



Uniwersytet Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



Programowanie w języku Java

w ramach projektu

„Trzecia Misja Uniwersytetu Ekonomicznego we Wrocławiu dla dzieci i młodzieży”

Część I

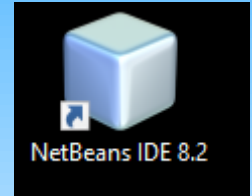
Rok szkolny 2021/22

Prowadzący: dr inż. Piotr Tutak

Przedstawiamy się - lista obecności

Informacje wstępne

- Środowisko programistyczne: NetBeans IDE 8.2
- Język programowania: JAVA



NetBeans IDE

NetBeans IDE jest to środowisko programistyczne (Integrated Development Environment) służące do tworzenia, kompilowania, uruchamiania i testowania programów.

Całe IDE jest napisane w Javie, jednak umożliwia ono również tworzenie programów w innych językach. Ponadto dostępne są również moduły rozszerzające jego możliwości np. usługi sieciowe, aplikacje mobilne.

NetBeans IDE jest produktem dostępnym za darmo oraz bez żadnych ograniczeń co do jego używania.



Uniwersytet
Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



JAVA – wprowadzenie cz.1

JAVA (wym. dżawa) - obiektowy język programowania

- Stworzony przez grupę Jamesa Goslinga z firmy Sun Microsystems z Kalifornii w 1990 roku
- Początkowa nazwa to OAK czyli dąb
- Podstawowe koncepcje przejęte z języka Smalltalk oraz z języka C++ (duża część składni i słów kluczowych)
- Uniwersalny język programowania, raz napisany kod można wykorzystać w dowolnym środowisku, do którego przeniesiono JVM (Java Virtual Machine, jest to rodzaj wirtualnego komputera)



Uniwersytet
Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

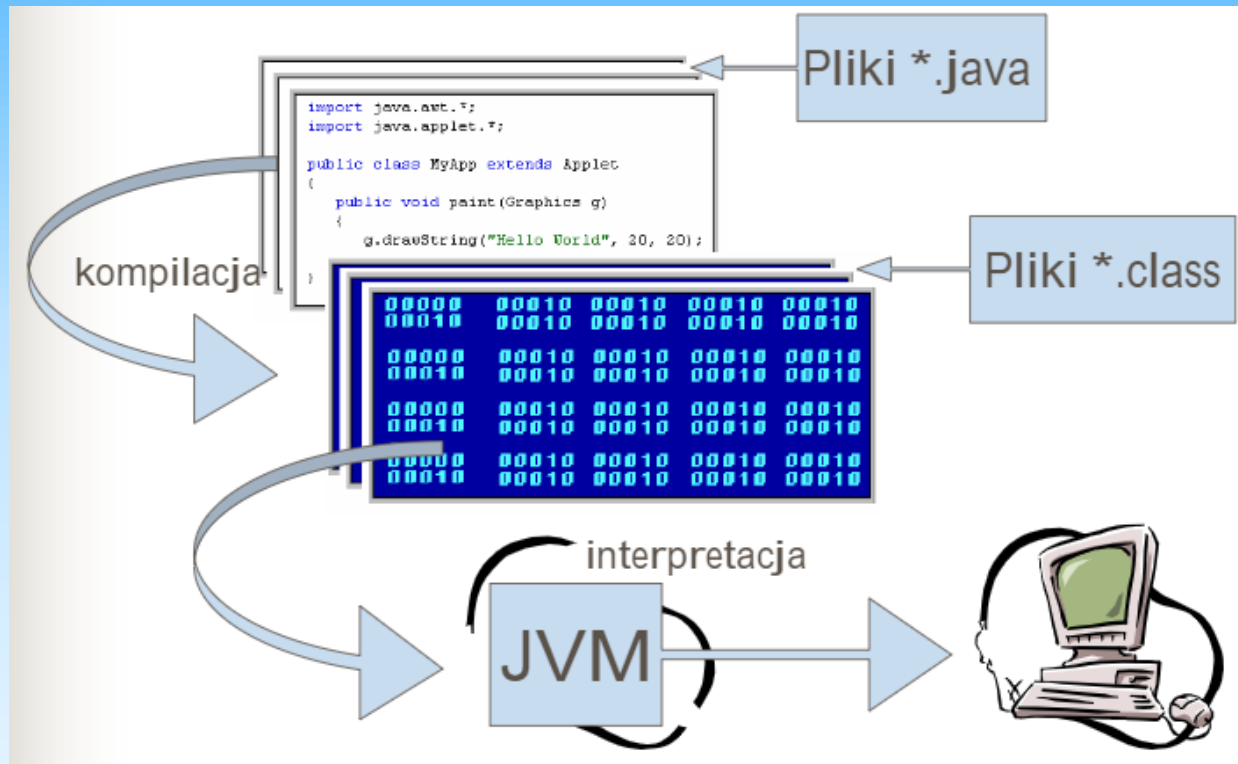
Unia Europejska
Europejski Fundusz Społeczny



JAVA – wprowadzenie cz.2

- Dzięki standaryzacji maszyny wirtualnej, programy napisane w Javie są uniwersalne tzn. wykonują się identycznie w każdym systemie operacyjnym
- Programy napisane w Javie są kompilowane do poziomego kodu pośredniego, nazywany kodem bajtowym Javy
- Kod bajtowo jest interpretowany przez JVM do postaci programu wykonywalnego dla danego systemu operacyjnego

JAVA – wprowadzenie cz.3



1. Kod źródłowy
2. Kompilator
3. Kod wynikowy
4. Maszyna wirtualna

JAVA – Edytory

Zintegrowane środowiska programistyczne

Borland JBuilder – <http://www.borland.com/products/download/>

Eclipse – <http://www.eclipse.org/>

IBM VisualAge for Java – <http://www7.software.ibm.com/vad.nsf>

JCreator – <http://www.jcreator.com/>

Kawa – <http://www.macromedia.com/>



NetBeans – <http://www.netbeans.org/>

Sun Forte for Java – <http://www.sun.com/forte/ffj/index.html>

Sun One Studio – <http://forte.sun.com/ffj/index.html>

VIM – <http://www.vim.org/>

Instalacja programu NetBeans cz. 1

1

java jdk netbeans

2

www.oracle.com › java › downloads ▾ Tłumaczenie strony

JDK 8u111 with NetBeans 8.2 - Oracle

This distribution of the JDK includes the Java SE bundle of NetBeans IDE, which is a powerful integrated development environment for developing applications ...

<https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

Overview Downloads Documentation Community Technologies Training

JDK 8u111 with NetBeans 8.2

This distribution of the JDK includes the Java SE bundle of NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the JDK 8u111 and NetBeans 8.2 Cobundle License Agreement to download this software.

☐ Accept License Agreement ☒ Decline License Agreement

Java SE and NetBeans Cobundle (JDK 8u111 and NB 8.2)		
Product / File Description	File Size	Download
Linux x86	286.73 MB	jdk-8u111-nb-8_2-linux-i586.sh
Linux x64	282.57 MB	jdk-8u111-nb-8_2-linux-x64.sh
Mac OS X x64	342.99 MB	jdk-8u111-nb-8_2-macosx-x64.dmg
Windows x86	317.21 MB	jdk-8u111-nb-8_2-windows-i586.exe
Windows x64	326.03 MB	jdk-8u111-nb-8_2-windows-x64.exe

Informacje o systemie
Aplikacja

Aplikacje

- System Configuration >
- NI Distributed System Manager 2020 >
- NI Distributed System Manager 2018 >
- Faksowanie i skanowanie w systemie Windows >
- Narzędzia administracyjne systemu Windows >
- Diagnostyka pamięci systemu Windows >

Wyszukaj w sieci Web

SYSTEM - Zobacz wyniki z sieci Web >

Dokumenty (7+)

Foldery

Ustawienia

SYSTEM

Informacje o systemie
Aplikacja

Otwórz

4

3

6

Informacje o systemie

Plik Edycja Widok Pomoc

Podsumowanie systemu

Element	Wartość
Nazwa systemu operacyjnego	Microsoft Windows 10 Pro
Wersja	10.0.18363 Kompilacja 18363
Dodatkowy opis systemu oper...	Niedostępne
Producent systemu operacyjne...	Microsoft Corporation
Nazwa systemu	DESKTOP-KMQIQC6
Producent systemu	Hewlett-Packard
Model systemu	HP ZBook 17
Type systemu	x64-based PC

5



Uniwersytet
Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój

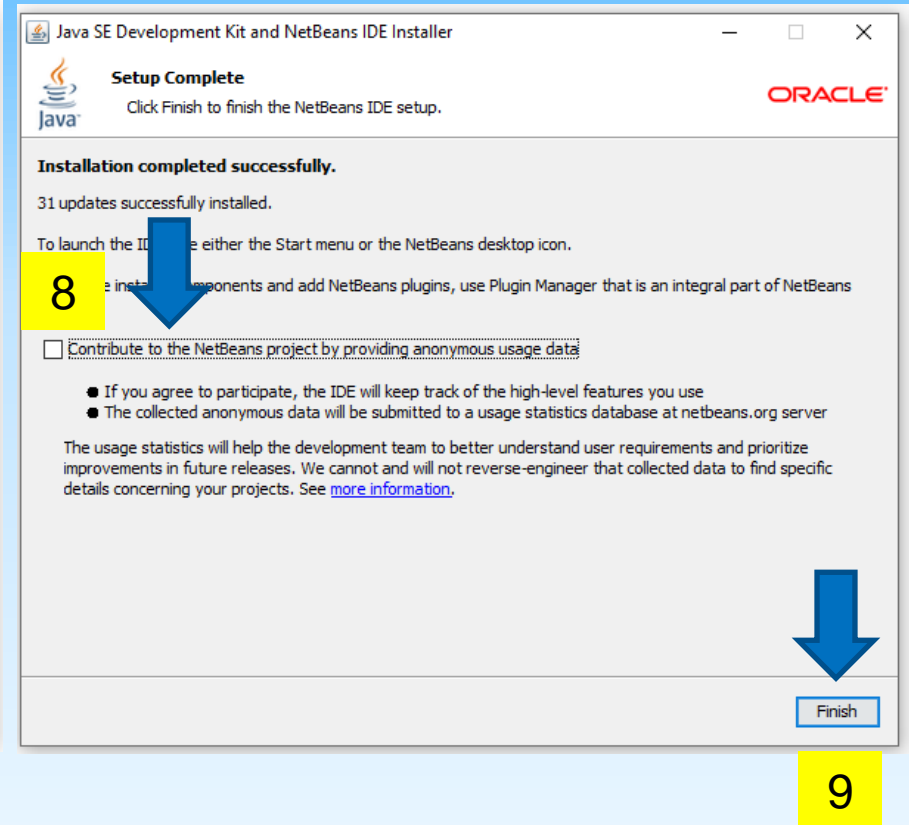
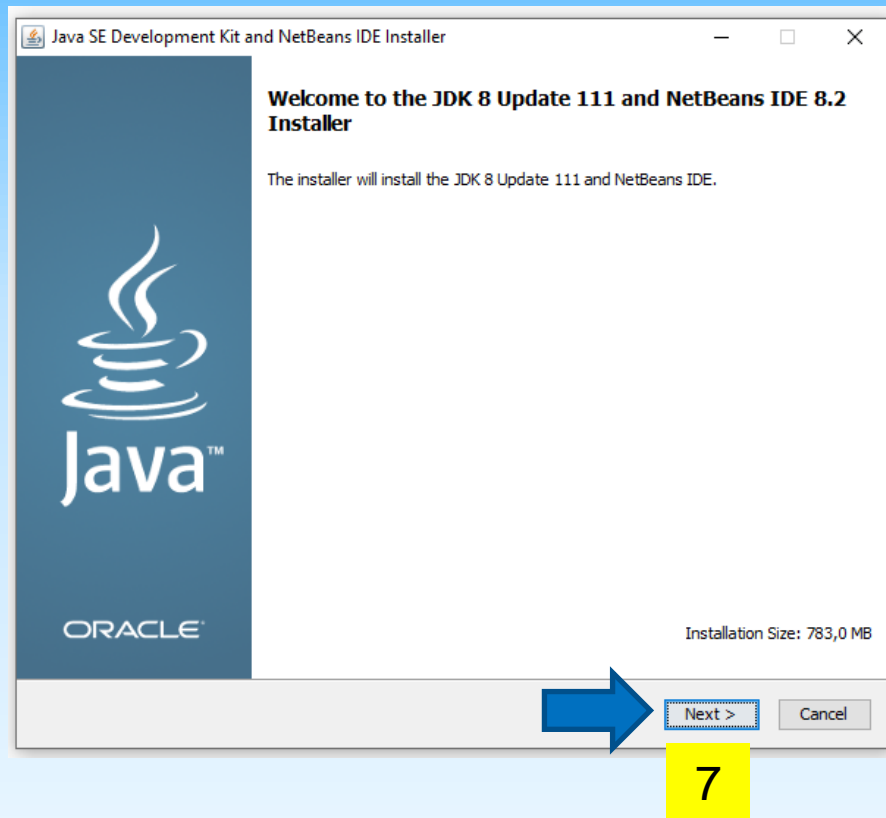


Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



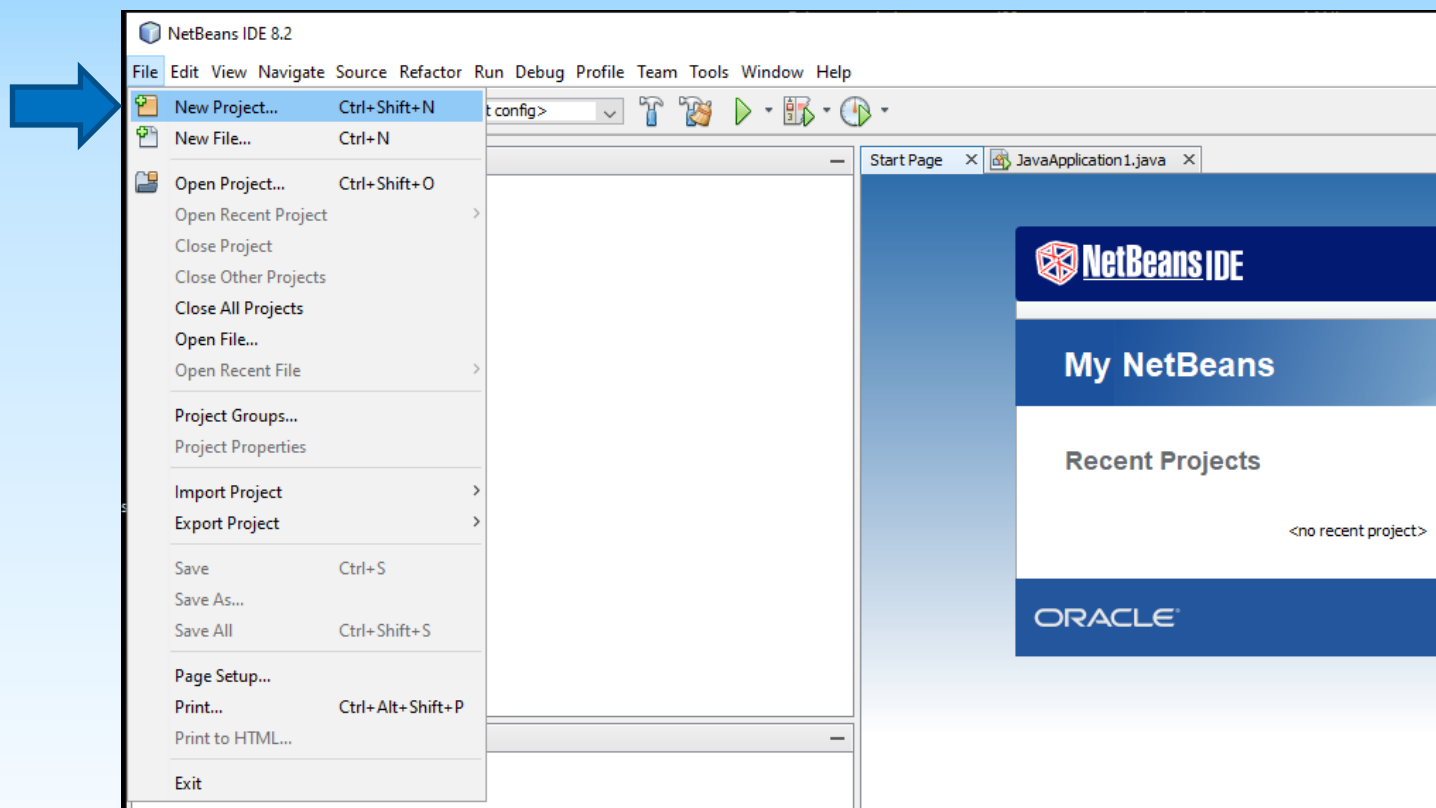
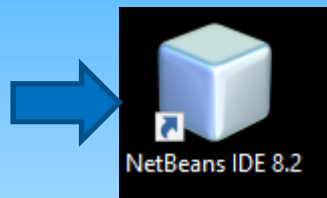
Instalacja programu NetBeans cz. 2



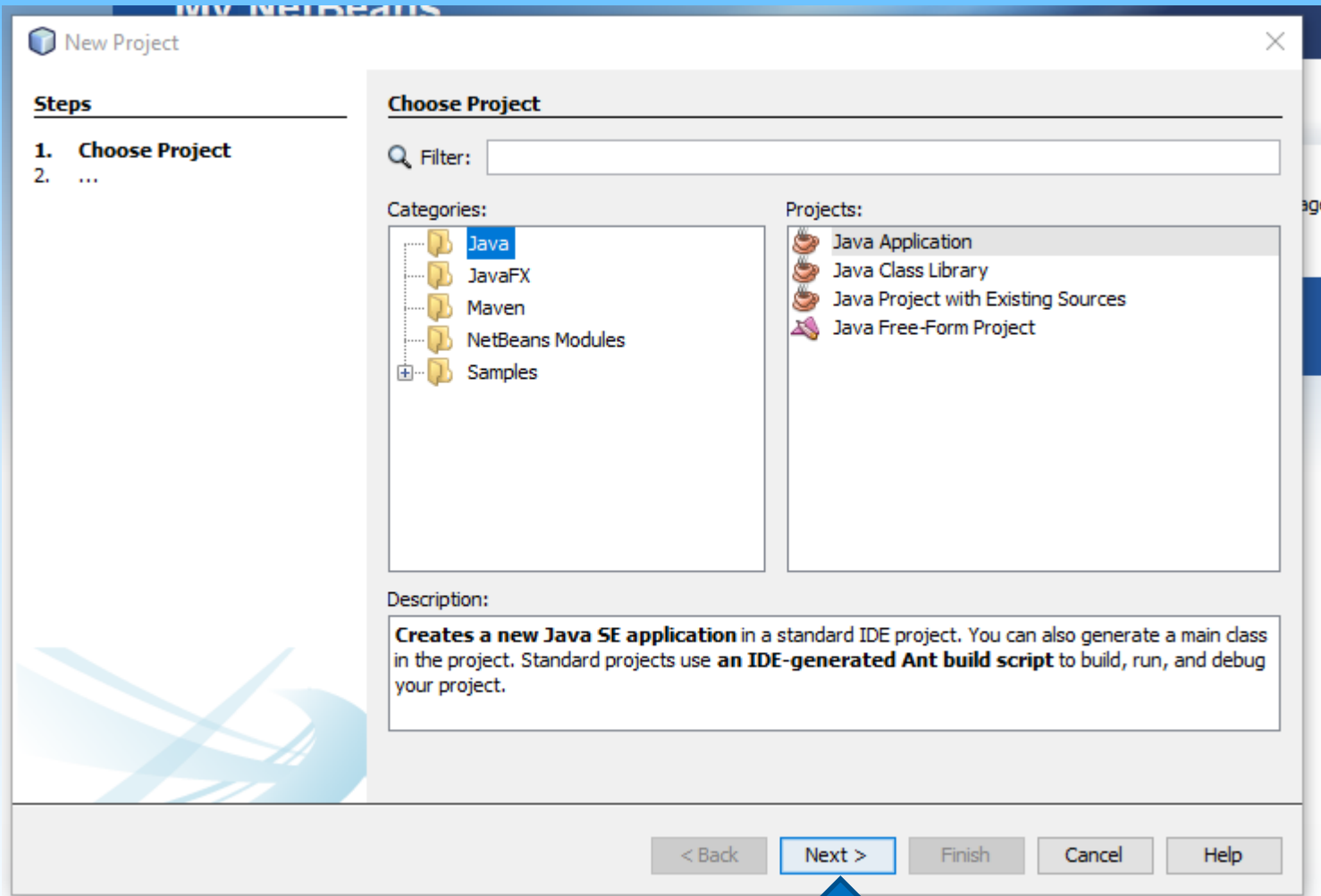
Programujemy ?



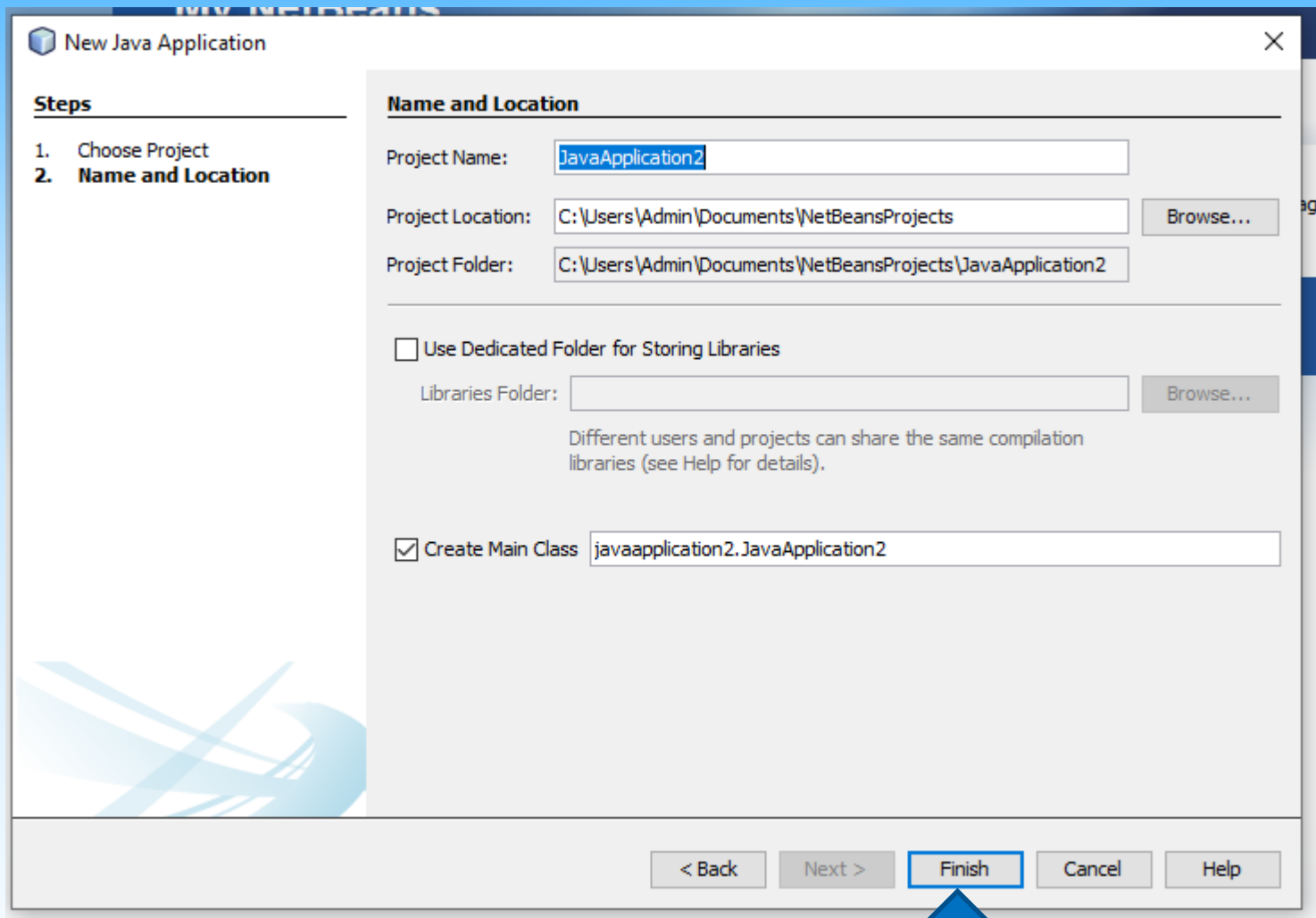
START cz.1



START cz.2



START cz.3



New Java Application

Steps

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

Name and Location

Project Name:

Project Location:

Project Folder:

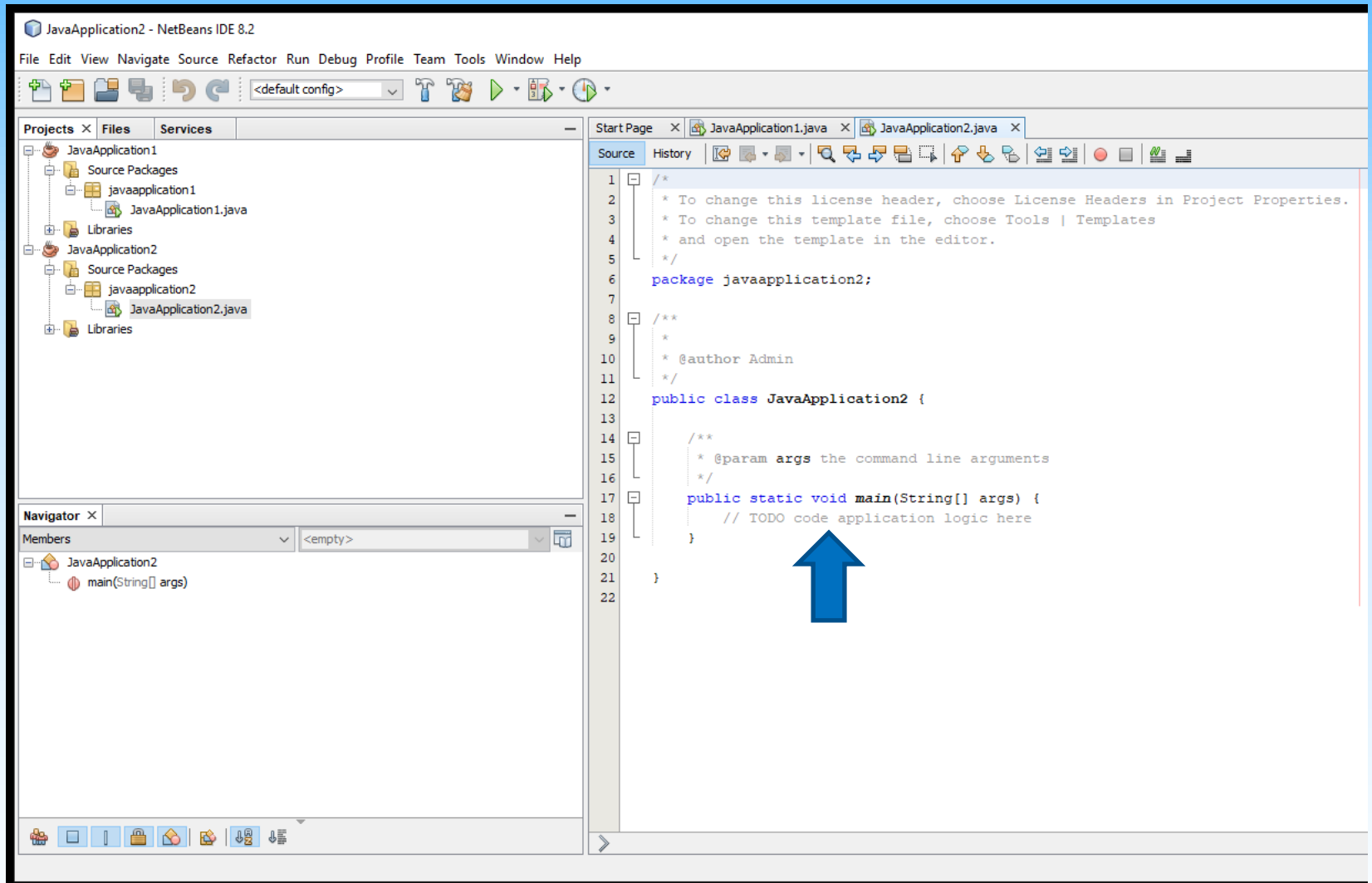
☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

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

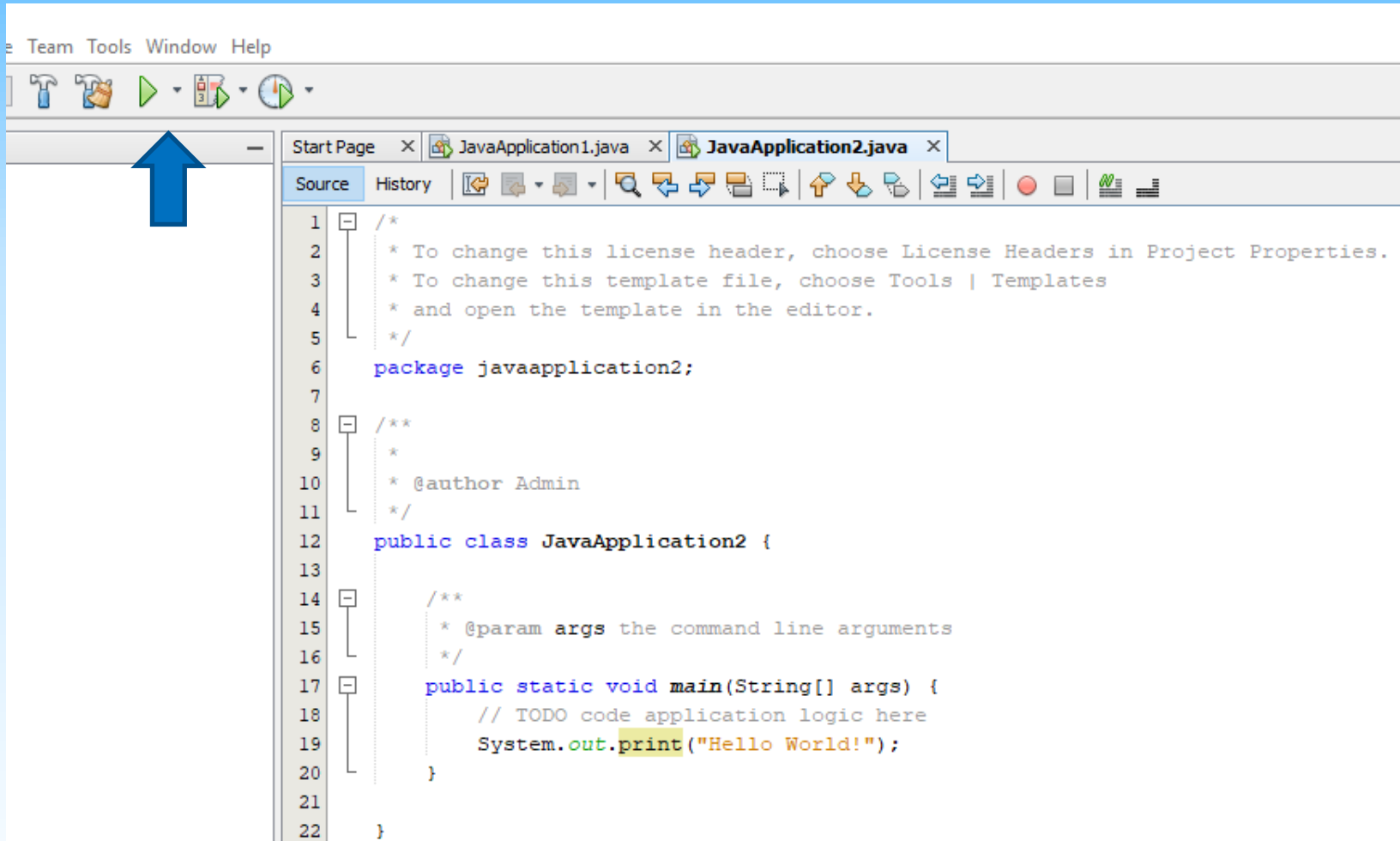
☒ Create Main Class

START cz.4



Program nr 1

Hello World!



The screenshot shows an IDE window with the following components:

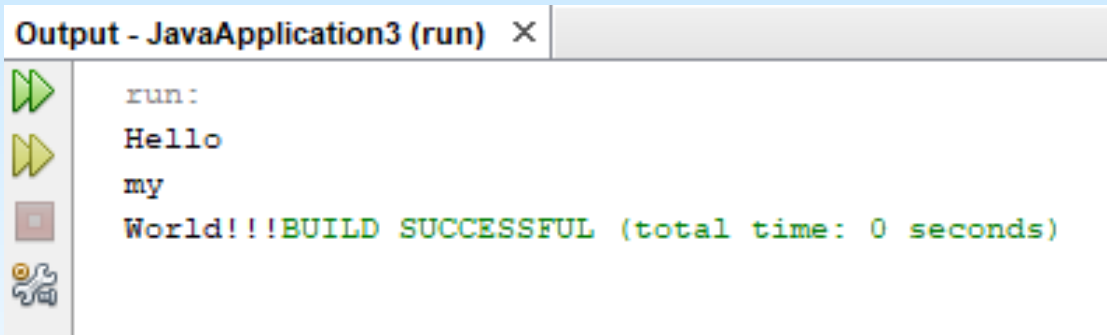
- Menu Bar:** File, Team, Tools, Window, Help.
- Toolbar:** Contains icons for Run (a green play button), which is highlighted by a blue arrow, and other development tools like a magnifying glass and a list icon.
- Tab Bar:** Shows two open files: 'JavaApplication1.java' and 'JavaApplication2.java'.
- Source Editor:** Displays the code for 'JavaApplication2.java'. The code is as follows:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package javaapplication2;
7
8   /**
9    *
10   * @author Admin
11   */
12   public class JavaApplication2 {
13
14       /**
15        * @param args the command line arguments
16        */
17       public static void main(String[] args) {
18           // TODO code application logic here
19           System.out.print("Hello World!");
20       }
21
22   }
```


Program nr 2

Przejsie do nowej linii definiujemy poleceniem \n

```
public static void main(String[] args) {  
    System.out.print("Hello\nmy\nWorld!!!");  
}
```



```
run:  
Hello  
my  
World!!!BUILD SUCCESSFUL (total time: 0 seconds)
```

Program nr 3

Proszę napisać program, który wyświetli na ekranie użytkownika:

Hello

*

Zmienne cz. 1

Liczby całkowite

typ	l. bitów	Zakres liczb
byte	8	−128 do 127
short	16	−32 768 do 32 767
int	32	−2 147 483 648 do 2 147 483 647
long	64	−9 223 372 036 854 775 808 do 9 223 372 036 854 775 807

Liczby zmiennopozycyjne

typ	l. bitów	Zakres liczb
float	8	$-3,4 \cdot 10^{38}$ do $3,4 \cdot 10^{38}$
double	16	$-1,8 \cdot 10^{308}$ do $1,8 \cdot 10^{308}$

nazwa zmiennej

```
int wiek=23;
```

typ zmiennej

przypisana wartość zmiennej

Program nr 4

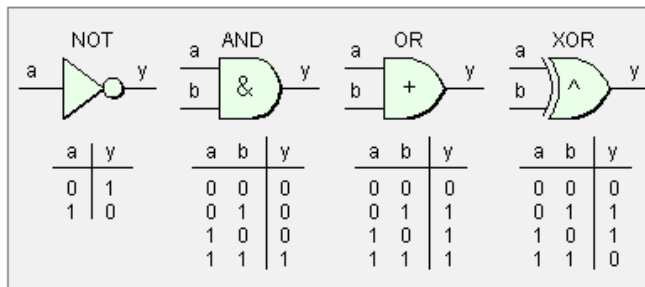
```
public static void main(String[] args) {  
    String message = "To jest mój pierwszy program";  
    System.out.println(message);  
}
```

Program nr 5 – ćwiczenie

== równe,
< mniejsze,
<= mniejsze bądź równe,
> większe,
>= większe bądź równe

&& logiczne i,
|| logiczne lub.

Operacje logiczne



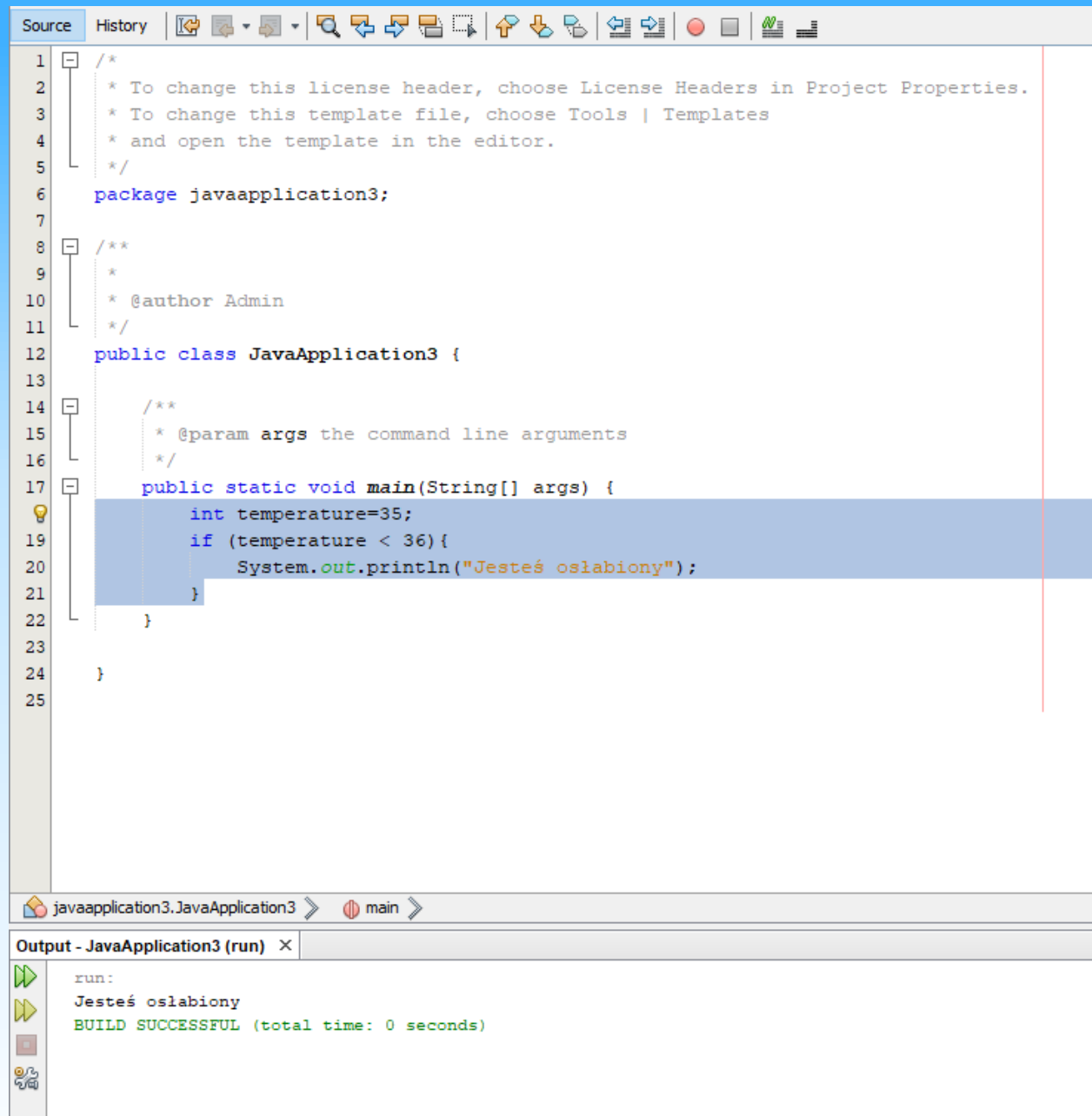
```
int x = 1;
int y = 1;
int z = 2;

x == y && z > y; // true && true => true
x <= y && z <= x; // true && false => false
x == y || z > y; // true || true => true
x <= y || z <= x; // true || false => true
```

Operacja	Wynik
prawda i prawda	prawda
prawda i fałsz	fałsz
fałsz i prawda	fałsz
fałsz i fałsz	fałsz
prawda lub prawda	prawda
prawda lub fałsz	prawda
fałsz lub prawda	prawda
fałsz lub fałsz	fałsz

Program nr 6

```
if (temperature < 36) {  
    System.out.println("Jesteś osłabiony");  
}
```



The screenshot shows an IDE window with a Java source file. The code defines a package, a class, and a main method. The main method contains a temperature check and a print statement. The code is highlighted in blue. Below the code editor, the 'Output' window shows the execution results, including the printed message and a successful build status.

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package javaapplication3;
7
8   /**
9    *
10   * @author Admin
11   */
12   public class JavaApplication3 {
13
14       /**
15        * @param args the command line arguments
16        */
17       public static void main(String[] args) {
18           int temperature=35;
19           if (temperature < 36){
20               System.out.println("Jesteś osłabiony");
21           }
22       }
23
24   }
25
```

javaapplication3.JavaApplication3 > main >

Output - JavaApplication3 (run) X

```
run:
Jesteś osłabiony
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program nr 7

```
if (temperature < 36) {  
    System.out.println("Jesteś osłabiony?");  
}  
else {  
    System.out.println("Masz 36 lub więcej stopni.");  
}
```


Program nr 8

```
int temperature = 38;
if (temperature < 36) {
    System.out.println("Jesteś osłabiony?");
}
else if (temperature < 37) {
    System.out.println("Wszystko w normie!");
}
else if (temperature < 38) {
    System.out.println("Jesteś przeziębiony?");
}
else {
    System.out.println("Masz co najmniej 38 stopni! Biegiem do lekarza!");
}
```

```
public static void main(String[] args) {
    double t=40;
    if(t<36){
        System.out.println("jestes oslabiony");
    }
    else if(t<37){
        System.out.println("jestes OK");
    }
    else if(t<42){
        System.out.println("jestes przeziębiony");
    }
}
```

Program nr 9

Proszę napisać program, który na podstawie wprowadzonego wieku przypisze nas do jednej z trzech grup wiekowych:

1. $0 - 12 \rightarrow$ dziecko
2. $13 - 18 \rightarrow$ młodzież
3. $18 > x \rightarrow$ dorośli

Program nr 10

```
int temperature = 37;
switch (temperature) {
    case 35:
        System.out.println("Jesteś osłabiony?");
        break;
    case 36:
        System.out.println("Wszystko w normie!");
        break;
    case 37:
        System.out.println("Jesteś przeziębiony?");
        break;
    case 38:
        System.out.println("Chyba jesteś chory.");
        break;
    default:
        System.out.println("Nie wiem jak się czujesz :(");
        break;
}
```

Program nr 11 – funkcja

Napisać funkcję generującą 5 gwiazdek.

```
public class JavaApplication5 {  
  
    public static void rysuj_gwiazdki(){  
        System.out.println("*****");  
    }  
  
    public static void main(String[] args) {  
        rysuj_gwiazdki();  
    }  
}
```



```
Output - JavaApplication5 (run) X  
run:  
*****  
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
public class JavaApplication5 {  
  
    public static void rysuj_gwiazdki(){  
        System.out.println("*****");  
    }  
  
    public static void main(String[] args) {  
        //rysuj_gwiazdki();  
    }  
}
```

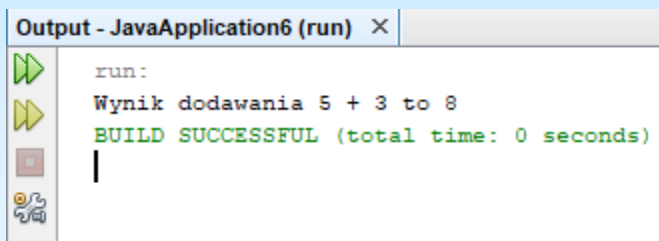


```
Output - JavaApplication5 (run) X  
run:  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

Program nr 12

Działania arytmetyczne: dodawanie

```
public static void main(String[] args) {  
    int a=5;  
    int b=3;  
    int w=a+b;  
    System.out.println("Wynik dodawania "+a+" + "+b+" to "+w);  
}
```



```
run:  
Wynik dodawania 5 + 3 to 8  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program nr 13

Działania arytmetyczne: dodawaniem, odejmowanie, mnożenie i dzielenie

```
public static void main(String[] args) {  
    int a=8;  
    int b=2;  
    int w=a+b;  
    System.out.println("Wynik dodawania "+a+" + "+b+" to "+w);  
  
    w=a-b;  
    System.out.println("Wynik odejmowania "+a+" - "+b+" to "+w);  
  
    w=a*b;  
    System.out.println("Wynik mnożenia "+a+" * "+b+" to "+w);  
  
    w=a/b;  
    System.out.println("Wynik dzielenia "+a+" / "+b+" to "+w);  
}
```



```
Output - JavaApplication6 (run) ×  
run:  
Wynik dodawania 8 + 2 to 10  
Wynik odejmowania 8 - 2 to 6  
Wynik mnożenia 8 * 2 to 16  
Wynik dzielenia 8 / 2 to 4  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program nr 14 – kalkulator (1/3)

Zadanie

Stworzyć program kalkulator, który na podstawie wybranej funkcji będzie wykonywał jedna z czterech działanie arytmetyczne:

- A. Dodawaniem – case 1
- B. Odejmowanie – case 2
- C. Mnożenie – case 3
- D. Dzielenie – case 4

Uwagi.

Użyć instrukcji switch

```
int temperature = 37;
switch (temperature) {
    case 35:
        System.out.println("Jesteś osłabiony?");
        break;
    case 36:
        System.out.println("Wszystko w normie!");
        break;
    case 37:
        System.out.println("Jesteś przeziębiony?");
        break;
    case 38:
        System.out.println("Chyba jesteś chory.");
        break;
    default:
        System.out.println("Nie wiem jak się czujesz :(");
        break;
}
```

Program nr 14 – kalkulator (2/3)

Rozwiązanie – krok 1

```
public class JavaApplication6 {  
  
    public static void działanie(int wybor, int a, int b){  
        switch(wybor){  
            case 1:  
                int wynik=a+b;  
                System.out.println("Wynik dodawania "+a+" + "+b+" to "+wynik);  
                break;  
            default:  
                System.out.println("Błąd");  
        }  
    }  
  
    public static void main(String[] args) {  
        działanie(1,2,4);  
    }  
}
```



```
Output - JavaApplication6 (run) X  
run:  
Wynik dodawania 2 + 4 to 6  
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
public class JavaApplication6 {  
  
    public static void działanie(int wybor, int a, int b){  
        switch(wybor){  
            case 1:  
                int wynik=a+b;  
                System.out.println("Wynik dodawania "+a+" + "+b+" to "+wynik);  
                break;  
            default:  
                System.out.println("Błąd");  
        }  
    }  
  
    public static void main(String[] args) {  
        działanie(5,2,4);  
    }  
}
```



```
Output - JavaApplication6 (run) X  
run:  
Błąd  
BUILD SUCCESSFUL (total time: 0 seconds)
```


Program nr 14 – kalkulator (3/3)

Rozwiązanie pełne

```
14 public static void działanie(int wybor, int a, int b){
15     switch(wybor){
16     case 1:
17         int wynik=a+b;
18         System.out.println("Wynik dodawania "+a+" + "+b+" to "+wynik);
19         break;
20     case 2:
21         wynik=a-b;
22         System.out.println("Wynik odejmowania "+a+" - "+b+" to "+wynik);
23         break;
24     case 3:
25         wynik=a*b;
26         System.out.println("Wynik mnożenia "+a+" * "+b+" to "+wynik);
27         break;
28     case 4:
29         wynik=a/b;
30         System.out.println("Wynik dzielenia "+a+" / "+b+" to "+wynik);
31         break;
32     default:
33         System.out.println("Błąd");
34     }
35 }
36 public static void main(String[] args) {
37     działanie(2,2,4);
38 }
39
40 }
41
```

javaapplication6.JavaApplication6 > main >

Output - JavaApplication6 (run) ×

```
run:
Wynik odejmowania 2 - 4 to -2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Zmienne

1. Liczby całkowite

typ	I. bitów	Zakres liczb
byte	8	-128 do 127
short	16	-32 768 do 32 767
int	32	-2 147 483 648 do 2 147 483 647
long	64	-9 223 372 036 854 775 808 do 9 223 372 036 854 775 807

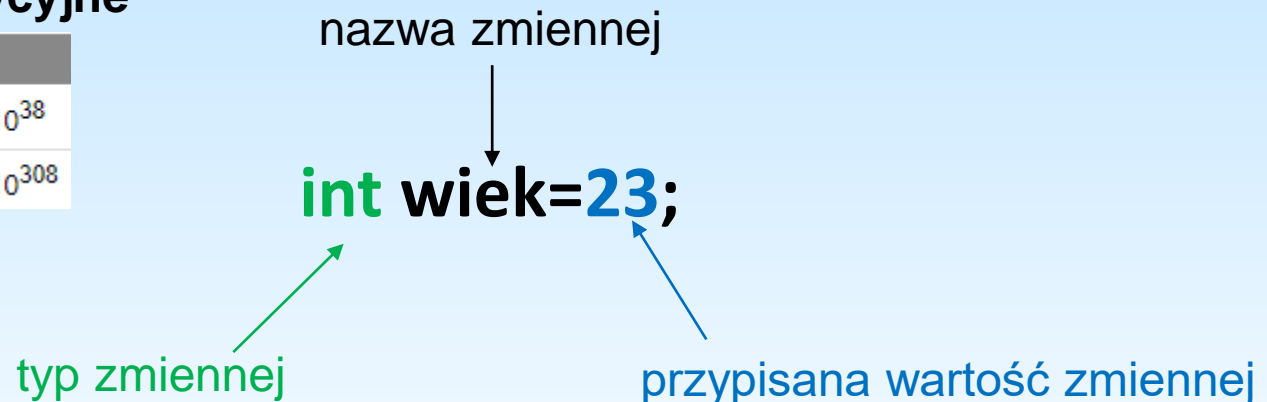
2. Liczby zmiennopozycyjne

typ	I. bitów	Zakres liczb
float	8	$-3,4 \cdot 10^{38}$ do $3,4 \cdot 10^{38}$
double	16	$-1,8 \cdot 10^{308}$ do $1,8 \cdot 10^{308}$

3. Boolean:

- 0 (false – fałsz)
- 1 (true – prawda)

4. String – ciąg znaków



Tablice

`int [] array =`

Wartość:	15	19	17	11	16
Index:	0	1	2	3	4

20



W informatyce liczenie zawsze zaczynamy od zera dlatego najmniejszym indeksem w tablicy jest zero.

- Uzyskanie wartości danego indeksu tablicy: `array[4] → 16`
- Modyfikacja wartości danego indeksu tablicy: `array[4] = 20`
- Metoda `array.length` sprawdzi nam długość tablicy: 5
Długość tablicy jest zawsze o 1 większa od ostatniego numeru indeksu tablicy.

Tablice – tworzenie. Sposób 1

```
public static void main(String[] args) {  
    // I Sposób  
    int[] tab=new int[5]; // tablica przyznaje jeden rozmiar, który zostaje już do końca  
    System.out.println(tab[0]); // tablica po stworzeniu jest wypełniona elementami domyślnymi, dla int są to zera  
    tab[0]=2;  
    tab[1]=13;  
    tab[2]=5;  
    tab[3]=8;  
    tab[4]=12;  
    System.out.println(tab[0]);  
    System.out.println(tab[1]);  
    System.out.println(tab[2]);  
    System.out.println(tab[3]);  
    System.out.println(tab[4]);  
    System.out.println(tab.length);  
    //System.out.println(tab[5]);  
}
```



```
run:  
0  
2  
13  
5  
8  
12  
5  
BUILD SUCCESSFUL (total time: 0 seconds)
```

*Obsługa błędu → tab [5]

```
int i=4;  
if (i>=0 && i < tab.length)  
    System.out.println(tab[i])
```

Tablice – tworzenie. Sposób 2

```
public static void main(String[] args) {  
    // II Sposób  
    String[] kolory = {"czerwony", "zielony", "niebieski"};  
    System.out.println(kolory[1]);  
    System.out.println("Długość" + kolory.length);  
    kolory[1]="czarny";  
    System.out.println(kolory[1]);  
}
```



```
run:  
zielony  
Długość3  
czarny  
BUILD SUCCESSFUL (total time: 0 seconds)
```


Wczytywanie danych

```
package in_out;

import java.util.Scanner;

public class In_out {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Ile masz lat?");
        String wiek=scan.nextLine();
        System.out.println("A więc masz "+wiek);
    }
}
```



```
run:
Ile masz lat?
25
A więc masz 25
BUILD SUCCESSFUL (total time: 2 seconds)
```

nextInt() – odczytuje kolejną liczbę całkowitą
nextDouble() – odczytuje kolejną liczbę zmiennoprzecinkową
nextBoolean() – odczytuje kolejną liczbę typu boolean

JAVA oferuje gotowe klasy, które w prosty sposób umożliwiają wprowadzenie danych przez użytkownika „z klawiatury”.

Klasa Scanner znajduje się w pakiecie java.util, dlatego aby skorzystać z niej należy ją zaimportować → import

Następnie stworzyć obiekt typu Scanner

Metoda nextLine() wstrzymuje działanie programu do momentu wpisania przez użytkownika danych „z klawiatury”, zatwierdzając enterem

- do wyświetlania danych w konsoli posługujemy się strumieniem, przypisanym do obiektu System.out
- do wczytywania danych z klawiatury skorzystamy ze strumienia System.in

Program nr 15 – równanie kwadratowe (1/3)

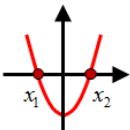
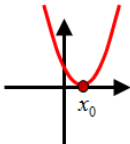
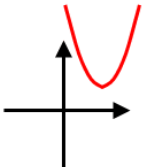
$$x^2 + 3x - 4 = 0$$

$$a=1 \quad b=3 \quad c=-4$$

$$\Delta = b^2 - 4 \cdot a \cdot c = 3^2 - 4 \cdot 1 \cdot (-4) = 9 + 16 = 25 \quad \sqrt{\Delta} = \sqrt{25} = 5$$

$$x_1 = \frac{-b + \sqrt{\Delta}}{2 \cdot a} = \frac{-3 + 5}{2 \cdot 1} = \frac{2}{2} = 1$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2 \cdot a} = \frac{-3 - 5}{2 \cdot 1} =$$

$\Delta > 0$	$\Delta = 0$	$\Delta < 0$
$x_1 = \frac{-b - \sqrt{\Delta}}{2 \cdot a}$ $x_2 = \frac{-b + \sqrt{\Delta}}{2 \cdot a}$	$x_0 = \frac{-b}{2a}$	Brak pierwiastków
		

Program nr 15 – równanie kwadratowe (2/3)

```
import java.util.Scanner;

public class Rownanie_kwadratowe {

    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Wprowadź a:");
        int a=scan.nextInt();
        System.out.println("Wprowadź b:");
        int b=scan.nextInt();
        System.out.println("Wprowadź c:");
        int c=scan.nextInt();
        System.out.println("a="+a+" b="+b+" c="+c);
    }
}
```


Program nr 15 – równanie kwadratowe (3/3)

```
package matematyka;

import static java.lang.Math.*;

public class Matematyka {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        double first = 9.0;
        int second = 3;
        double sqrt = sqrt(first); //pierwiastek kwadratowy
        double power = pow(first, second); //9 do potęgi 3

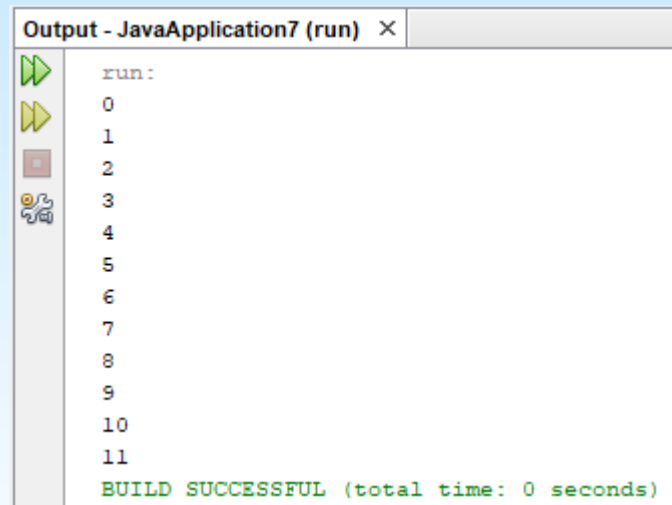
        System.out.println("Pierwiastek z " + first + " wynosi: " + sqrt);
        System.out.println("Liczba " + first + " podniesiona do potęgi " + second + " to " + power);
    }
}
```

Pętla for

Pętla for

```
for(wyrażenie początkowe ; warunek ; modyfikator_licznika){  
    instrukcje do wykonania  
}
```

```
public static void main(String[] args) {  
    for (int i=0;i<12;i++){  
        System.out.println(i);  
    }  
}
```



```
Output - JavaApplication7 (run) ×  
run:  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pętla for – kombinacja 3 cyfrowa

```
public static void main(String[] args) {  
  
    int i,j,k;  
    for (i = 0; i < 37; i++) {  
        for (j = 0; j < 37; j++) {  
            for (k = 0; k < 37; k++) {  
                System.out.println(i+" "+j+" "+k+"\n");  
            }  
        }  
    }  
}
```

Output - JavaApplication16 (run) ×

▶▶	0 0 34
▶▶	0 0 35
■	0 0 36
🔍	0 1 0
	0 1 1
	0 1 2

Program nr 16 – pętla for i funkcja (1/2)

Zadanie

Napisz funkcję rysującą prostokąt o zadanych wymiarach, który składa się z samych gwiazdek.

Na przykład po wywołaniu tej funkcji z parametrami 8 i 3 powinniśmy dostać:

Program nr 16 – pętla for i funkcja (2/2)

Rozwiązanie

```
12 public class JavaApplication7 {
13
14     public static void rysuj_prostokat(int a, int b) {
15         for (int i=0;i<b;i++){
16             for (int k=0;k<a;k++){
17                 System.out.print("*");
18             }
19             System.out.println("");
20         }
21     }
22
23
24     public static void main(String[] args) {
25         rysuj_prostokat(8, 3);
26     }
27
28 }
29
```

javaapplication7.JavaApplication7 > main >

Output - JavaApplication7 (run) ×

run:

BUILD SUCCESSFUL (total time: 0 seconds)

Pętla while

Pętla while

```
while(warunek){  
    instrukcje do wykonania  
}
```

```
public class JavaApplication19 {  
  
    public static void main(String[] args) {  
        int licznik = 0;  
        while(licznik<5){  
            System.out.println("To jest petla");  
            licznik++;  
        }  
        System.out.println("Koniec petli");  
    }  
}
```

```
Output - JavaApplication19 (run) x  
run:  
To jest petla  
To jest petla  
To jest petla  
To jest petla  
To jest petla  
Koniec petli  
BUILD SUCCESSFUL (total time: 0 seconds)
```


Pętla do while

Pętla do while

```
do{  
    instrukcje do wykonania  
}  
while(warunek);
```

```
public static void main(String[] args) {  
    int licznik = 0;  
    do{  
        System.out.println("To jest petla");  
        licznik++;  
    }  
    while(licznik<0);  
    System.out.println("Koniec pętli");  
}
```

Output - JavaApplication19 (run) ×

```
run:  
To jest petla  
Koniec pętli  
BUILD SUCCESSFUL (total time: 0 seconds)
```

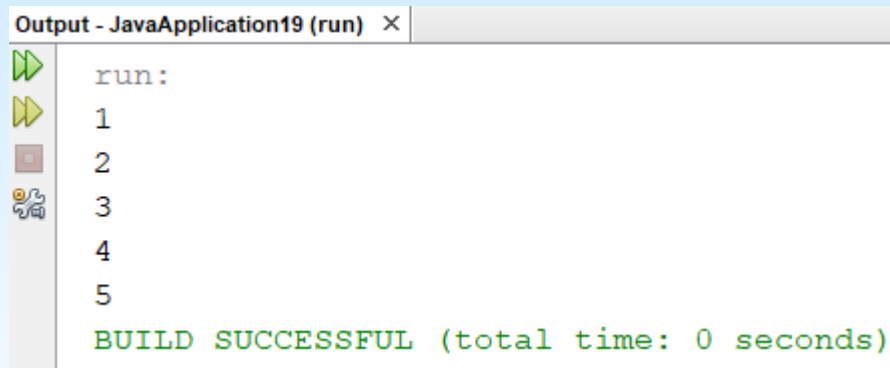
Pętla for each

Pętla for each

→ dla każdego elementu z tablicy

```
for(Typ_Objektu nazwa_objektu : nazwa_tablicy){ ... }
```

```
public static void main(String[] args) {  
    int[] tablica = new int[5]; //wypełnienie tablicy  
    for (int i = 0; i < 5; i++) {  
        tablica[i] = i + 1;  
    }  
    for (int x : tablica) { //wyświetlenie przy użyciu pętli for each  
        System.out.println(x);  
    }  
}
```



Output - JavaApplication19 (run) ×

run:
1
2
3
4
5
BUILD SUCCESSFUL (total time: 0 seconds)

Tablice – pętla for each

```
public static void main(String[] args) {  
    String[] kolory = {"czerwony", "zielony", "niebieski", "czarny", "biały"};  
  
    System.out.println("For:");  
    for (int i=0; i<kolory.length ;i++)  
        System.out.println (kolory[i]);  
  
    System.out.println("For each:"); // pętla obiektowa dla każdego elementu  
    for (String item : kolory){ // (zmienna pomocnicza typu jak tablica : tablica)  
        System.out.println(item);  
    }  
}
```



```
run:  
For:  
czerwony  
zielony  
niebieski  
czarny  
biały  
For each:  
czerwony  
zielony  
niebieski  
czarny  
biały  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program nr 17

Napisać program za pomocą jednej z poznanych pętli który wyświetli na ekranie użytkownika wszystkie numery dni wprowadzonego miesiące.

Podpowieź.

Zastosować instrukcje warunkową



Uniwersytet
Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



Dostęp do tej klasy z poziomu całego projektu

```
public class JavaApplication19 {
```

```
1 {  
  2 public class Test{  
    public static void main(String[] args){  
      System.out.println("Witaj Javo");  
    }  
  }  
}
```

1 – deklaracja klasy o nazwie **T**est, ciało klasy

2 – deklaracja funkcji o nazwie **m**ain, ciało metody

- Pierwsza litera nazwy klasy przyjęło się zapisywać wielką literą
- Nazwy metod piszemy małymi literami

Dzięki temu wiemy co jest klasą a co metodą.

Test.java → kompilator → Test.class → uruchomienie JVM → wynik na konsoli

Plik źródłowy
(Nasz kod – język java
zrozumiały dla nas)

Plik pośredni, System operacyjny
plik wykonywalny
(kod pośredni, kod bajtowy,
układ 0101100)



Uniwersytet
Ekonomiczny
we Wrocławiu



Fundusze
Europejskie
Wiedza Edukacja Rozwój



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz Społeczny



Write once, run anywhere

Nawiasy klamrowe – 2 konwencje zapisu

```
public class Hello{  
    public static void main(String[] args){  
        System.out.print("Hello World");  
    }  
}
```

```
public class Hello  
{  
    public static void main(String[] args)  
    {  
        System.out.print("Hello World");  
    }  
}
```


Tutorial

- <https://www.samouczekprogramisty.pl/kurs-programowania-java>
- <https://javastart.pl/baza-wiedzy>
- <https://www.javappa.com/kurs-java>
- <http://programowaniejava.pl>

Źródła

- <https://matfiz24.pl/funkcja-kwadratowa/rownanie-kwadratowe>

Dziękuję za uwagę!