

# Programowanie obiektowe – zastosowanie języka Java SE

Wstęp do programowania  
obektowego w Javie

Autor: dr inż. Zofia Kruczkiewicz

# Java ?

## Java

### język programowania

- obiektowo zorientowany
- wysokiego poziomu

### platforma Javy

- z maszyny wirtualnej VM
- API (interfejs programowania aplikacji).

### **Rezultat**

- **niezależność od platformy,**
- **duże możliwości,**
- **stabilność,**
- **łatwość rozwoju,**
- **bezpieczeństwo**

### Rodzaje platform Javy:

- ◆ Java Platform, Standard Edition (Java SE)
- ◆ Java Platform, Enterprise Edition (Java EE)
- ◆ Java Platform, Micro Edition (Java ME)
- ◆ Java Platform CARD

# 1. Charakterystyka języka Java

1. **Obiektowy język Java** – składnia języka podobna do języka C++.

Pliki źródłowe:

**nazwa\_klasy\_publicznej.java,**

gdzie nazwa musi być nazwą klasy publicznej, zdefiniowanej w tym pliku.

2. **Kompilator** przetwarza program nazwa\_klasy\_publicznej.java na kod binarny zwany B-kod (bytecode, J-code)

Pliki po kompilacji:

**nazwa\_klasy\_publicznej.class**

B-kod może być zinterpretowana i wykonywana przez maszynę wirtualną Java (JVM Java Virtual Machine), czyli urządzenie logiczne

3. **Maszyna wirtualna Java** (JVM Java Virtual Machine).

JVM jest abstrakcyjnym komputerem, który wykonuje programy nazwa.class:

3.1. interpretator wbudowany w przeglądarkę WWW,

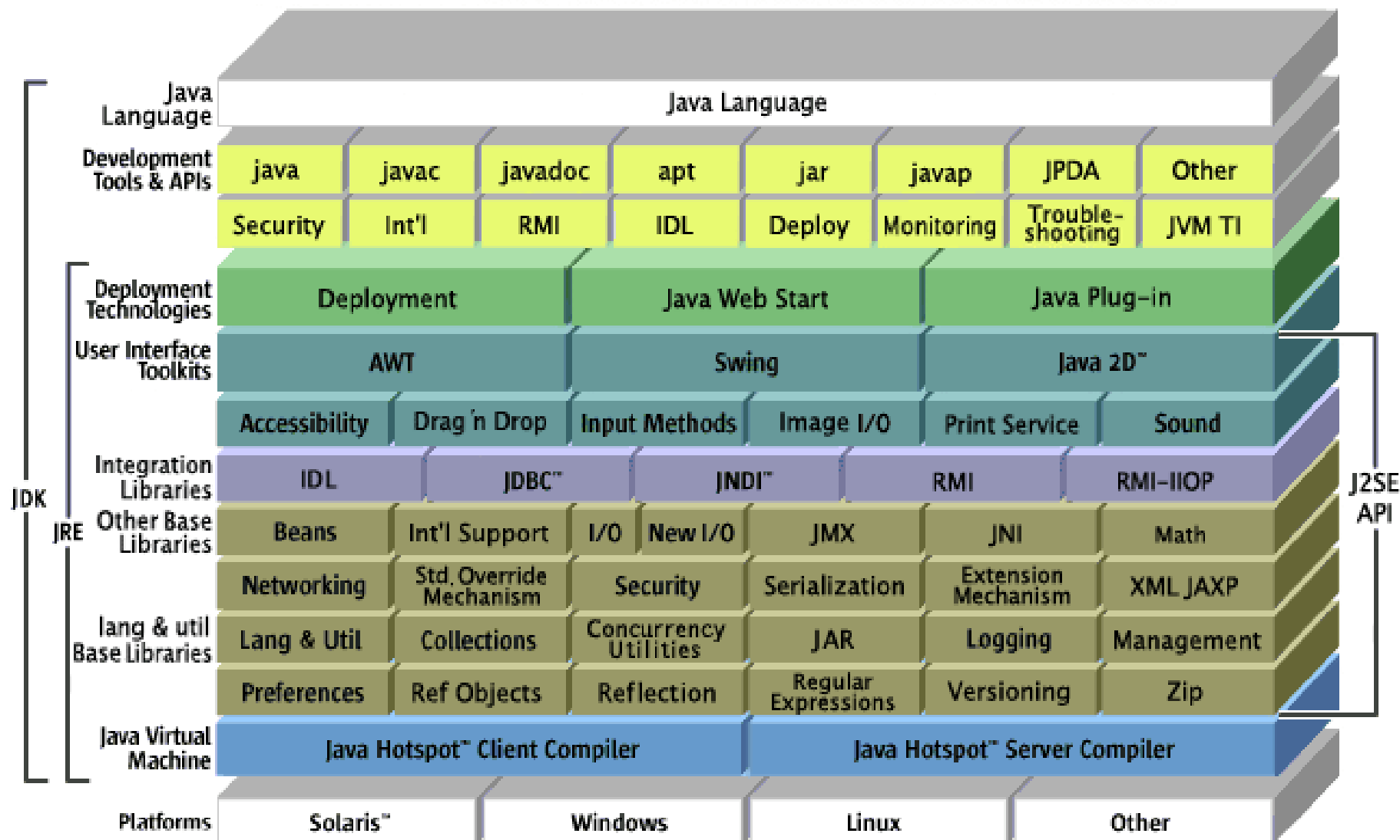
3.2. oddzielny program

3.3. Just-In-Time (przetworzenie nazwa.class na program wykonalny specyficzny dla danej maszyny)

4. **Biblioteka Javy** – pakiety z oprogramowaniem wspomagającym tworzenie programów działających w sieci np. Internet, umożliwiającym tworzenie interfejsu użytkownika, ogólnego przeznaczenia

# Java SE 5.0

Java™ 2 Platform Standard Edition 5.0



## 2. Tworzenie programu w Javie

### Aplikacja (application)

Program interpretujący aplikacje **java.exe** typu JVM jest uruchamiany w systemie operacyjnym.

Uruchamiana aplikacja (program użytkownika) zawiera między innymi jeden moduł źródłowy, którego klasa publiczna zawiera publiczną metodę klasową o nagłówku

```
public static void main(String args[])
```

### Aplet (applet)

Program interpretujący aplety jest wbudowany np. w przeglądarkę www.

Program typu aplet zawiera między innymi jeden moduł źródłowy, którego klasa publiczna zawiera między innymi podstawowe metody: **init()**, **start()**, **stop()**, **paint()**, **destroy()**

**Uwaga:** możliwe jest napisanie programu w Javie, który będzie pracował jako aplet i jako aplikacja.

## 2.1. Tekst źródłowy w Javie

```
public class Witaj
{
    public static void main(String args[])
    {
        System.out.print("Dzien dobry, nazywam się Jan Kowalski\n");
    }
}
```

## 2.2. Kompilacja

```
javac Witaj.java
```

gdzie położenie (katalog) programu **javac** (kompilator Javy) powinno być znane systemowi operacyjnemu, a katalog bieżący powinien zawierać plik źródłowy **Witaj.java**.

Zostanie wygenerowany plik **Witaj.class** z instrukcjami dla **JVM**.

## 2.3. Interpretacja

### java Witaj

interpretator java (położenie znane systemowi operacyjnemu)

- wyszuka plik o nazwie **Witaj.class** w katalogu bieżącym
- sprawdzi, czy klasa *Witaj* posiada publiczną metodę statyczną **main**
- wykona instrukcje zawarte w bloku funkcji **main**, czyli wyświetli na ekranie napis

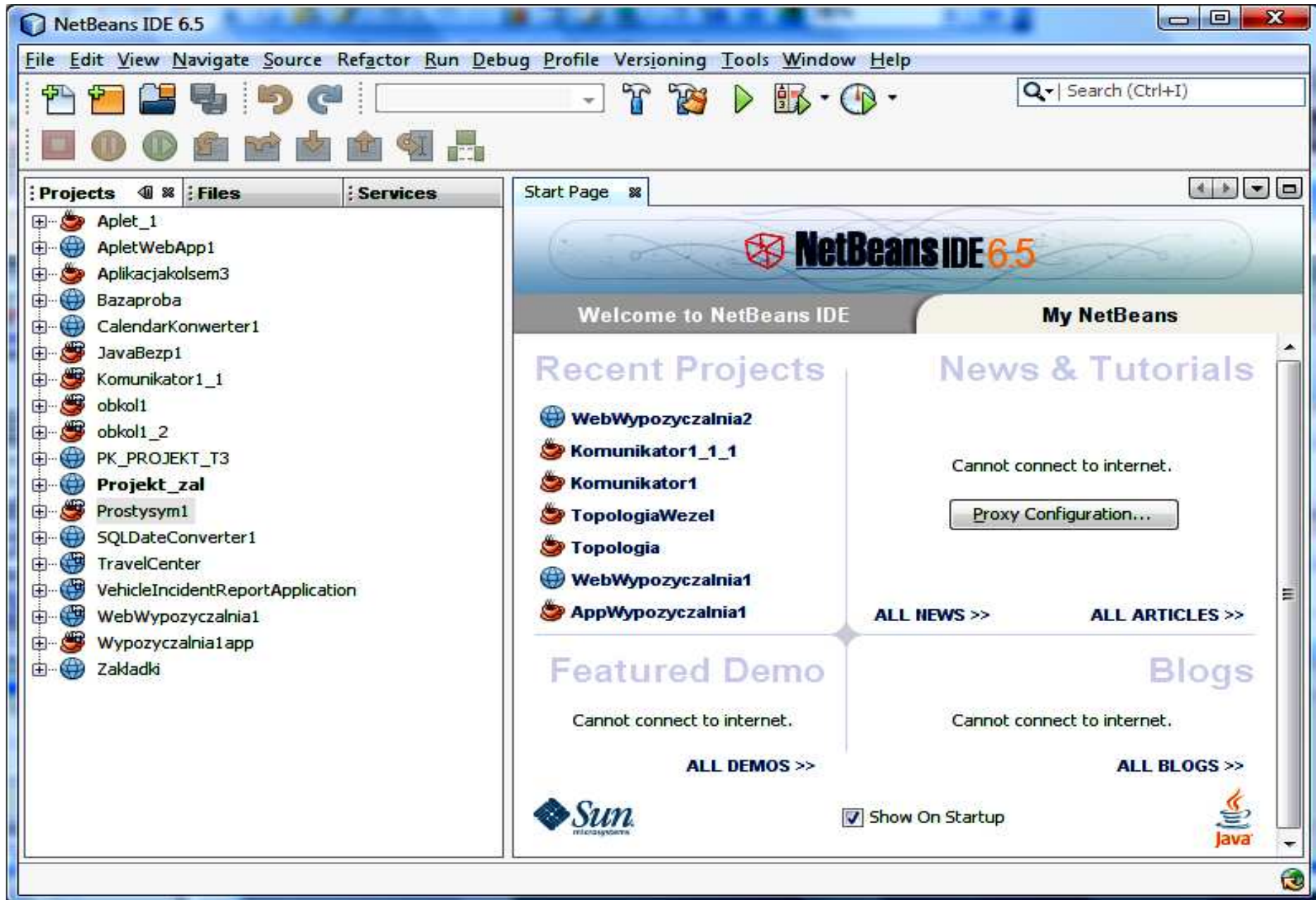
**Dzien dobry, nazywam się Jan Kowalski**

i przejdzie do następnego wiersza

### Uwagi:

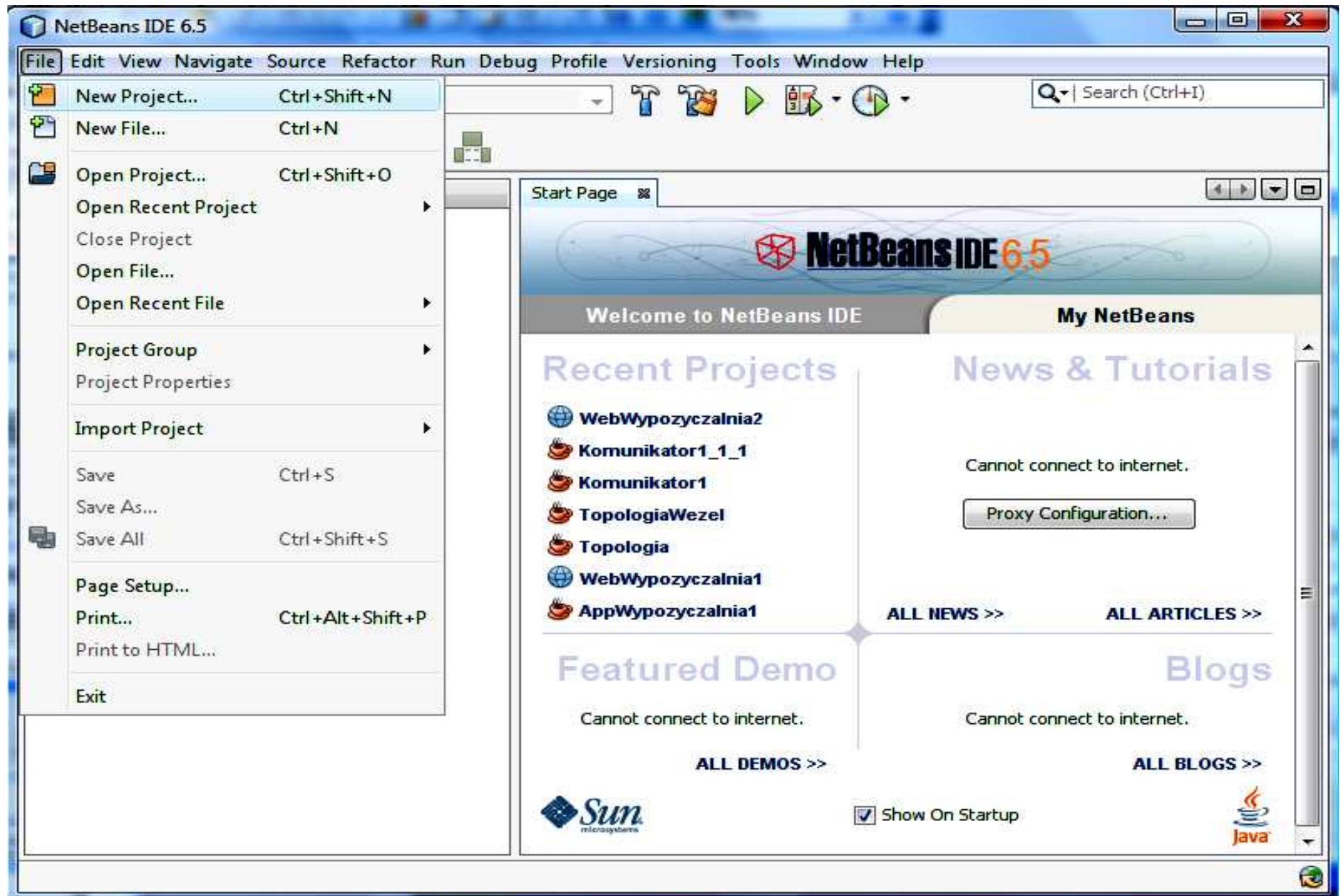
- do metody **main** z wiersza rozkazowego jako parametr jest przekazywana tablica **args** obiektów (łańcuchów) klasy **String** - w klasie **Witaj** jest ona pomijana
- każda instrukcja kończy się średnikiem
- standardowa klasa **System**:
  - a) zawiera statyczny obiekt składowy typu **PrintStream** o nazwie **out**
  - b) wywołanie **System.out.print** oznacza pisanie łańcucha typu **String** do standardowego strumienia wyjściowego, w tym wypadku ekranu
  - c) metoda **print** generuje jeden wiersz wyjściowy i powraca do metody **main**

# Przykład 1: Tworzenie aplikacji w środowisku NetBeans 6.5.

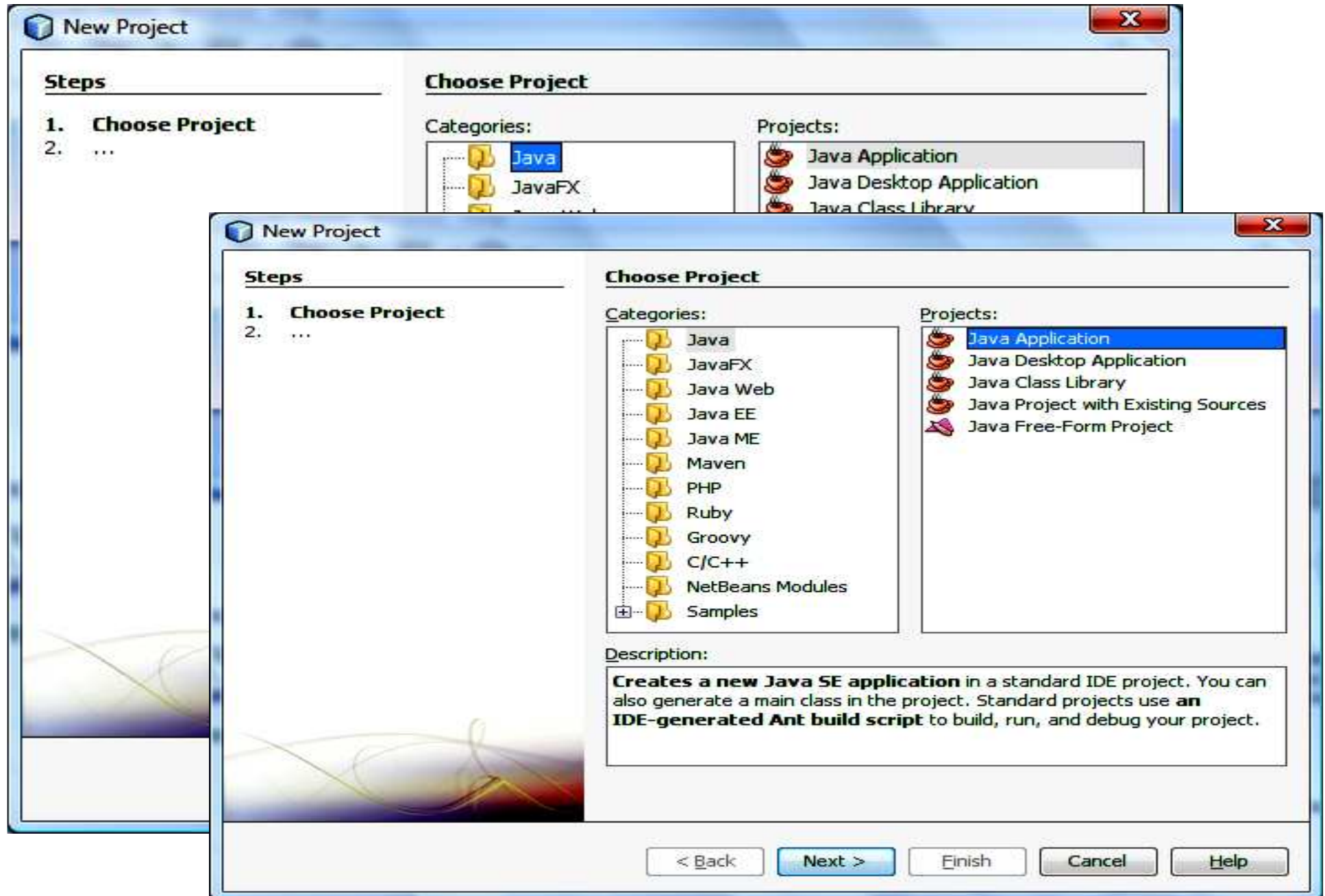




# Zakładanie nowego projektu



# Wybór kategorii projektu Java typu Java Application



## Domyślny formularz projektu po naciśnięciu klawisza **Next**

**New Java Application**

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

Set as Main Project

Podanie własnych ustawień projektu: nazwy projektu (**Project Name**), katalogu (**Project Location**) oraz nazwy klasy głównej zawierającej funkcję main (**Create Main Class**)

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name: Witaj

Project Location: E:\moodle\ZK\Modul1

Project Folder: E:\moodle\ZK\Modul1\Witaj

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class witaj.WitajJavo

Set as Main Project

< Back   Next >   **Finish**   Cancel   Help

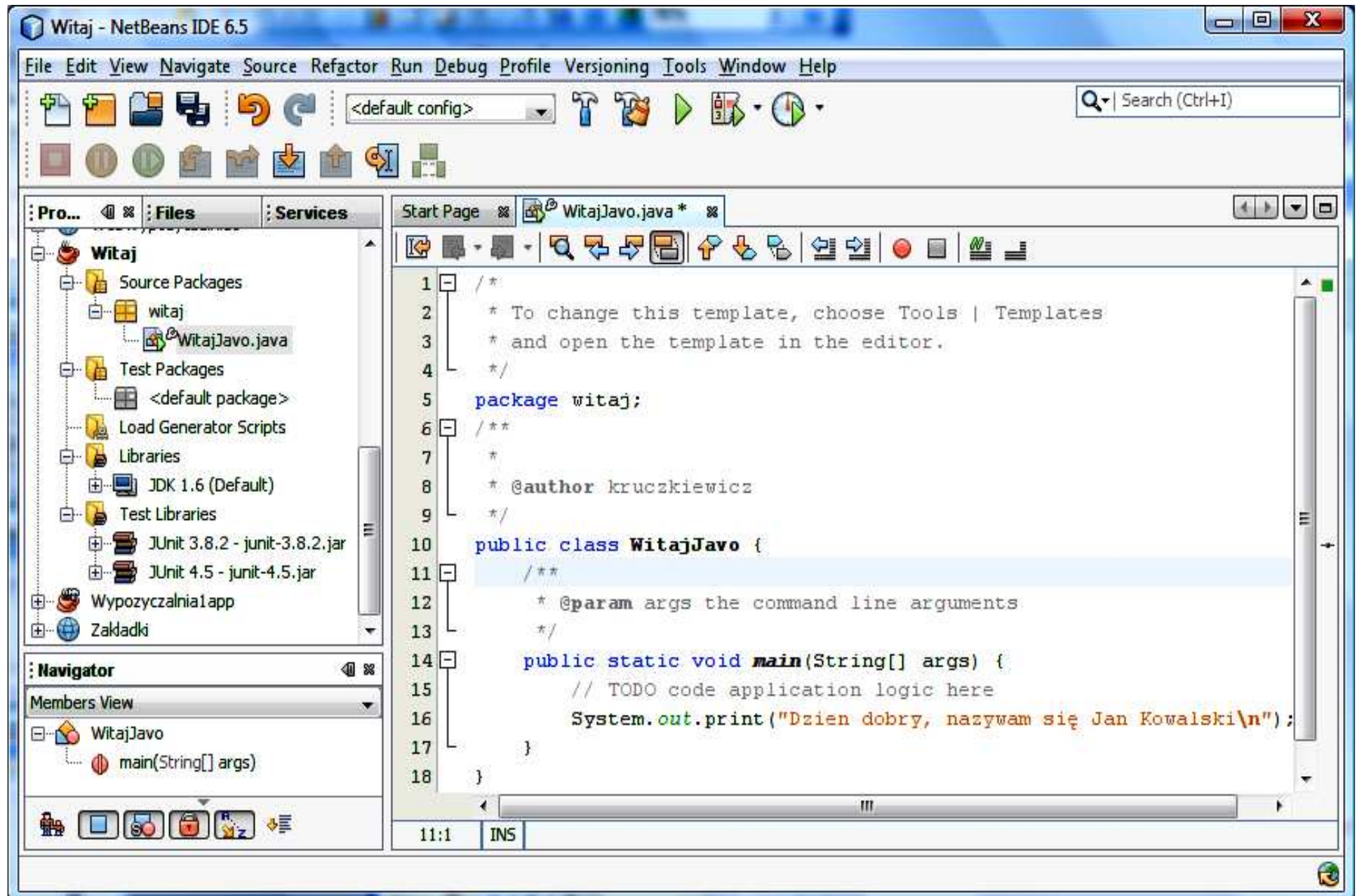
# Formularz edycji programu źródłowego

The screenshot displays the NetBeans IDE 6.5 interface. The title bar reads "Witaj - NetBeans IDE 6.5". The menu bar includes "File", "Edit", "View", "Navigate", "Source", "Refactor", "Run", "Debug", "Profile", "Versioning", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and development actions. The left sidebar shows a project tree with "Witaj" selected, containing "Source Packages" (with "witaj" sub-package) and "Test Packages". The "Navigator" window shows the "Members View" for "WitajJavo" with a "main(String[] args)" method. The main editor window shows the source code for "WitajJavo.java":

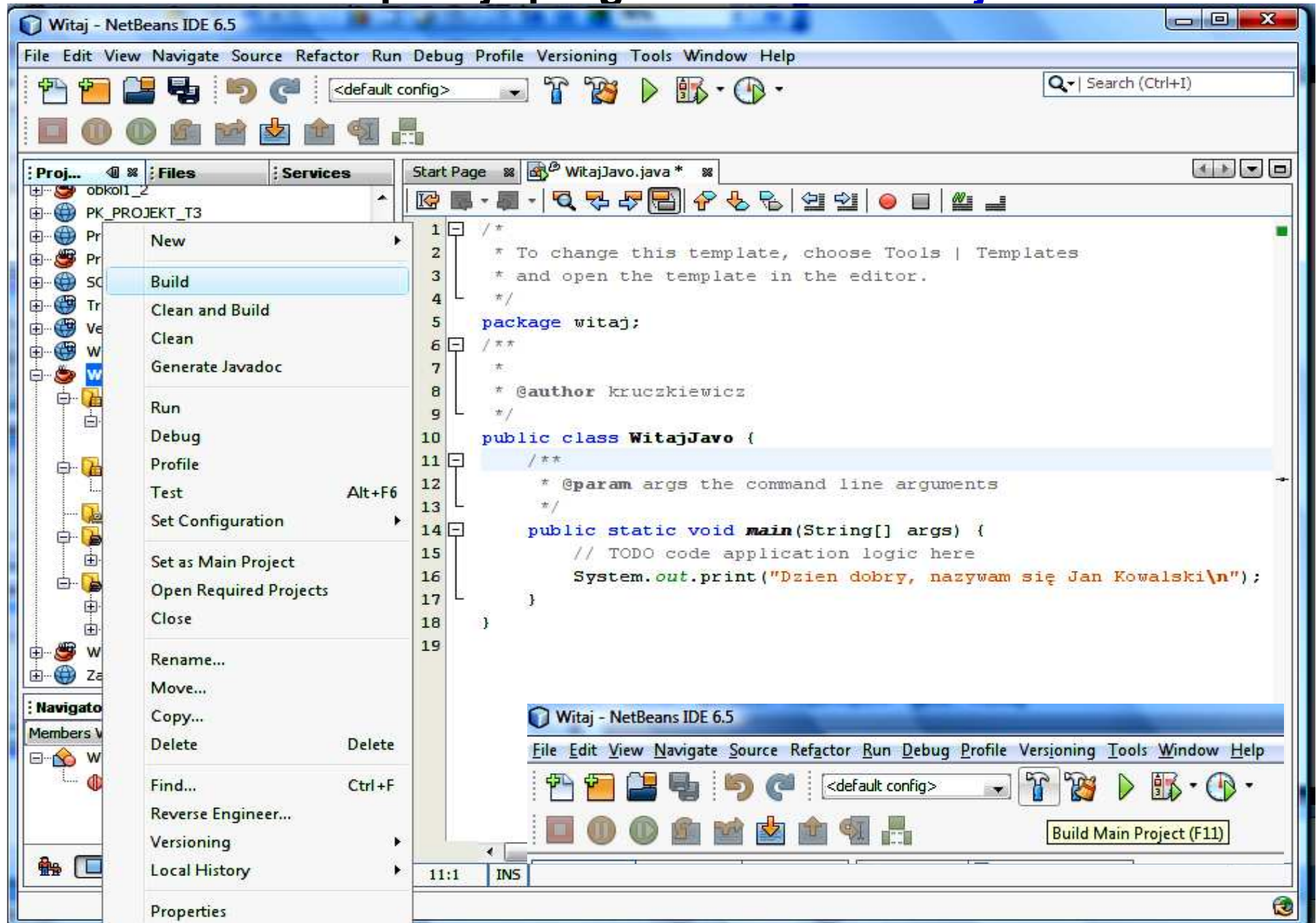
```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package witaj;
7
8  /**
9   *
10  * @author kruczkiewicz
11  */
12  public class WitajJavo {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20
21 }
```

The status bar at the bottom shows "1:1" and "INS". The bottom right corner of the window shows the path "E:\moooc".

# Wpisanie elementarnej instrukcji w funkcji **main**



# Kompilacja programu – Build Project



# Komunikaty z przebiegu kompilacji w oknie Output

The screenshot displays the NetBeans IDE 6.5 interface. The main editor window shows the source code for `WitajJava.java`. The code defines a package `witaj` and a public class `WitajJava` with a `main` method. The `main` method prints the message "Dzien dobry, nazywam się Jan Kowalski\n".

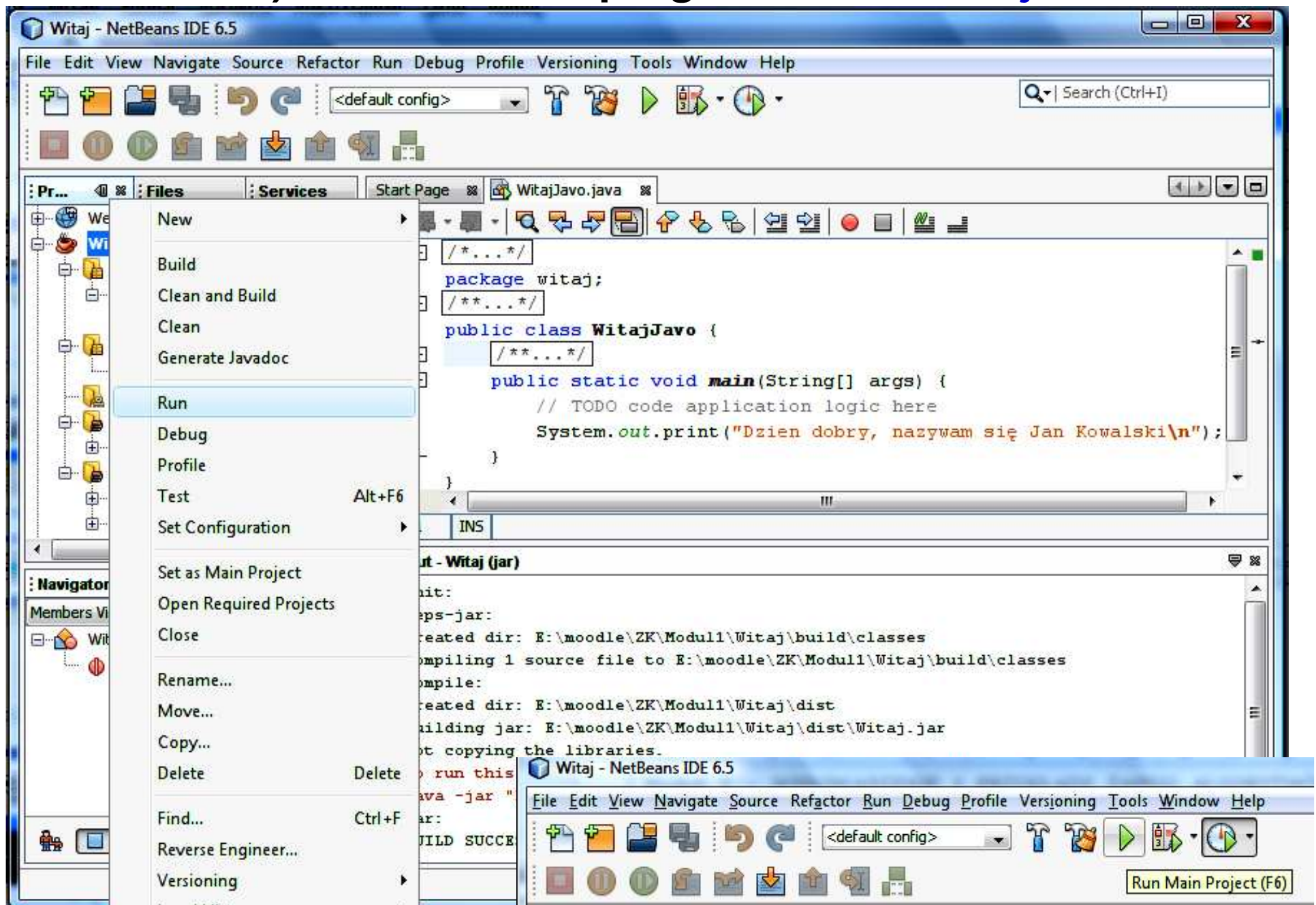
```
1  /*...*/
5  package witaj;
6  /**...*/
10 public class WitajJava {
11     /*...*/
14     public static void main(String[] args) {
15         // TODO code application logic here
16         System.out.print("Dzien dobry, nazywam się Jan Kowalski\n");
17     }
18 }
```

The `Output - Witaj (jar)` window at the bottom shows the compilation process:

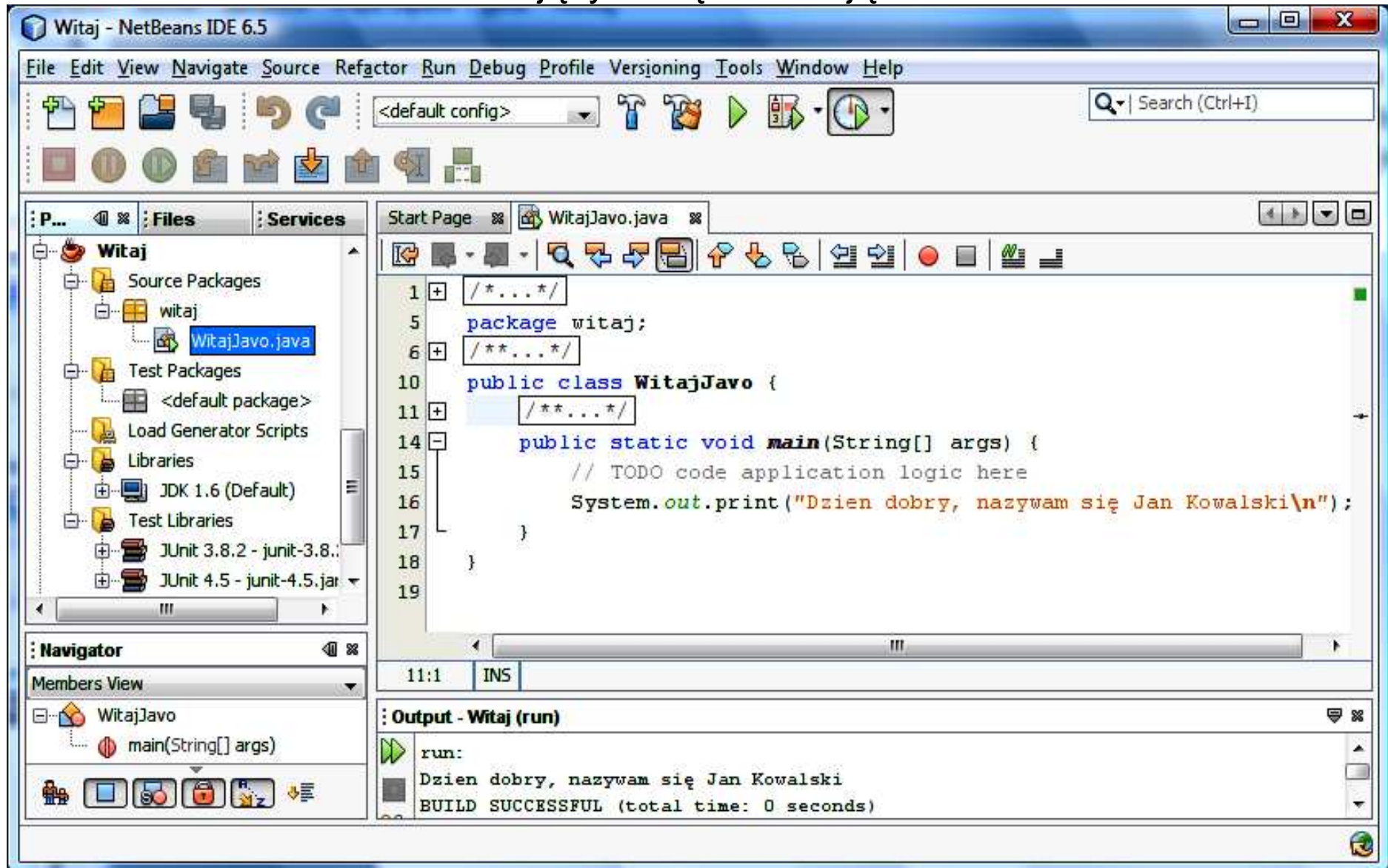
```
init:
deps-jar:
Created dir: E:\moodle\ZK\Modull\Witaj\build\classes
Compiling 1 source file to E:\moodle\ZK\Modull\Witaj\build\classes
compile:
Created dir: E:\moodle\ZK\Modull\Witaj\dist
Building jar: E:\moodle\ZK\Modull\Witaj\dist\Witaj.jar
Not copying the libraries.
To run this application from the command line without Ant, try:
java -jar "E:\moodle\ZK\Modull\Witaj\dist\Witaj.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```



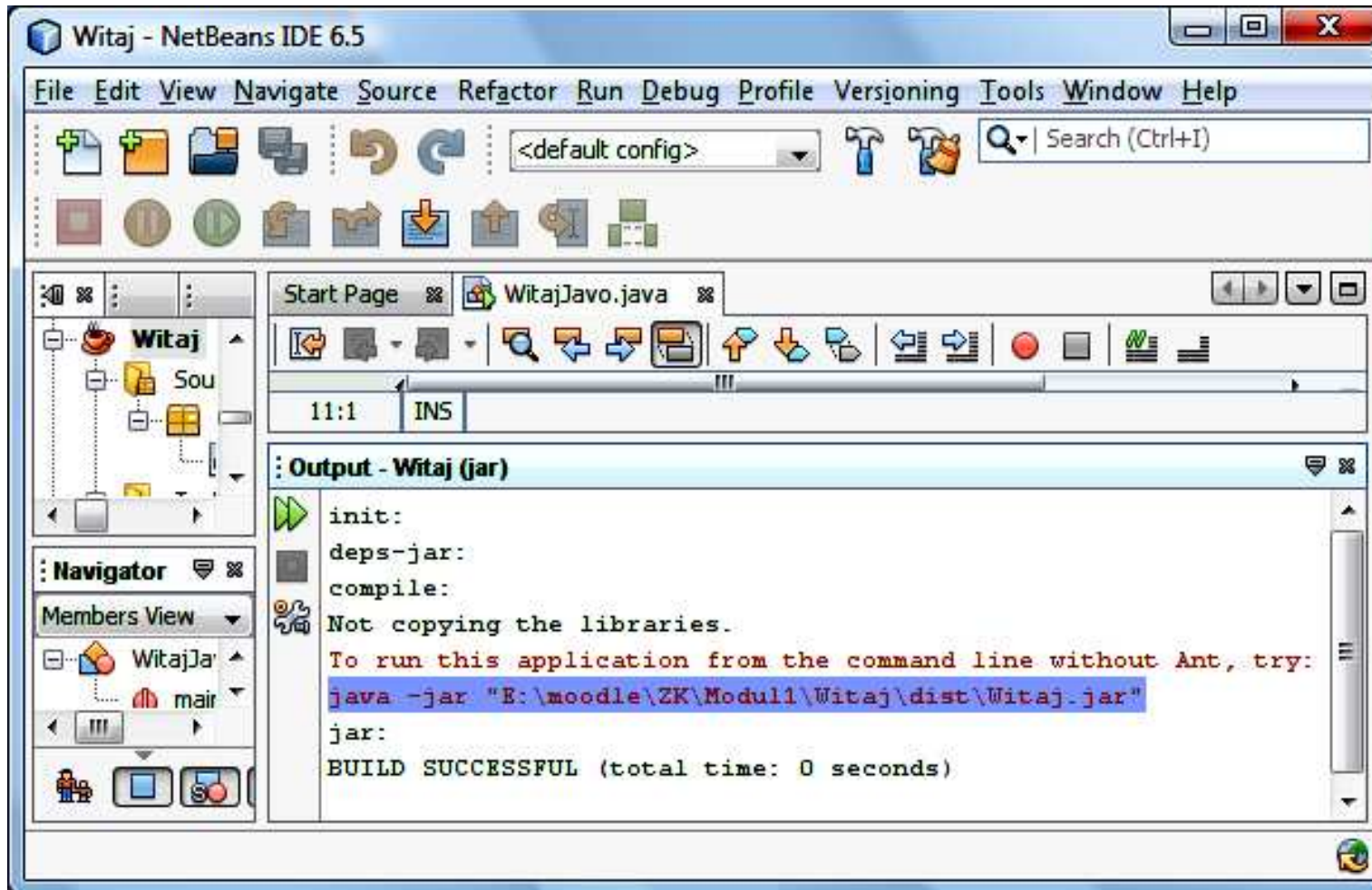
## a) Uruchamianie programu – Run Project



Wykonanie programu za pomocą VMT w środowisku narzędzia NetBeans  
5.5.1 – okno **Output**. W środowisku **NetBeans** tworzony jest plik spakowany typu **jar**, który może zawierać wiele plików, wśród których znajduje się plik zawierający klasę z funkcją **main**.



## b) Uruchomienie programu konsolowego z linii poleceń – skopiowanie łańcucha uruchamiającego program do schowka



[java -jar "E:\moodle\ZK\Modul1\Witaj\dist\Witaj.jar"](#)

Projects | Files | Services


- VehicleIncidentReportApplication
- WebWypożyczalnia1
- Witaj
  - Source Packages
  - witaj

Start Page | WitajJavo.java

```
1  /*...*/
5  package witaj;
6  /**...*/
10 class WitajJavo {
    static void main(String[] args) {
        // code application logic here
        System.out.print("Dzien dobry, nazywam się J
```

- Windows Defender
- Windows Live Messenger — pobieranie
- Windows Media Player
- Windows Movie Maker
- Windows Update
- Akcesoria
  - Centrum mobilności w systemie Windows
  - Centrum synchronizacji
  - Eksplorator Windows
  - Kalkulator
  - Narzędzie Wycinanie
  - Notatnik
  - Paint
  - Pasek boczny systemu Windows
  - Podłączanie pulpitu zdalnego
  - Połącz z projektorem sieciowym
  - Rejestrator dźwięku
  - System Windows — Zapraszamy!
  - Uruchom
  - Wiersz polecenia
  - Wykonuje funkcje tekstowe (wiersza polecenia).
  - Narzędzia systemowe
  - Tablet PC
  - Ułatwienia dostępu

Wstecz



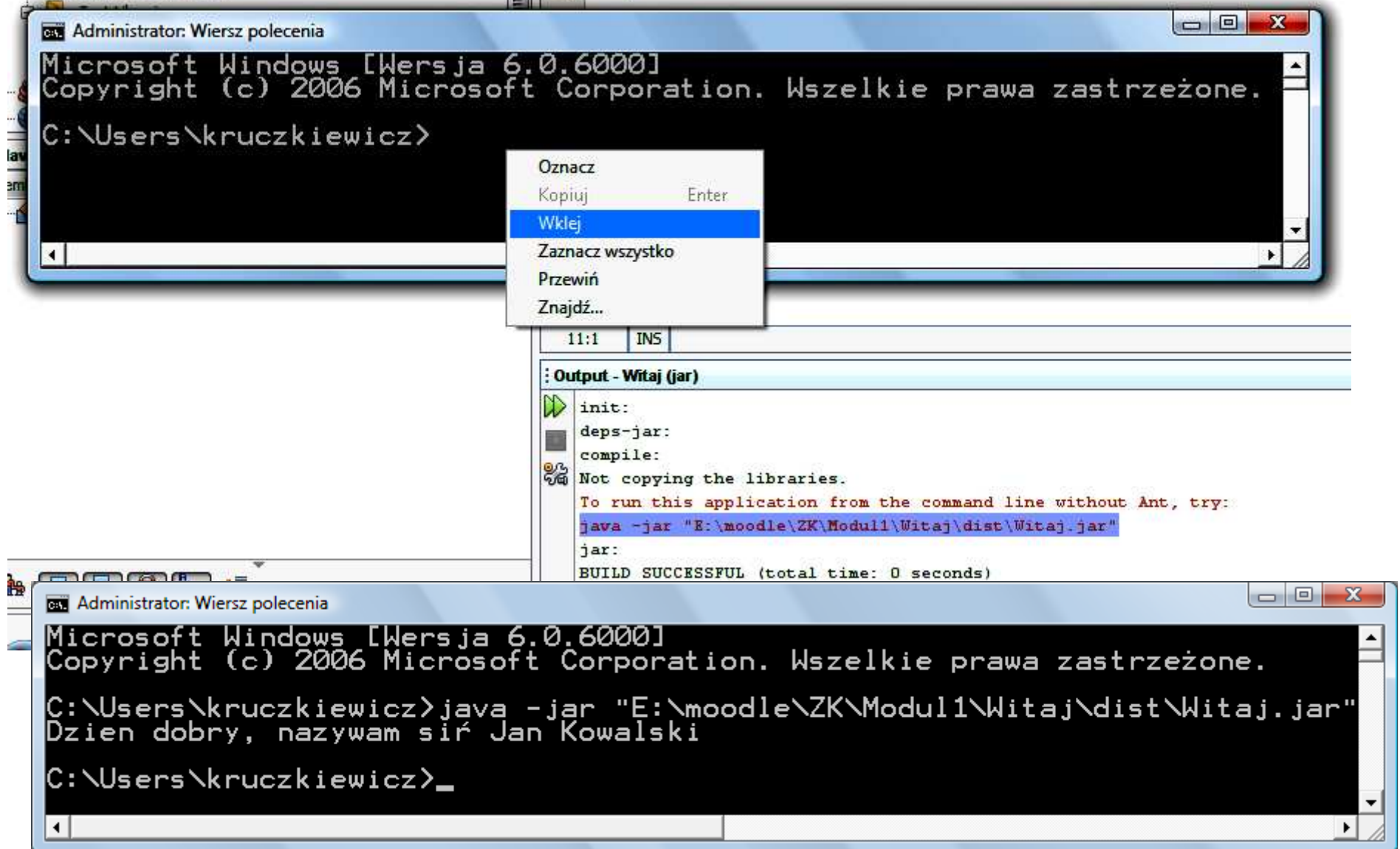
- kruczkiewicz
- Dokumenty
- Obrazy
- Muzyka
- Wyszukaj
- Bieżące elementy
- Komputer
- Sieć
- Połącz z
- Panel sterowania
- Programy domyślne
- Pomoc i obsługa techniczna

```
aries.
tion from the command line without Ant, try:
\\ZK\Modull\Witaj\dist\Witaj.jar"
total time: 0 seconds)
```

Rozpocznij wyszukiwanie

Google | 3 Eksplor... | 2 Micros... | Microsoft ... | 2 Java(T... | Bez tytułu -...

# Uruchomienie z linii poleceń programu typu aplikacja: przeniesienie łańcucha uruchomienia programu ze „schowka” przez naciśnięcie prawego klawisza myszy



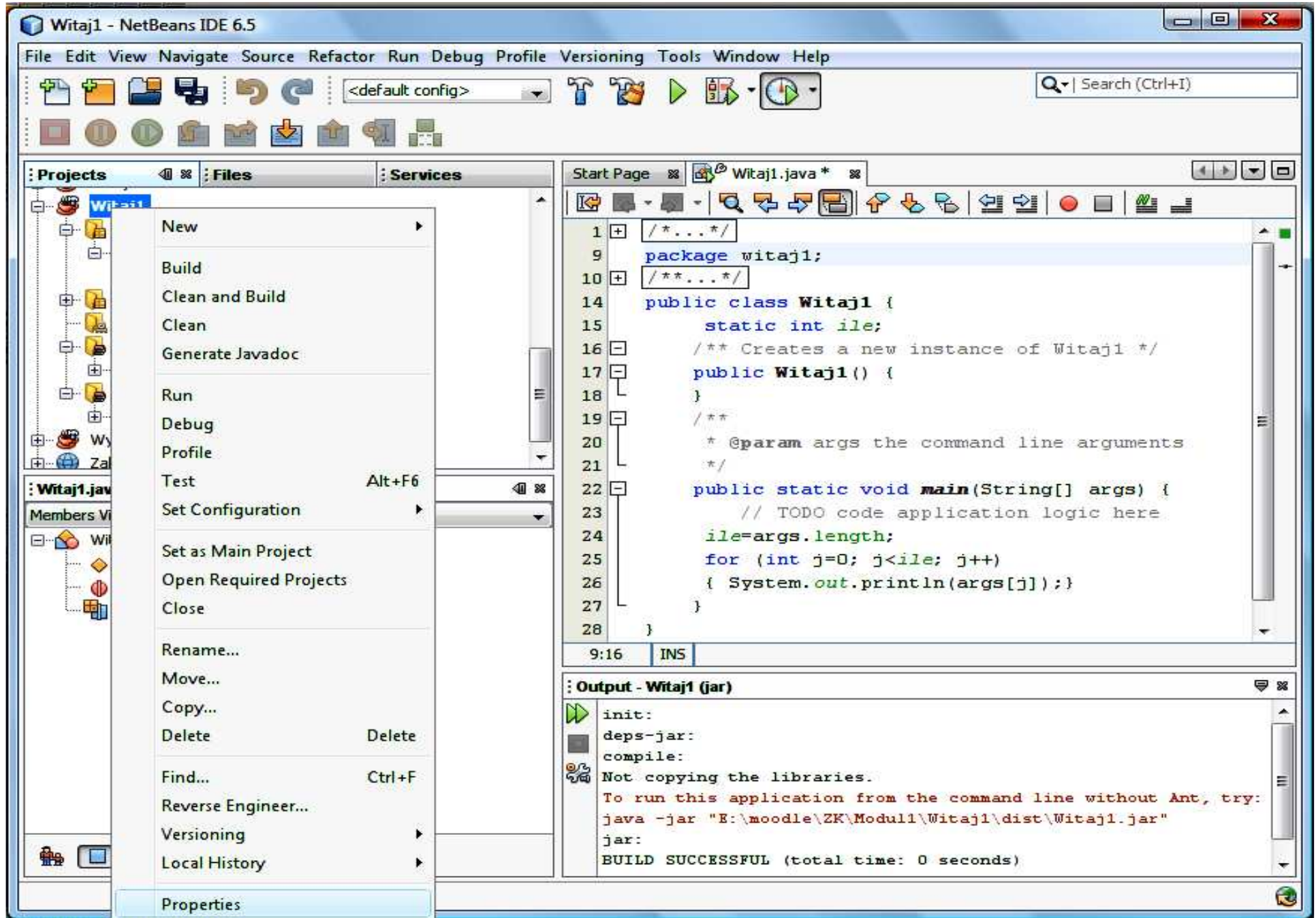
Przykład 2: Wywołanie programu z listą argumentów  
**java Witaj1 Jan Kowalski**

```
public class Witaj1 // klasa publiczna
{
    static int ile; //składowa klasowa

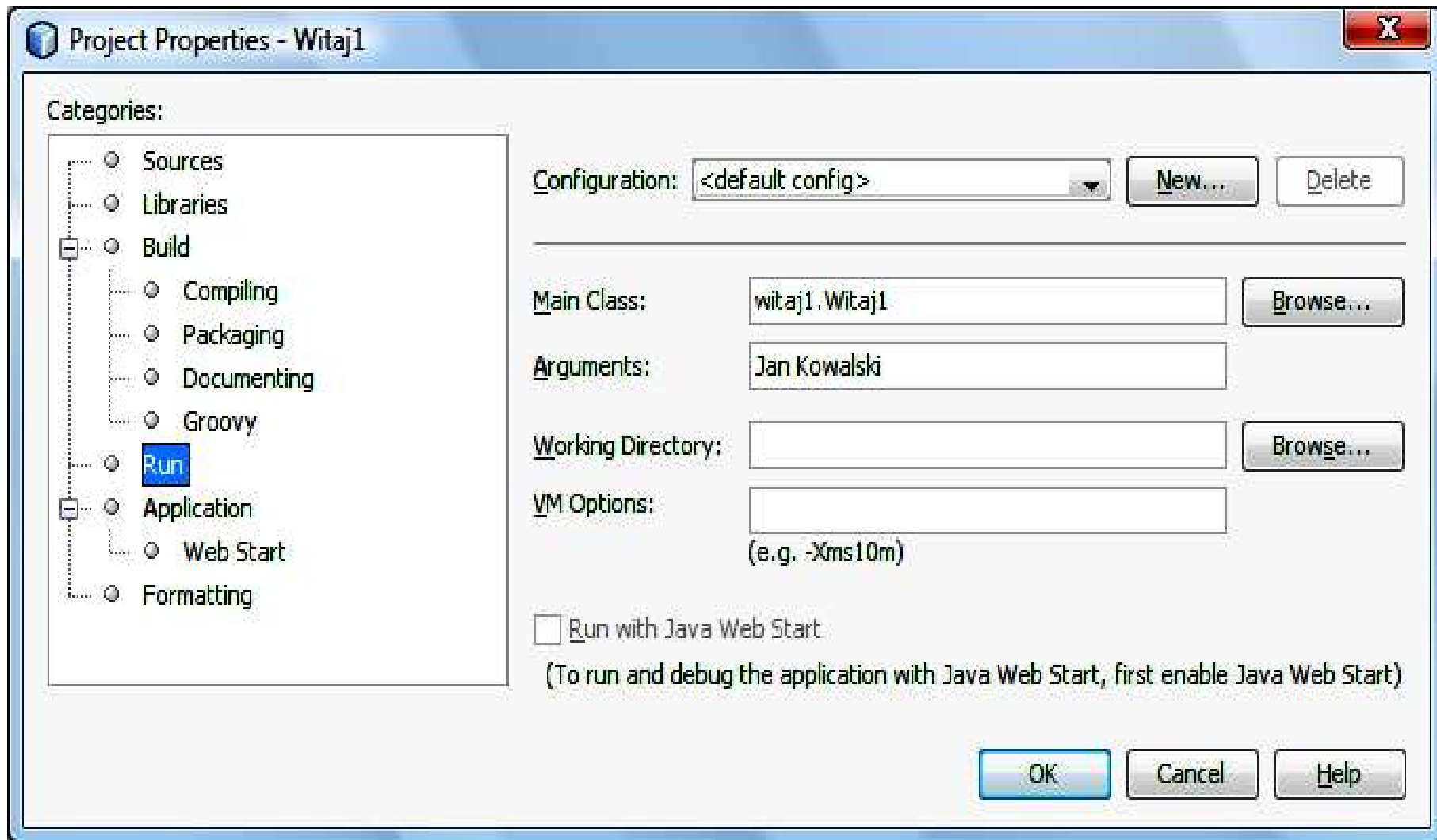
    public static void main(String args[])
    {
        //pobranie liczby parametrów (w przykładzie 2)
        // ile musi być składową typu static !
        ile=args.length;
        for (int j=0; j<ile; j++)
            //args[0] – Jan (łańcuch bez białych znaków)
            //args[1] - Kowalski
            System.out.println(args[j]);
    }
}
```

----- programowanie -----

# Wybór opcji **Properties** aktywnego projektu



## W oknie **Properties** wybór opcji **Run** i wpisanie łańcucha w linii **Arguments**





# Wpisanie tekstu programu korzystającego z argumentów programu i kompilacja **Build Project**

The screenshot displays the NetBeans IDE 6.5 interface for a project named "Witaj1". The main editor window shows the source code for "Witaj1.java":

```
1  /*...*/
9  package witaj1;
10 /*...*/
14 public class Witaj1 {
15     static int ile;
16     /** Creates a new instance of Witaj1 */
17     public Witaj1() {
18     }
19     /**
20      * @param args the command line arguments
21      */
22     public static void main(String[] args) {
23         // TODO code application logic here
24         ile=args.length;
25         for (int j=0; j<ile; j++)
26             { System.out.println(args[j]);}
27     }
28 }
```

The left sidebar shows the project structure with "Witaj1" as the main package. Below it, the "Witaj1.java - Navigator" window shows the class members: "Witaj1()", "main(String[] args)", and "ile : int".

The bottom output window, titled "Output - Witaj1 (jar)", shows the following build process:

```
init:
deps-jar:
compile:
Building jar: E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar
Not copying the libraries.
To run this application from the command line without Ant, try:
java -jar "E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

## a) Uruchomienie w środowisku NetBeans

The screenshot displays the NetBeans IDE 6.5 environment. The main window is titled "Witaj1 - NetBeans IDE 6.5". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Versioning, Tools, Window, and Help. The toolbar contains various icons for file operations and execution. The "Files" view on the left shows a project named "Witaj1" with a sub-package "witaj1" containing "Witaj1.java". The "Witaj1.java - Navigator" view shows the class structure with members "Witaj1()", "main(String[] args)", and "ile : int". The code editor displays the following Java code:

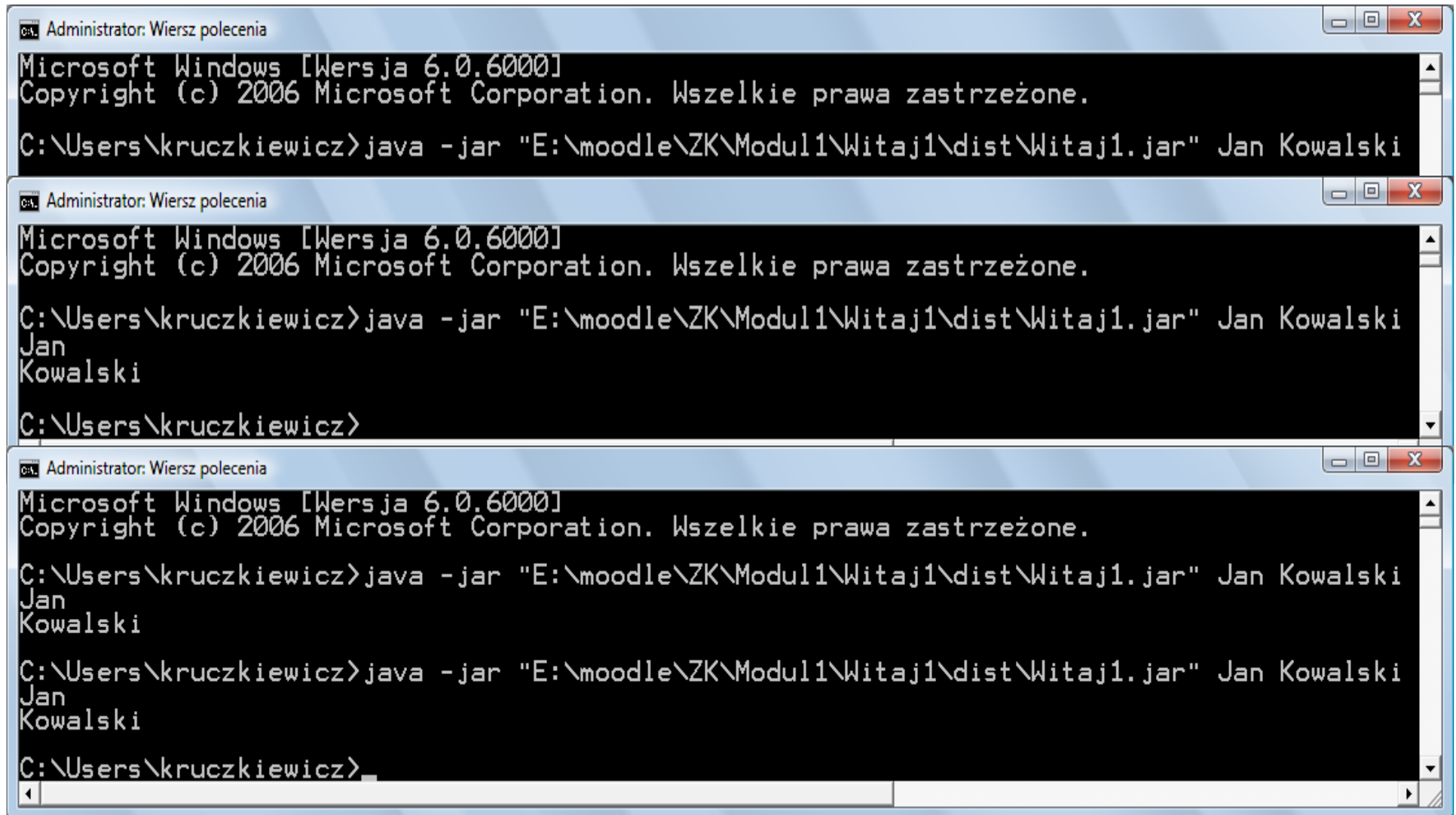
```
1  /*...*/
9  package witaj1;
10 /*...*/
14 public class Witaj1 {
15     static int ile;
16     /** Creates a new instance of Witaj1 */
17     public Witaj1() {
18     }
19     /**
20      * @param args the command line arguments
21      */
22     public static void main(String[] args) {
23         // TODO code application logic here
24         ile=args.length;
25         for (int j=0; j<ile; j++)
26             { System.out.println(args[j]);}
27     }
```

The "Output - Witaj1 (run)" window shows the execution results:

```
init:
deps-jar:
Compiling 1 source file to E:\moodle\ZK\Modull\Witaj1\build\classes
compile:
run:
Jan
Kowalski
BUILD SUCCESSFUL (total time: 1 second)
```

## b) Uruchomienie programu z linii poleceń

Przeniesienie ze schowka, naciskając prawy klawisz myszy, łańcucha uruchomienia pobranego z okienka **Output** po wykonaniu **Build Project** dla programu z **przykładu 2** i dopisaniu łańcucha **Jan Kowalski**. Po jednorazowym przeniesieniu łańcucha można powtórzyć uruchomienie programu z linii poleceń naciskając klawisz **F5**.



```
Administrator: Wiersz polecenia
Microsoft Windows [wersja 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\kruczkiewicz>java -jar "E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar" Jan Kowalski

Administrator: Wiersz polecenia
Microsoft Windows [wersja 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\kruczkiewicz>java -jar "E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar" Jan Kowalski
Jan
Kowalski

C:\Users\kruczkiewicz>

Administrator: Wiersz polecenia
Microsoft Windows [wersja 6.0.6000]
Copyright (c) 2006 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\kruczkiewicz>java -jar "E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar" Jan Kowalski
Jan
Kowalski

C:\Users\kruczkiewicz>java -jar "E:\moodle\ZK\Modul1\Witaj1\dist\Witaj1.jar" Jan Kowalski
Jan
Kowalski

C:\Users\kruczkiewicz>
```