

TECHNOLOGIE WEB I JAVA

Prowadzący: dr hab. inż. **Jan Prokop**, prof. PRz, e-mail: jprokop@prz.edu.pl,
Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki

LABORATORIUM

ĆWICZENIE nr 8

Temat: Podstawy JAVA SE

Java - wybrane zasoby sieci WWW

<https://www.oracle.com/java/>
<https://www.java.com/pl/>
<http://www.javaworld.com/>
<http://www.java.pl/>

Strona przedmiotu

<http://java.prz.edu.pl/>

Java SE Development Kit – instalacja, konfiguracja

<http://www.oracle.com/technetwork/java/javase/downloads/>

Instalacja tylko środowiska Java

- jre-x-windows-i586.exe

Instalacja pakietu JDK

- jdk-x-windows-i586.exe

Instalacja pakietu JDK z NetBeans IDE

- jdk-x-nb-x-win.exe

Instalacja dokumentacji

- jdk-x-doc.zip

Java Control Panel - Panel sterowania ⇒ Programy ⇒ Java

Narzędzia - C:\Program Files\Java\jdk1.x.0\bin ↵

>set path=%path%;path_catalog_jdk_bin ↵

Sprawdzenie działania i opcji narzędzi:

>javac ↵

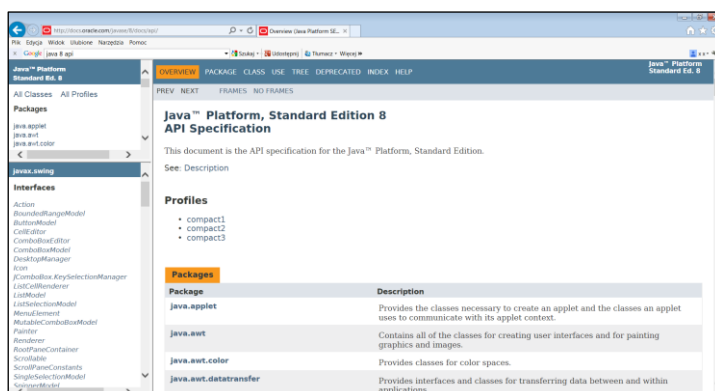
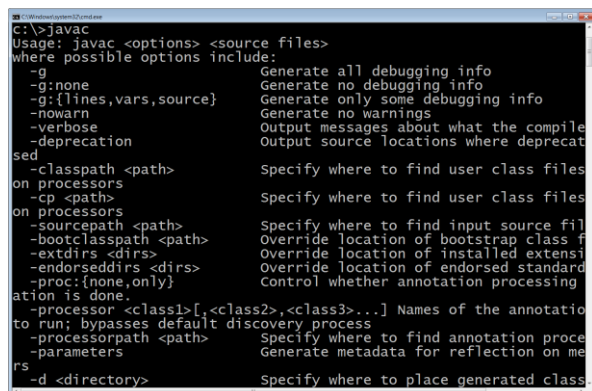
>java ↵ (>javaw ↵)

>java -version ↵

Przegląd dokumentacji API:

<https://docs.oracle.com/en/java/javase/13/docs/api/index.html>

<http://docs.oracle.com/javase/8/docs/api/>



1. Aplikacje konsolowe (CLI)

1.1. Plik 1Program.java

```
/* Pierwszy program
public class PierwszyProgram {
    public static void main(String[] args) {
        System.out.println("Mój pierwszy program w języku Java !");
    }
}
```

Skompilować (usunąć błędy !!!) i uruchomić program z poziomu konsoli DOS

```
>javac PierwszyProgram.java ↵
```

```
>java PierwszyProgram ↵
```

Sprawdzić działanie opcji kompilatora: >javac -verbose PierwszyProgram.java ↵

1.2. Plik DrugiProgram.java – przekazywanie parametrów do aplikacji – usunąć błąd wykonania !

```
public class DrugiProgram {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("No arguments");
            System.exit(0);
        }
        for (int i = 0; i <= args.length; i++) {
            System.out.println("args[" + i + "]: " + args[i]);
        }
    }
}
```

```
>java DrugiProgram Jeden Dwa Trzy ↵
```

Zadanie: Zmodyfikować aplikację tak aby wypisywała na konsolę wynik dzielenia dwóch liczb podanych jako argumenty. Obsłużyć możliwe błędy, w szczególności dzielenie przez zero ! Podpowiedź: `double d = Double.parseDouble("10.2"); try { } catch(ArithmeticException e) { }`

1.3. Aplikacja Quiz.java – czytanie z pliku questions.txt – program konsolowy interaktywny

Quiz.java	questions.txt
<pre>import java.io.*; import java.util.*; public class Quiz { public static void main(String[] args) { int sum = 0; try { Scanner file = new Scanner(new File("questions.txt")); Scanner user = new Scanner(System.in); while (file.hasNext()) { for (int i = 0; i < 4; i++) { System.out.println(file.nextLine()); } String ok = file.nextLine(); System.out.println("What is the correct ? "); } } catch (Exception e) { e.printStackTrace(); } } }</pre>	<pre>Question 1 ? A - Answer 1 B - Answer 2 C - Answer 3 A Question 2 ? A - Answer 1 B - Answer 2 C - Answer 3 C Question 3 ? A - Answer 1 B - Answer 2 C - Answer 3</pre>

<pre>String ans = user.next(); ans = ans.toUpperCase(); if (ans.length() > 1) { ans = ans.substring(0, 1); } if (ans.equals(ok)) { sum++; System.out.println("OK !\n"); } else { System.out.println("NO !\n"); } } System.out.println("\nResult:" + sum + " of 3"); } catch (FileNotFoundException e) { System.out.println("No questions !"); } } }</pre>	B
--	---

>java Quiz ↵

Zadanie: podaje prowadzący w każdej grupie laboratoryjnej

1.4. Plik MyFile.java – aplikacja konsolowa zapisująca do pliku tekstowego Myfile.txt – usunąć błędy !

```
import java.io.File;
import java.io.IOException;
public class MyFile {
    public void main(String[] args) {
        try {
            File file = new File("./Myfile.txt");
            if (file.createNewFile()) {
                System.out.println("Success!");
            } else {
                System.out.println("Error, file already exists.");
            }
            FileWriter writer = new FileWriter(file);
            writer.write("My file test data");
            writer.close();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
}
```

>java MyFile ↵

Zadania: Inne zadania aplikacji CLI podaje prowadzący w każdej grupie laboratoryjnej

2. Aplikacje okienkowe (GUI)

2.1. Biblioteka AWT `java.awt.*` - Program `AWTDemo.java`

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

public class AWTDemo extends Frame {
    public AWTDemo() {
        super("AWT Demo");
        prepareGUI();
    }
    public static void main(String[] args) {
        AWTDemo awtDemo = new AWTDemo();
        awtDemo.setVisible(true);
    }
    private void prepareGUI() {
        setSize(800,800);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent) {
                System.exit(0);
            }
        });
    }
    public void paint(Graphics g) {
        g.setColor(Color.RED);
        Font font = new Font("Serif", Font.BOLD, 54);
        g.setFont(font);
        g.drawString("AWT Demo", 50, 150);
        g.setColor(Color.BLUE);
        g.drawRect(100,200,150,150);
        g.setColor(Color.GREEN);
        g.fillRect(100,200,400,400);
    }
}
```

Zadanie: podaje prowadzący w każdej grupie laboratoryjnej

2.2. Biblioteka Swing `javax.swing.*` - Program `SwingProgram.java` – usunąć błędy !

```
package pl.edu.prz.jprokop;
import java.awt.*;

public class SwingProgram extends JFrame {
    public SwingConstructor() {
        super("SwingProgram");
        String input = JOptionPane.showInputDialog("Please enter your name");
        String name = "Hello " + input + " ! ";
        JLabel label = new JLabel(name, JLabel.CENTER);
        label.setFont(new Font("Dialog", Font.BOLD, 50));
        label.setBackground(Color.yellow);
        label.setForeground(Color.blue);
        label.setOpaque(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        add(label);
        setSize(600,400);
        //setVisible(true);
    }
    public static void main(String[] args) {
        new SwingProgram();
    }
}
```

Zadanie: zmodyfikować powyższą aplikację tak aby możliwe było przekazywanie z linii poleceń łańcucha tekstu wyświetlanego w oknie aplikacji oraz wielkości czcionki (zastosować metodę `Integer.parseInt()`).

2.3. Biblioteka JavaFX

<https://docs.oracle.com/javase/8/javafx/api/> - **JavaFX 8 API**

Przykład 1 - Plik `FXProgram.java`

```
package pl.edu.prz.jprokop;
import javafx.scene.*;
import javafx.scene.control.*;
import javafx.scene.image.*;
import javafx.scene.layout.*;

public class FXProgram extends Application {
    public static void main(String[] args) {
        Application.launch(args);
    }
    @Override
    public void start(Stage stage) {
        Image image = new Image("myImage.jpg");
        ImageView imageView = new ImageView(image);
        DatePicker datePicker = new DatePicker();
        VBox root = new VBox(10, datePicker, imageView);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.setTitle("MyTitle");
        //stage.show();
    }
}
```

Przykład 2 - Plik `PieChartSample.java`

```
package pl.edu.prz.jprokop;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.chart.*;
import javafx.scene.Group;

public class PieChartSample extends Application {
    @Override
    public void starts(Stage stage) {
        Scene scene = new Scene(new Group());
        stage.setTitle("Języki programowania");
        stage.setWidth(500);
        stage.setHeight(500);
        ObservableList<PieChart.Data> pieChartData
            = FXCollections.observableArrayList(
                new PieChart.Data("Java", 21),
                new PieChart.Data("C", 14),
                new PieChart.Data("C++", 6),
                new PieChart.Data("C#", 3.8),
                new PieChart.Data("PHP", 3),
                new PieChart.Data("Python", 3.2));
        final PieChart chart = new PieChart(pieChartData);
        chart.setTitle("Języki programowania");
    }
}
```

```
((Group) scene.getRoot()).getChildren().add(chart);
stage.setScene(scene);
//stage.show();
}
public static void main(String[] args) {
    launch(args);
}
}
```

Zadanie: podaje prowadzący w każdej grupie laboratoryjnej

3. Sposoby uruchamiania aplikacji Java

3.1. Uruchamianie aplikacji z linii poleceń

```
>java PierwszyProgram ↵
```

3.2. Uruchamianie za skryptów *.bat (Windows) lub *.sh (UNIX)

PierwszyProgram.bat

```
@echo off
javac PierwszyProgram.java
java PierwszyProgram
pause
```

```
>PierwszyProgram.bat ↵
```

3.3 . Uruchamianie aplikacji z archiwów Javy - pliki JAR

```
>jar ↵ Sprawdzić opcje narzędzia jar
```

■ Tworzenie nowego archiwum

```
>jar cvf nazwa_archiwum.jar plik1.class plik2.class ↵
>jar cvf nazwa_archiwum.jar * ↵
>jar cvf nazwa_archiwum.jar .class nazwa_podkatalogu/ .gif ↵
```

■ Wyświetlenie listy plików z archiwum

```
>jar tf nazwa_archiwum.jar ↵
```

■ Tworzenie rozpakowanej kopii archiwum

```
>jar xf nazwa_archiwum.jar ↵
```

■ Uruchomienie aplikacji z pliku archiwum

```
>java -jar app.jar ↵
```

Kroki kompilacji i uruchomienia aplikacji z pliku archiwum jar

```
> javac *.java ↵
> echo Main-Class: MyClass >manifest.txt ↵
> jar cvfm MyClass.jar manifest.txt *.class ↵
> jar cvfe MyClass.jar MyClass *.class ↵
> MyClass.jar ↵
> java -jar MyClass.jar ↵
```

3.4. JavaFX Package – javafxpackager

```
> javac -d . HelloWorld.java ↵
> javafxpackager -createjar -appclass jp.HelloWorldMain -srcdir . -outdir
  out -outfile helloworld.jar -v ↵
> javaw -jar out/ helloworld.jar ↵
Double-click helloworld.jar
```

4. Tworzenie dokumentacji programu za pomocą narzędzia *javadoc*

Utworzyć dokumentację programów za pomocą narzędzia *javadoc*

```
/**
 * @author Jan Prokop
 * @version 1.0
 */
```

```
>javadoc ↵
```

5. Java - narzędzia

- **Nakładki na pakiet JDK**

jGRASP – <http://www.jgrasp.org>
 JCreator – <http://www.jcreator.com>
 TextPad – <http://www.textpad.com>
 DrJava – <http://drjava.org>

- **IDE**

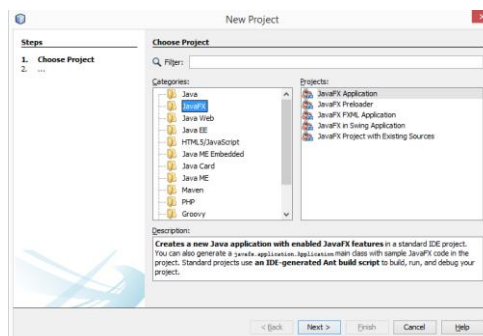
IntelliJ IDEA – <http://www.jetbrains.com/idea/>
 NetBeans IDE – <http://www.netbeans.org>
 Eclipse – <http://www.eclipse.org/downloads/>
 JDeveloper – <http://www.oracle.com/technology/products/jdev>

- **Inne**

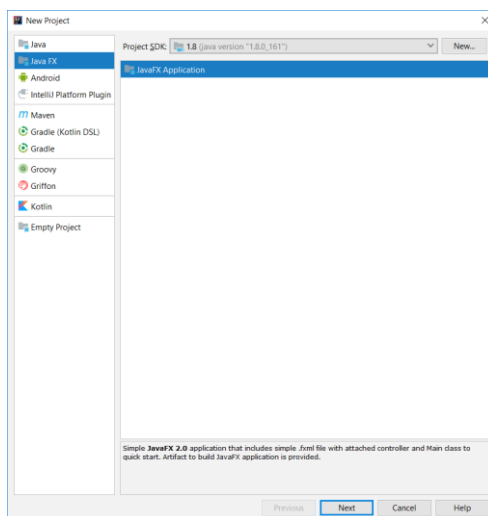
Ant – <http://ant.apache.org/>
 JUnit – <http://www.junit.org/>
 DbUnit – <http://www.dbunit.org/>
 Log4J – <http://logging.apache.org/log4j/>
 Tomcat – <http://tomcat.apache.org/>

6. IDE - Budowa aplikacji

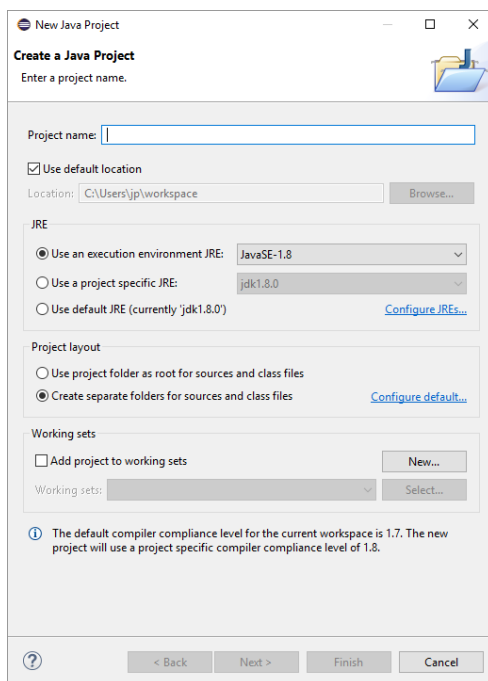
- **NetBeans** – <http://www.netbeans.org>



- **IntelliJ IDEA** - <http://www.jetbrains.com/idea/>



- **Eclipse IDE for Java Developers** - <http://eclipse.org/>



7. Zadania

Podaje prowadzący w każdej grupie laboratoryjnej