

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
import java.awt.*;
import java.util.*;
import java.awt.event.*;
import javax.swing.*;

//KLASA GŁÓWNA:

public class Prod_kons extends JFrame
{
    public Prod_kons()
    {
        super("Model producent - konsument");
        this.setSize(800,600);
        this.setLocation(50,50);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container zawartosc=this.getContentPane();
        zawartosc.add(new PanelGlowny());
        this.setVisible(true);
    }

    public static void main(String[] args)
    {
        new Prod_kons();
    }
}

//KLASA PANELU:

class PanelGlowny extends JPanel implements ActionListener, ItemListener
{
    Font naglowek1, naglowek2, przyc, tekst;
    Color tlo, nieb, czerw, granat, bialy;

    //magazyn towarów:
    LinkedList magazyn;

    //wielkość magazynu:
    int max=4;

    //czy koniec symulacji:
    boolean koniec=false;

    //opóźnienia
    int opoznProd=2, opoznKons=10;

    JTextField infoProd, infoKons, stan;
    ButtonGroup przelProd, przelKons;
    JRadioButton prod2, prod5, prod10, kons2, kons5, kons10;
    JComboBox pojemnosc;
    TextArea raport;
    JButton start, stop;

    //wątki:
    Konsument kons;
    Producent prod;

    public PanelGlowny()
    {
        magazyn=new LinkedList();
```

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
this.setSize(800,600);  
  
naglowek1=new Font("Times", Font.BOLD, 32);  
naglowek2=new Font("Times", Font.BOLD | Font.ITALIC, 22);  
przyc=new Font("Times", Font.BOLD, 20);  
tekst=new Font("Times", Font.BOLD, 20);  
tlo=new Color(0, 213, 226);  
nieb=new Color(6, 11, 220);  
czerw=new Color(200, 2, 2);  
granat=new Color(15, 19, 173);  
bialy=new Color(255, 255, 255);  
  
this.setBackground(tlo);  
this.setLayout(null);  
  
JLabel etykProd1=new JLabel("Producent", JLabel.CENTER);  
etykProd1.setLocation(50,20);  
etykProd1.setSize(200,45);  
etykProd1.setFont(naglowek1);  
etykProd1.setForeground(czerw);  
this.add(etykProd1);  
  
JLabel etykProd2=new JLabel("Wątek 1", JLabel.CENTER);  
etykProd2.setLocation(50,70);  
etykProd2.setSize(200,30);  
etykProd2.setFont(naglowek2);  
etykProd2.setForeground(nieb);  
this.add(etykProd2);  
  
infoProd=new JTextField(35);  
infoProd.setLocation(25,120);  
infoProd.setSize(250,45);  
infoProd.setFont(tekst);  
infoProd.setEditable(false);  
this.add(infoProd);  
  
JLabel etykProd3=new JLabel("Opóźnienie [sek]:", JLabel.CENTER);  
etykProd3.setLocation(50,180);  
etykProd3.setSize(200,30);  
etykProd3.setFont(naglowek2);  
etykProd3.setForeground(nieb);  
this.add(etykProd3);  
  
przelProd=new ButtonGroup();  
  
prod2=new JRadioButton("2", true);  
prod2.setLocation(50,220);  
prod2.setSize(50,30);  
prod2.setFont(przyc);  
prod2.addItemListener(this);  
przelProd.add(prod2);  
this.add(prod2);  
  
prod5=new JRadioButton("5", false);  
prod5.setLocation(110,220);  
prod5.setSize(50,30);  
prod5.setFont(przyc);  
prod5.addItemListener(this);
```

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
przelProd.add(prod5);  
this.add(prod5);
```

```
prod10=new JRadioButton("10", false);  
prod10.setLocation(170,220);  
prod10.setSize(50,30);  
prod10.setFont(przyc);  
prod10.addItemListener(this);  
przelProd.add(prod10);  
this.add(prod10);
```

```
JLabel etykMagaz1=new JLabel("Magazyn", JLabel.CENTER);  
etykMagaz1.setLocation(300,20);  
etykMagaz1.setSize(200,45);  
etykMagaz1.setFont(naglowek1);  
etykMagaz1.setForeground(czerw);  
this.add(etykMagaz1);
```

```
JLabel etykMagaz2=new JLabel("Zasób współdzielony", JLabel.CENTER);  
etykMagaz2.setLocation(275,70);  
etykMagaz2.setSize(250,30);  
etykMagaz2.setFont(naglowek2);  
etykMagaz2.setForeground(nieb);  
this.add(etykMagaz2);
```

```
stan=new JTextField(10);  
stan.setLocation(300,120);  
stan.setSize(200,45);  
stan.setFont(tekst);  
stan.setEditable(false);  
this.add(stan);
```

```
JLabel etykMagaz3=new JLabel("Pojemność magazynu:", JLabel. CENTER);  
etykMagaz3.setLocation(275,180);  
etykMagaz3.setSize(250,30);  
etykMagaz3.setFont(naglowek2);  
etykMagaz3.setForeground(nieb);  
this.add(etykMagaz3);
```

```
String[] opcjePojemn={"4", "8", "12"};  
pojemnosc=new JComboBox(opcjePojemn);  
pojemnosc.setLocation(300,210);  
pojemnosc.setSize(200,30);  
pojemnosc.setFont(przyc);  
pojemnosc.addItemListener(this);  
this.add(pojemnosc);
```

```
JLabel etykKons1=new JLabel("Konsument", JLabel.CENTER);  
etykKons1.setLocation(550,20);  
etykKons1.setSize(200,45);  
etykKons1.setFont(naglowek1);  
etykKons1.setForeground(czerw);  
this.add(etykKons1);
```

```
JLabel etykKons2=new JLabel("Wątek2", JLabel.CENTER);  
etykKons2.setLocation(550,70);  
etykKons2.setSize(200,30);  
etykKons2.setFont(naglowek2);  
etykKons2.setForeground(nieb);
```

```
this.add(etykKons2);

infoKons=new JTextField(35);
infoKons.setLocation(525,120);
infoKons.setSize(250,45);
infoKons.setFont(tekst);
infoKons.setEditable(false);
this.add(infoKons);

JLabel etykKons3=new JLabel("Opóźnienie [sek]:", JLabel.CENTER);
etykKons3.setLocation(550,180);
etykKons3.setSize(200,30);
etykKons3.setFont(naglowek2);
etykKons3.setForeground(nieb);
this.add(etykKons3);

przelKons=new ButtonGroup();

kons2=new JRadioButton("2", false);
kons2.setLocation(550,220);
kons2.setSize(50,30);
kons2.setFont(przyc);
kons2.addItemListener(this);
przelKons.add(kons2);
this.add(kons2);

kons5=new JRadioButton("5", false);
kons5.setLocation(610,220);
kons5.setSize(50,30);
kons5.setFont(przyc);
kons5.addItemListener(this);
przelKons.add(kons5);
this.add(kons5);

kons10=new JRadioButton("10", true);
kons10.setLocation(670,220);
kons10.setSize(50,30);
kons10.setFont(przyc);
kons10.addItemListener(this);
przelKons.add(kons10);
this.add(kons10);

start=new JButton("start");
start.setLocation(650,320);
start.setSize(80,40);
start.addActionListener(this);
start.setActionCommand("start");
start.setFont(przyc);
start.setBackground(granat);
start.setForeground(bialy);
this.add(start);

stop=new JButton("stop");
stop.setLocation(650,380);
stop.setSize(80,40);
stop.addActionListener(this);
stop.setActionCommand("stop");
stop.setFont(przyc);
stop.setBackground(granat);
```

```
stop.setForeground(bialy);
this.add(stop);

JLabel etykRaport1=new JLabel("Raport symulacji:");
etykRaport1.setLocation(50,270);
etykRaport1.setSize(300,40);
etykRaport1.setFont(naglowek1);
etykRaport1.setForeground(czerw);
this.add(etykRaport1);

raport=new TextArea(5,30);
raport.setLocation(50,320);
raport.setSize(550,230);
raport.setFont(tekst);
raport.setEditable(false);
this.add(raport);

this.setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
    String ktory=e.getActionCommand();

    if(ktory.equals("start"))
    {
        start.setEnabled(false);
        stop.setEnabled(true);
        synchronized(magazyn)
        {
            magazyn.clear();
            raport.setText("");
            koniec=false;
            magazyn.notifyAll();
        }
        kons=new Konsument();
        prod=new Producent();
        kons.start();
        prod.start();
        pojemnosc.setEnabled(false);
    }

    if(ktory.equals("stop"))
    {
        stop.setEnabled(false);
        start.setEnabled(true);
        synchronized(magazyn)
        {
            magazyn.clear();
            koniec=true;
            magazyn.notifyAll();
        }
        pojemnosc.setEnabled(true);
    }
}

public void itemStateChanged(ItemEvent i)
{
    int poj;
```

```
poj=pojemnosc.getSelectedIndex();

if(prod2.isSelected())
{
    opoznProd=2;
}
if(prod5.isSelected())
{
    opoznProd=5;
}
if(prod10.isSelected())
{
    opoznProd=10;
}
if(kons2.isSelected())
{
    opoznKons=2;
}
if(kons5.isSelected())
{
    opoznKons=5;
}
if(kons10.isSelected())
{
    opoznKons=10;
}
if(poj==0)
{
    max=4;
}
if(poj==1)
{
    max=8;
}
if(poj==2)
{
    max=12;
}
}
```

```
//Klasa producenta:
class Producent extends Thread
{
    Object towar;

    public void run()
    {
        while(true)
        {
            synchronized(magazyn)
            {
                //Jeśli kolejka pełna:
                while(magazyn.size()==max)
                {
                    try
                    {
                        if(koniec) break;
                    }
                }
            }
        }
    }
}
```

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
        infoProd.setText("Producent czeka");
        raport.append("\nProducent czeka.");
        magazyn.wait();
    }
    catch(InterruptedException ex)
    {
    }
}

//Jeśli koniec symulacji zakończ wątek:
if(koniec) break;

//Dodaj nowy towar do kolejki:
magazyn.addFirst(towar);
infoProd.setText("Producent dodał towar");
raport.append("\nProducent dodał towar. Ilość towarów:
"+magazyn.size()+".");
stan.setText("Stan aktualny: "+String.valueOf(magazyn.size()));
//Poinformuj o tym inne wątki:
magazyn.notifyAll();
}

try
{
    Thread.sleep(1000);
}
catch(InterruptedException ex)
{
}

try
{
    infoProd.setText("Producent uśpiony");
    Thread.sleep((opoznProd-1)*1000);
}
catch(InterruptedException ex)
{
}
}
}

//Klasa konsumenta:
class Konsument extends Thread
{
    Object towar;

    public void run()
    {
        while(true)
        {
            synchronized(magazyn)
            {
                //Jeśli kolejka pusta:
                while(magazyn.size()==0)
                {
                    try
                    {
```

Krzysztof Pancerz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
        if(koniec) break;
        infoKons.setText("Konsument czeka");
        raport.append("\nKonsument czeka.");
        magazyn.wait();
    }
    catch(InterruptedException ex)
    {
    }
}

//Jeśli koniec symulacji zakończ wątek:
if(koniec) break;

//Weź towar z kolejki:
towar=magazyn.removeLast();
infoKons.setText("Konsument wziął towar");
raport.append("\nKonsument wziął towar. Ilość towarów:
"+magazyn.size()+".");
stan.setText("Stan aktualny: "+String.valueOf(magazyn.size()));
//Poinformuj o tym inne wątki:
magazyn.notifyAll();
}

try
{
    Thread.sleep(1000);
}
catch(InterruptedException ex)
{
}

try
{
    infoKons.setText("Konsument uśpiony");
    Thread.sleep((opoznKons-1)*1000);
}
catch(InterruptedException ex)
{
}

}

}

}
```