



PW

Lab 1
A. Baran

Java – język programowania

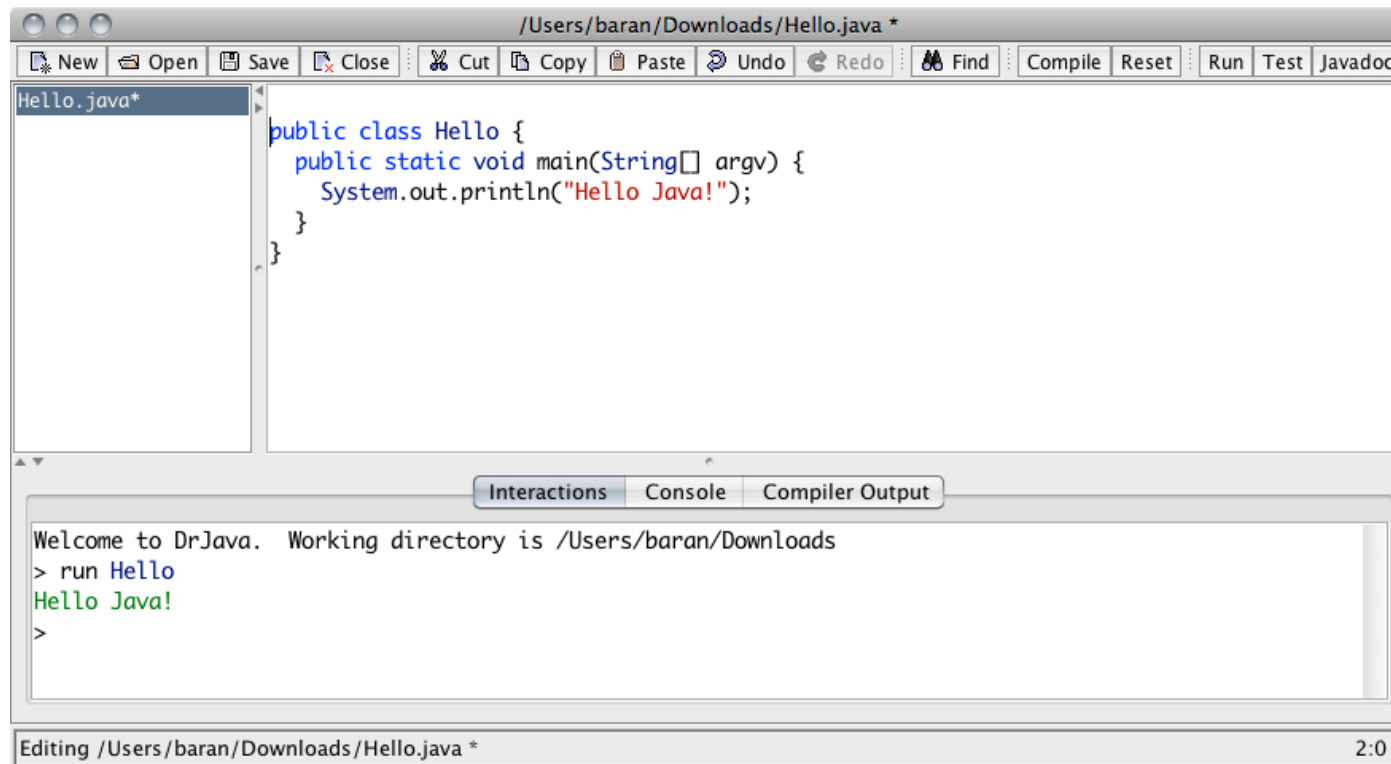
- Java – Sun (Uniwersalny + grafika + ...)
- Środowiska programistyczne:
 - Eclipse <http://www.eclipse.org/>
 - Netbeans <http://netbeans.org/>
 - DrJava <http://www.drjava.org/>
- Hello.java – pierwszy program tekstowy/graficzny
- Uruchamianie w oknie terminala:
 - Javac Hello.java
 - Java Hello

Hello.java

```
public class Hello {  
    public static void main(String[] argv) {  
        System.out.println("Hello Java!");  
    }  
}
```

Zapisać program do pliku Hello.java. Uruchomić.

DrJava



The screenshot shows the DrJava IDE interface. The title bar indicates the file path: `/Users/baran/Downloads/Hello.java *`. The menu bar includes `New`, `Open`, `Save`, `Close`, `Cut`, `Copy`, `Paste`, `Undo`, `Redo`, `Find`, `Compile`, `Reset`, `Run`, `Test`, and `Javadoc`. The main editor window displays the following Java code:

```
public class Hello {  
    public static void main(String[] argv) {  
        System.out.println("Hello Java!");  
    }  
}
```

Below the editor, there are three tabs: `Interactions`, `Console`, and `Compiler Output`. The `Console` tab is active, showing the following output:

```
Welcome to DrJava. Working directory is /Users/baran/Downloads  
> run Hello  
Hello Java!  
>
```

The status bar at the bottom indicates `Editing /Users/baran/Downloads/Hello.java *` and the time `2:0`.



Proste obliczenia

Rozwiązać numerycznie równanie ruchu cząstki o masie m w polu siły elastycznej $F = -kx$, gdzie $k = \text{const}$ z warunkami $x(0) = 0$, $v(0) = 1$ ($m = k = 1$). Porównać wynik metody Eulera z dokładnym rozwiązaniem.

Z prawa Newtona: $F = m a = m dv/dt$, $v = dx/dt =$ prędkość cząstki.

Dyskretyzacja: $x \rightarrow x_1, x_2, \dots, x_n$
 $v_i = (x_{(i+1)} - x_i) / (t_{(i+1)} - t_i) = (x_{(i+1)} - x_i) / T.$

Tutaj t jest czasem, $T = t_{(i+1)} - t_i$ dla $i = 1, 2, \dots, n-1$. Podobnie, przyspieszenie w chwili t_i (w położeniu x_i):

$$a_i = (v_{(i+1)} - v_i) / (t_{(i+1)} - t_i) = (v_{(i+1)} - v_i) / T.$$

Stąd $x_{(i+1)} = x_i + T \cdot v_i$; $v_{(i+1)} = v_i + T/m \cdot f_i$

cd1

```
// An example of studying the motion of a particle in
// one dimension under an elastic force.

import java.lang.*;

public class Motion {
    static final int n = 100000, j = 500;
    public static void main(String argv[]) {
        double x[] = new double[n+1];
        double v[] = new double[n+1];
        // Assign time step and initial position and velocity
        double dt = 2*Math.PI/n;
        x[0] = 0;
        v[0] = 1;
    }
}
```

cd2

```
// Calculate other position and velocity recursively
for (int i=0; i<n; ++i) {
    x[i+1] = x[i]+v[i]*dt;
    v[i+1] = v[i]-x[i]*dt;
}
// Output the result in every j time steps
double t = 0;
double jdt = j*dt;
for (int i=0; i<=n; i+=j) {
    System.out.println(t + " " + x[i] + " " + v[i]);
    t += jdt;
}
}
```

WitajJava.java

```
import javax.swing.*;

public class WitajJava
{
    public static void main( String[] argum ) {
        JFrame ramka = new JFrame( "Witaj Java" );
        ramka.getContentPane().add( new KomponentWitaj() );
        ramka.setSize( 300, 300 );
        ramka.setVisible( true );
    }
}

class KomponentWitaj extends JComponent {
    public void paintComponent( java.awt.Graphics g ) {
        g.drawString( "Witaj Java!", 125, 95 );
    }
}
```


Inna realizacja...

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

public class WitajJava2
{
    public static void main( String[] argum ) {
        JFrame ramka = new JFrame( "WitajJava2" );
        ramka.getContentPane().add(
            new KomponentWitaj2("Witaj Java!") );
        ramka.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        ramka.setSize( 300, 300 );
        ramka.setVisible( true );
    }
}
```

...cd

```
class KomponentWitaj2 extends JComponent
    implements MouseMotionListener {
    String komunikat;
    int komunikatX = 125, komunikatY = 95; // Wspolrzedne komunikatu
    public KomponentWitaj2( String komunikat1 ) {
        komunikat = komunikat1;
        addMouseMotionListener(this);
    }
    public void paintComponent( Graphics g ) {
        g.drawString(komunikat, komunikatX, komunikatY);
    }
    public void mouseDragged(MouseEvent e) {
        // Zapamietaj wspolrzedne myszy i narysuj komunikat.
        komunikatX = e.getX();
        komunikatY = e.getY();
        repaint();
    }
    public void mouseMoved(MouseEvent e) {
    }
}
```