



Programowanie współbieżne

Laboratorium-3

Symulacja argonu

- Potencjał Lennarda-Jonesa

$$U(r) = \epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right]$$

$$\epsilon/k_B = 119.8K, \quad \sigma = 3.822\text{\AA}$$

Krok czasowy: 10^{-14} s, w jednostkach $(m\sigma^2/\epsilon)^{1/2}$

odpowiada to ok. 0,004.



Symulacja: dynamika molekularna

- Przykład symulacji gazów. Patrz np. strona www Amita Kumara:

http://www.personal.psu.edu/auk183/MolDynamics/Molecular_Dynamics_Simulations.html

- Skopiować źródła programu. Uruchomić program w środowisku DrJava
- Wykonać obliczenia dla danych z poprzedniej strony.



Tutorial

- W celu dokładniejszego zapoznania się z wątkami Java najlepiej jest sięgnąć do oryginalnego źródła informacji na ten temat:

<http://download.oracle.com/javase/tutorial/essential/concurrency/index.html>



Wątki (Thread) – przykład

```
public class Watek {
    public static void main( String argum [] ) {
        new PokazWatek("Foo").start( );
        new PokazWatek("Bar").start( );
    }
    static class PokazWatek extends Thread {
        String komunikat;
        PokazWatek( String komunikat ) {
            this.komunikat = komunikat;
        }
        public void run( ) {
            while ( true )
                System.out.println( komunikat );
        }
    }
}
```



Wątki

```
//plik: Apletaktualizacji.java
public class ApletAktualizacji extends java.applet.Applet
    implements Runnable {
    private Thread watekAktualizacji;
    int interwalAktualizacji = 1000;
    public void run( ) {
        while ( watekAktualizacji != null ) {
            try {
                Thread.sleep( interwalAktualizacji );
            }
            catch (InterruptedException e ) {
                return;
            }
            repaint( );
        }
    }
    public void start( ) {
        if ( watekAktualizacji == null ) {
            watekAktualizacji = new Thread(this);
            watekAktualizacji.start( );
        }
    }
    public void stop( ) {
        if ( watekAktualizacji != null ) {
            Thread r = watekAktualizacji;
            watekAktualizacji = null; // oznacz do wyjścia
            r.interrupt( ); // obudź, jeśli śpi
        }
    }
}
```



Wątek „Zegar”

```
//plik: Zegar.java
public class Zegar extends ApletAktualizacji {
    public void paint( java.awt.Graphics g ) {
        g.drawString( new java.util.Date().toString( ), 10, 25 );
    }
}
```

Zegar jest rozszerzeniem apletu `ApletAktualizacji` z poprzedniej strony. Korzysta z interfejsu `Runnable`. (przykład z podręcznika Niemeyera i Knudsen: *Java, wprowadzenie*. Wyd. O'Reilly.)



Problemy domowe

- Problem biesiadujących filozofów (patrz np. Wikipedia: dining philosophers)
- Generowanie prędkości cząstek maxwellowskich (metoda dystrybuanty, metoda Neumanna). Zadana temperatura. Zadana energia układu.
- Wątki (class `Thread`) w Java – prosty przykład (B. Eckel. Thinking in Java, Wyd. 3. Rozdz. 13)
- `interface Runnable` (Java). Metody `start()` oraz `run()`



Problemy cd.

- Analiza programu `ThreadsTest3.java`
- Test programu `ThreadsTest3.java` (). Poszukiwanie liczb pierwszych w zadanym przedziale.
- Uruchamianie programu dla 1, ... , 4 wątków. Analiza czasów wykonania.
- Problem synchronizacji dostępu do zasobów: dyrektywa `synchronize`; metoda `notify()`

