

# Потоки, анімація, багатовіконність на Java

The background features a light gray gradient with numerous thin, vertical, slightly wavy lines in a teal color. A solid teal horizontal band is positioned at the bottom of the image, containing the title text.

# Робота з потоками та анімація

```
import java.awt.*;
import java.awt.event.*;
public class NervousText extends Frame implements Runnable
{
    char separated[];
    String s = null;
    Thread killme = null;
    int x_coord = 0, y_coord = 0;
    public NervousText()
    {
        super("Nervo");
        s = "Java is the best"; // задаємо довільний текст
        separated = new char [s.length()];
        s.getChars(0, s.length(), separated, 0); // робимо з нього масив символів
        setSize(250, 100); // задаємо розмір вікна, зверніть увагу, щоб помістився текст
        setFont(new Font("TimesRoman", Font.BOLD, 36));
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {System.exit(0);} // реакція на закриття
        });
        show();
    }
}
```

```
public static void main(String[] args)
{
    NervousText nerv=new NervousText();
    nerv.start();
}
public void start()
{
    if (killme == null)
    {
        killme = new Thread(this);
        killme.start();
    }
}
public void stop()
{
    killme = null;
}
public void run()
{
    while (killme != null)
    {
        try {Thread.sleep(100);} // ВИВІД ПОТОКУ ІЗ ЗАТРИМКОЮ
catch (InterruptedException e){}
        repaint();
    }
    killme = null;
}
public void paint(Graphics g) // функція виводу кожної букви у випадкову коорд.
{
    for (int i=0; i<s.length(); i++)
    {
        x_coord = (int) (Math.random()*10 + 15*i);
        y_coord = (int) (Math.random()*10 + 66);
        g.drawChars(separated, i, 1, x_coord, y_coord); //ВИВЕД. СИМВОЛУ НА ЕКРАН
    }
}
}
```

# Коментарі до коду

- Типова програма, що використовує метод `run()` для роботи з потоками інтерфейсу `Runnable`, виглядає так:

```
• public class MyClass extends Frame implements Runnable
• {
•     private Thread myThread = null; // об'ява потоку
•     ...
•     public void start()
•     {
•         if (myThread == null)
•         {
•             myThread = new Thread(this);
•             myThread.start();
•         }
•     }
•     public void run()
•     {
•         ...
•     }
• }
```

- Що ж містить метод `run()`? Якщо потік використовується для виконання будь-якої періодичної роботи, цей метод містить цикл виду:

```
• while (myThread != null)
```

- При цьому можна вважати, що код додатка та код метода `run()` працюють одночасно як різні потоки. Коли цикл закінчується та метод `run()` повертає управління, потік завершує роботу нормальним чином.
- А що знаходиться в середині циклу `while`? Як правило, він містить виклик методу `repaint()` для перерисовки, а також виклик методу `sleep()` класу `Thread`, який робить затримку в мілісекундах.

# Динамічна зміна iPanels В iFrame

<https://www.youtube.com/watch?v=nm5ULDfbKD0>

# Створення багатовіконної програми

<https://www.youtube.com/watch?v=bKPGEqJHWaE>