

Ewolucja gazu sieciowego jednogatunkowego z anihilacją cząstek ($A+A \rightarrow 0$). Uzupełnij metodę `step()` w której następuje przesunięcie cząstek z ewentualną anihilacją.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
public class anihilacja_aa extends JApplet implements ActionListener {
    int applet_width=550;
    int applet_height=550;
    Wykresaa wyk = new Wykresaa();
    public void init(){
        setSize(applet_width, applet_height);
        JButton przycisk = new JButton("START");
        przycisk.addActionListener(this);
        add(wyk, BorderLayout.CENTER);
        add(przycisk, BorderLayout.SOUTH);
        add(wyk.etykpanel, BorderLayout.NORTH);
    }
    public void actionPerformed(ActionEvent e){
        wyk.czystart=true;
        wyk.repaint();
    }
}

class Wykresaa extends JPanel{
    boolean czystart;
    int wielkosc=500;
    int [][] siec = new int [wielkosc][wielkosc];
    int krok=0;
    int ile_czastek=0;
    Label etykpanel = new Label("anihilacja: A + A -> 0");
    public Wykresaa(){czystart=false;
        for (int i=0; i<wielkosc; i++){
            for (int j=0; j<wielkosc; j++){
                siec[i][j]=1;
            }
        }
    }
    public Dimension getPreferredSize()
    {return new Dimension(wielkosc, wielkosc);
    }
    void step (){
        int x=0, y=0;
```

```

        for (int wezel=0;wezel<wielkosc*wielkosc;wezel++){
int ix=(int)(wielkosc*Math.random());
int iy=(int)(wielkosc*Math.random());
x=ix;
y=iy;
double losowa=Math.random();
if(losowa<0.25) {x=ix+1;if (x==wielkosc){x=0;}}
else if(losowa<0.5) {x=ix-1;if (x==-1){x=wielkosc-1;}}
else if(losowa<0.75) {y=iy+1;if (y==wielkosc){y=0;}}
else {y=iy-1;if (y==-1){y=wielkosc-1;}}
}
}
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);
ile_czastek=0;
for (int i=0;i<wielkosc;i++){
for (int j=0;j<wielkosc;j++){
if(siec[i][j]==1){graf.setColor(Color.BLUE);graf.drawRect(i,j,1,1);ile_
czastek++;}
}
if((czystart)&&(krok<10000)){
krok++;
step();
try {
Thread.sleep(50); // sleep for 10 msec
} catch (InterruptedException t){}
repaint();
setForeground(Color.black);
etykpanel.setText(""+ile_czastek);
System.out.println("krok="+krok+" #czastek="+ile_czastek);
}
}
}
}

```

Zapis ilości cząstek (ile_czastek) w funkcji czasu (krok) do pliku.

```
try {
```

```

        FileWriter outFile = new
FileWriter("anihilaa.txt",true);
        PrintWriter out1 = new PrintWriter(outFile);
        out1.println(""+krok+" "+ile_czastek);
        out1.close();
    } catch (IOException e){
        e.printStackTrace();
    }
}

```

Ewolucja gazu sieciowego dwugatunkowego z anihilacją cząstek ($A+B \rightarrow 0$). Zapis ilości cząstek (ile_czastek) w funkcji czasu (krok) do pliku:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
public class anihilacja_ab extends JApplet implements
ActionListener {
    int applet_width=550;
    int applet_height=550;
    Wykres2ab wykr = new Wykres2ab();
    public void init(){
        setSize(applet_width,applet_height);
        JButton przycisk = new JButton("START");
        przycisk.addActionListener(this);
        add(wykr, BorderLayout.CENTER);
        add(przycisk, BorderLayout.SOUTH);
        add(wykr.etykpanel, BorderLayout.NORTH);

    }
    public void actionPerformed(ActionEvent e){
        wykr.czystart=true;
        wykr.repaint();
    }
}

class Wykres2ab extends JPanel{
    boolean czystart;
    int wielkosc=300;
}

```

```

int [][] siec = new int [wielkosc][wielkosc];
int krok=0;
int ile_czastek=0;
Label etykpanel = new Label("ilość cząstek");
public Wykres2ab(){czystart=false;
    for (int i=0;i<wielkosc;i++){
        for (int j=0;j<wielkosc;j++){

siec[i][j]=(int)(2*Math.random()+1;
        }
    }
}
public Dimension getPreferredSize()
{return new Dimension(wielkosc,wielkosc);
}
void step (){
    int x=0,y=0;
    for (int
wezel=0;wezel<wielkosc*wielkosc;wezel++){
    int ix=(int)(wielkosc*Math.random());
    int iy=(int)(wielkosc*Math.random());
    x=ix;
    y=iy;
    int siecxy=siec[ix][iy];
    if(siecxy>0){
    double losowa=Math.random();
    if(losowa<0.25) {x=ix+1;if (x==wielkosc){x=0;}}
    else if(losowa<0.5) {x=ix-1;if (x==
1){x=wielkosc-1;}}
    else if(losowa<0.75) {y=iy+1;if
(y==wielkosc){y=0;}}
    else {y=iy-1;if (y==-1){y=wielkosc-1;}}
    if(siecxy!=siec[x][y]){ //anihilacja lub
przesunięcie
        siec[ix][iy]=0;
        siec[x][y]=(siecxy+siec[x][y])%3;
    }
}
}
}

```

```

        try {
            FileWriter outFile = new
FileWriter("anihil_ab.txt",true);
            PrintWriter out1 = new
PrintWriter(outFile);
            out1.println(""+krok+" "+ile_czastek);
            out1.close();
        } catch (IOException e){
            e.printStackTrace();
        }
    }
    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);
        ile_czastek=0;
        for (int i=0;i<wielkosc;i++){
            for (int j=0;j<wielkosc;j++){

if(siec[i][j]==1){graf.setColor(Color.BLUE);graf.drawRect(i,j
,1,1);ile_czastek++;}

if(siec[i][j]==2){graf.setColor(Color.RED);graf.drawRect(i,j,
1,1);ile_czastek++;}
            }
        }
        if((czystart)&&(krok<10000)){
            krok++;
            step();
            try {
                Thread.sleep(50); // sleep for
10 msec
            } catch (InterruptedException t){}
            repaint();
            setForeground(Color.black);
            etykpanel.setText(""+ile_czastek);

```

```

        System.out.println("krok="+krok+"
#czastek="+ile_czastek);
    }
}
}

```

Wprowadź uśrednianie przebiegów czasowych.

Wyznacz wykładnik określający potęgowy zanik ilości cząstek w czasie. Powtórz obliczenia dla gazu dwugatunkowego. Dla dużych czasów t gęstość wydaje się dość szybko zanikać do zera. W jaki sposób charakterystyczny czas takiego zaniku zależy od rozmiaru sieci?

Zbadaj ewolucję gazu dwugatunkowego gdzie stan początkowy jest typu szachownicy.

Gaz dwugatunkowy z dryfem. Zmodyfikuj metodę step tak aby cząstki rodzaju 1 miały dryf skierowany w prawo a 2 w lewo.

```

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
public class dwarodzaje extends JApplet implements ActionListener {
    int applet_width=500;
    int applet_height=500;
    Wykres222 wykr = new Wykres222();
    JButton przycisk = new JButton("START");
    public void init(){
        setSize(applet_width,applet_height);
        przycisk.addActionListener(this);
        add(wykr, BorderLayout.CENTER);
        add(przycisk, BorderLayout.SOUTH);
        add(wykr.bias_text, BorderLayout.NORTH);
        wykr.bias_text.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==przycisk){
            wykr.czystart=true;
            wykr.repaint();
        }
        else{
            wykr.bias=Double.parseDouble(wykr.bias_text.getText());
        }
    }
}

```

```

} // koniec jparticles

class Wykres222 extends JPanel{
    double bias=0.2;
    JTextField bias_text = new JTextField(""+bias);
    boolean czystart;
    int wielkosc=300;
    int [][] siec = new int [wielkosc][wielkosc];
    double gestosc=0.3;
    public Wykres222(){czystart=false;
    setBorder(BorderFactory.createLineBorder(Color.blue));
        for (int i=0;i<wielkosc;i++){
            for (int j=0;j<wielkosc;j++){
                if (Math.random()<gestosc){
                    siec[i][j]=(int)(2.0*Math.random()+1);
                }
                else{
                    siec[i][j]=0;}
            }
        }
    }

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

    void step (int ktory_x, int ktory_y){
        int old_x=ktory_x;
        int old_y=ktory_y;
        int siecxy=siec[ktory_x][ktory_y];
        if (siecxy!=0){
            double losowa=Math.random();
            if(losowa<0.25) {ktory_x=ktory_x+1;if
(ktory_x==wielkosc){ktory_x=0;}}
            else if(losowa<0.5) {ktory_x=ktory_x-1;if (ktory_x==
-1){ktory_x=wielkosc-1;}}
            else if(losowa<0.75) {ktory_y=ktory_y+1;if
(ktory_y==wielkosc){ktory_y=0;}}
            else {ktory_y=ktory_y-1;if (ktory_y==-1){ktory_y=wielkosc-1;}}
            if(siec[ktory_x][ktory_y]==0){
                siec[old_x][old_y]=0;
                siec[ktory_x][ktory_y]=siecxy;
            }
        }
    }

    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.setColor(Color.blue);graf.drawRect(i,j,1,1);}

                if(siec[i][j]==2){graf.setColor(Color.red);graf.drawRect(i,j,1,1);}
            }
        }

        if(czystart){

```

```
        for (int czastka=0;czastka<wielkosc*wielkosc;czastka++){
            int indeks1=(int)(wielkosc*Math.random());
            int indeks2=(int)(wielkosc*Math.random());
            step(indeks1,indeks2);
        }
    try {
        Thread.sleep(20); // sleep for 10 msec
    } catch (InterruptedException t){}
    repaint();
}
} // koniec Wykres2
```