

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

//KLASA PANELU:

```
class PanelGlowny extends JPanel implements ActionListener, ItemListener
{
```

```
    // ...
```

//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;
```

```
                        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)
```

```
    {}
```

```
}
```

Krzysztof Pancierz
Programowanie współbieżne i rozproszone
Symulacja modelu producent - konsument

```
    }  
}  
  
//Klasa konsumenta:  
class Konsument extends Thread  
{  
    Object towar;  
  
    public void run()  
    {  
        while(true)  
        {  
            synchronized(magazyn)  
            {  
                //Jeśli kolejka pusta:  
                while(magazyn.size()==0)  
                {  
                    try  
                    {  
                        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)  
            {}  
        }  
    }  
}
```