

PROSTA ANIMACJA - SWING

W oparciu o program

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.*;
import javax.swing.*;
public class japplet_jpanel_jbutton extends JApplet implements ActionListener {
    int applet_width=500;
    int applet_height=400;
    Wykres wykr = new Wykres();
    public void init(){
        setSize(applet_width,applet_height);
        JLabel etykieta = new JLabel("Mój japplet");
        setLayout(new FlowLayout());
        add(wykr);
        etykieta.setBorder(BorderFactory.createLineBorder(Color.orange));
        JButton przycisk = new JButton("START");
        przycisk.addActionListener(this);
        przycisk.setBorder(BorderFactory.createLineBorder(Color.green));
        add(etykieta);
        add(przycisk);
    }
    public void actionPerformed(ActionEvent e){
        wykr.repaint();
    }
}
class Wykres extends JPanel{
    int wielkosc=200;
    public Wykres(){setBorder(BorderFactory.createLineBorder(Color.blue));
    }

    public Dimension getPreferredSize()
    {return new Dimension(wielkosc,wielkosc);}

    public void paintComponent(Graphics graf){
        super.paintComponent(graf);
        graf.drawOval((int)(wielkosc*Math.random()),
(int)(wielkosc*Math.random()), 10, 10);
    }
}
```

Utwórz program, w którym wciśnięcie przycisku uruchamia animację (cykliczne wywoływanie metody paintComponent). Wewnątrz metody paintComponent zastosuj uśpienie

```
try {
    Thread.sleep(100); // sleep for 10 msec
} catch (InterruptedException t){}
```

Przycisk START zmień na START/STOP

BŁĄDZENIE PRZYPADKOWE

Napisz program który wizualizuje błądzenie przypadkowe w dwóch wymiarach. W metodzie `paintComponent` modyfikowane są losowo współrzędne cząstki (walker) oraz wartości w tablicy `siec[][]`. Następnie rysowany jest kwadrat (1x1) w miejscu gdzie tablica `siec` przyjmuje niezerową wartość. Po uruchomieniu cząstka jest widoczna w położeniu początkowym. Wciśnięcie `START` uruchamia animację.

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.*;
import javax.swing.*;
public class animation_random_walk extends JApplet implements ActionListener {
    int applet_width=500;
    int applet_height=400;
    Wykres2 wykres = new Wykres2();
    public void init(){
        setSize(applet_width,applet_height);
        JLabel etykieta = new JLabel("Mój applet");
        setLayout(new FlowLayout());
        wykres.setBorder(BorderFactory.createLineBorder(Color.blue));
        add(wykres);
        etykieta.setBorder(BorderFactory.createLineBorder(Color.orange));
        JButton przycisk = new JButton("START");
        przycisk.addActionListener(this);
        przycisk.setBorder(BorderFactory.createLineBorder(Color.green));
        add(etykieta);
        add(przycisk);
    }
    public void actionPerformed(ActionEvent e){
        wykres.repaint();
        wykres.rysuj=true;
    }
}
```

```
class Wykres2 extends JPanel{
    boolean rysuj;
    int wielkosc=300;
    int [][] siec = new int [wielkosc][wielkosc];
    Particle walker;
    public Wykres2(){setSize(wielkosc,wielkosc);
        rysuj=false;
        for (int i=0;i<wielkosc;i++){
            for (int j=0;j<wielkosc;j++){
                siec[i][j]=0;
            }
        }
        walker = new Particle(wielkosc/2,wielkosc/2);
        siec[wielkosc/2][wielkosc/2]=1;
    }
    public Dimension getPreferredSize()
    {return new Dimension(wielkosc,wielkosc);
    }

    public void paintComponent(Graphics graf){
```

```

        super.paintComponent(graf);

        for (int i=0;i<wielkosc;i++){
        for (int j=0;j<wielkosc;j++){
            if(siec[i][j]==1){graf.fillRect(i,j,1,1);}
        }
        }
        double losowa=Math.random();
            if(losowa<0.25) {x=x+1;if (x==wielkosc){x=0;}}
            else if(losowa<0.5) {x=x-1;if (x==-1){x=wielkosc-1;}}
            else if(losowa<0.75) {y=y+1;if (y==wielkosc){y=0;}}
            else {y=y-1;if (y==-1){y=wielkosc-1;}}

        } //end of paintComponent
    }

    class Particle {
        private int x,y;
        public Particle (int xx, int yy)
        {x=xx;y=yy;}
        int get_x(){return x;}
        int get_y(){return y;}
        void set_x(int xx){x=xx;}
        void set_y(int yy){y=yy;}
    }

```

Gaz sieciowy

Wizualizacja 3 cząstek gazu sieciowego. Periodyczne warunki brzegowe. Uzupełnij definicję metody step(int ktory). Metoda ta dla cząstki o indeksie ktory próbuje losowo zmienić jedną ze współrzędnych (o plus minus 1).

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class j3particles extends JApplet implements ActionListener {
    int applet_width=400;
    int applet_height=400;
    Wykres1 wykres = new Wykres1();
    JButton przycisk;
    public void init(){
        setSize(applet_width,applet_height);
        przycisk = new JButton("START");
        przycisk.addActionListener(this);
        add(wykr, BorderLayout.CENTER);
        add(przycisk, BorderLayout.SOUTH);

    }
    public void actionPerformed(ActionEvent e){
        wykres.rysowac=!wykres.rysowac;
        if (wykres.rysowac) {przycisk.setText("STOP");}
    else{przycisk.setText("START");}
        wykres.repaint();
    }
}

```

```

class Wykres1 extends JPanel{
    boolean rysowac;
    int wielkosc=300;
    int [][] siec = new int [wielkosc][wielkosc];
    int lczastek=3;
    Particle [] walker = new Particle [lczastek];
    public Wykres1(){rysowac=false;
        for (int i=0;i<wielkosc;i++){
            for (int j=0;j<wielkosc;j++){
                siec[i][j]=0;
            }
        }
        walker[0]=new Particle(wielkosc/2,wielkosc/2);
        walker[1]=new Particle(wielkosc/4,wielkosc/4);
        walker[2]=new Particle(wielkosc/2,wielkosc/4);

        siec[wielkosc/2][wielkosc/2]=1;
        siec[wielkosc/4][wielkosc/4]=1;
        siec[wielkosc/2][wielkosc/4]=1;
    }
    public Dimension getPreferredSize()
    {return new Dimension(wielkosc,wielkosc);
    }

    void step (int ktory){

    // TODO

    }
    public void paintComponent(Graphics graf){
        super.paintComponent(graf);
        graf.drawLine(0, 0, 0, wielkosc);
        graf.drawLine(0, 0, wielkosc,0);
        graf.drawLine(0, wielkosc, wielkosc, wielkosc);
        graf.drawLine(wielkosc, 0,wielkosc, wielkosc);
        for (int i=0;i<wielkosc;i++){
            for (int j=0;j<wielkosc;j++){
                if(siec[i][j]==1){graf.drawRect(i,j,1,1);}
            }
        }
        if(rysowac){
            for (int czastka=0;czastka<lczastek;czastka++){
                int indeks=(int)(lczastek*Math.random());
                step(indeks);
            }
            try {
                Thread.sleep(10); // sleep for 10 msec
            } catch (InterruptedException t){}
            repaint();
        }
    }
}

```