

Ewolucja gazu sieciowego jednogatunkowego z anihilacją cząstek ($A+A \rightarrow 0$). 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_aa extends JApplet implements ActionListener {
    int applet_width=550;
    int applet_height=550;
    Wykres2abz wyk = new Wykres2abz();
    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){
        wyk.czystart=true;
        wyk.repaint();
    }
}

class Wykres2abz 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 Wykres2abz(){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;
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;}}
siec[ix][iy]=0;
siec[x][y]=(siecxy+siec[x][y])%2;
}
}
try {
    FileWriter outFile = new
FileWriter("anihil.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_c
zastek++;}
}
}
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);
    }
}
}

```

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.*;
public class anihilacja_aa extends JApplet implements ActionListener {
    int applet_width=800;
    int applet_height=800;
    Wykres2 wykres = new Wykres2();
    public void init(){
        setSize(applet_width,applet_height);
        JButton przycisk = new JButton("START");
        przycisk.addActionListener(this);
        add(wykres, BorderLayout.CENTER);
        add(przycisk, BorderLayout.SOUTH);
    }
    public void actionPerformed(ActionEvent e){
        wykres.rysowac=true;
        wykres.repaint();
    }
}

class Wykres2 extends JPanel{
    boolean rysowac;
    int wielkosc=600;
    int [][] siec = new int [wielkosc][wielkosc];
    public Wykres2(){rysowac=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 ix,int iy){
    int x=ix;
    int y=iy;
    int siecxy=siec[ix][iy];
    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;}}
    siec[ix][iy]=0;
    siec[x][y]=(siecxy+siec[x][y])%2;
}
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);
    if(rysowac){
        for (int wezel=0;wezel<wielkosc*wielkosc;wezel++){
            step((int)(wielkosc*Math.random()),(int)(wielkosc*Math.random()));
        }
        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);}
            }
        }
        try {
            Thread.sleep(20); // sleep for 10 msec
        } catch (InterruptedException t){}
        repaint();
    }
}
}

```

Dodaj etykietę, na której uwidaczniana będzie aktualna liczba cząstek.