal){
    if (signal.equals("canLeaveSig")){
        if (ED.act.equals("waiting")){          //LiftIsWaiting
            LiftReactForCanLeaveSigWhenWaiting();
        }
        else if (ED.act.equals("waitingUp")){
            LiftReactForCanLeaveSigWhenWaitingUp();
        }
        else if (ED.act.equals("waitingDown")){
            LiftReactForCanLeaveSigWhenWaitingDown();
        }
    }
    else if (signal.equals("arrivingSig")){
        if (ED.act.equals("movingDown")){
            LiftReactForArrivingSigWhenMovingDown();
        }
        else if (ED.act.equals("movingUp")){
            LiftReactForArrivingSigWhenMovingUp();
        }
    }
}

void LiftReactForCanLeaveSigWhenWaiting(){
    if (ED.act.equals("waiting")){
        if (ToGoUp(ED.cf) && ToGoDown(ED.cf)){
            LiftLeaveUp();
        }
        else if (ToGoUp(ED.cf) && !ToGoDown(ED.cf)){
            ED.downreq.DownRequestRemove(ED.cf);     //挿入
            ED.downbtn.ItsButtonLightOff(ED.cf);
            LiftLeaveUp();
        }
        else if (!ToGoUp(ED.cf) && ToGoDown(ED.cf)){
            ED.upreq.UpRequestRemove(ED.cf);              //挿入
            ED.upbtn.ItsButtonLightOff(ED.cf);                  //挿入
            LiftLeaveDown();
        }
        else if (!ToGoUp(ED.cf) && !ToGoDown(ED.cf)){
            LiftWaitInEnv();
        }
    }
}

void LiftReactForArrivingSigWhenMovingUp(){
    if (ED.act.equals("movingUp")){
        if (ToArriveUp(ED.cf)){
            LiftArriveUp();
        }
        else {
            LiftContinueUpInEnv();
        }
    }
}

void LiftReactForCanLeaveSigWhenWaitingUp(){
    if (ED.act.equals("waitingUp")){
        if (ToGoUp(ED.cf)){
            LiftLeaveUp();
        }
        else if (!ToGoUp(ED.cf) && ToGoDown(ED.cf)){
            ED.upreq.UpRequestRemove(ED.cf);
            ED.upbtn.ItsButtonLightOff(ED.cf);
            LiftLeaveDown();
        }
        else if (!ToGoUp(ED.cf) && !ToGoDown(ED.cf)){
            LiftWaitInEnv();
        }
    }
}

void LiftReactForArrivingSigWhenMovingDown(){
    if (ED.act.equals("movingDown")){
        if (ToArriveDown(ED.cf)){
            LiftArriveDown();
        }
        else {
            LiftContinueDownInEnv();
        }
    }
}

void LiftReactForCanLeaveSigWhenWaitingDown(){
    if (ED.act.equals("waitingDown")){
        if (ToGoDown(ED.cf)){
```

```
                LiftLeaveDown();
            }
            else if (!ToGoDown(ED.cf) && ToGoUp(ED.cf)){
                ED.downreq.DownRequestRemove(ED.cf);
                ED.downbtn.ItsButtonLightOff(ED.cf);
                LiftLeaveUp();
            }
            else if (!ToGoDown(ED.cf) && !ToGoUp(ED.cf)){
                LiftWaitInEnv();
            }
        }
    }

    boolean ToGoUp(int f){
        return ED.liftreq.LiftRequestFromAboveFlr(f) || ED.upreq.UpRequestFromAb
oveFlr(f)
                || ED.downreq.DownRequestFromAboveFlr(f);
    }

    boolean ToArriveUp(int f){
        return (ED.liftreq.LiftRequestFromCurrentFlr(f) || ED.upreq.UpRequestFro
mCurrentFlr(f)
                ) || !ToGoUp(f);
    }

    boolean ToGoDown(int f){
        return ED.liftreq.LiftRequestFromBelowFlr(f) || ED.upreq.UpRequestFromBe
lowFlr(f)
                || ED.downreq.DownRequestFromBelowFlr(f);
    }

    boolean ToArriveDown(int f){
    return (ED.liftreq.LiftRequestFromCurrentFlr(f)
    || ED.downreq.DownRequestFromCurrentFlr(f)) || !ToGoDown(f);
    }
    void LiftWaitInEnv(){
        ED.lift.LiftWait();
    }

    void LiftLeaveUp(){
        ED.upreq.UpRequestRemove(ED.cf);
        ED.upbtn.ItsButtonLightOff(ED.cf);
        //
        ED.lift.LiftCommandLeavingUp();             //Remove/LightOffの後
    }

    void LiftLeaveDown(){
        ED.downreq.DownRequestRemove(ED.cf);
        ED.downbtn.ItsButtonLightOff(ED.cf);          //ItsDownButtonLightOff()
        //
        ED.lift.LiftCommandLeavingDown();           //後ろに
    }

    void LiftArriveUp(){
        ED.lift.LiftCommandArrivingUp();
        ED.liftreq.LiftRequestRemove(ED.cf);
        //
        ED.liftbtn.ItsButtonLightOff(ED.cf);          //ItsLiftButtonLightOff()
    }

    void LiftArriveDown(){
        ED.lift.LiftCommandArrivingDown();
        ED.liftreq.LiftRequestRemove(ED.cf);
        //
        ED.liftbtn.ItsButtonLightOff(ED.cf);          //ItsLiftButtonLightOff()
    }

    void LiftContinueUpInEnv(){
        ED.lift.LiftContinueLeavingUp();
    }

    void LiftContinueDownInEnv(){
        ED.lift.LiftContinueLeavingDown();
    }
}
```