

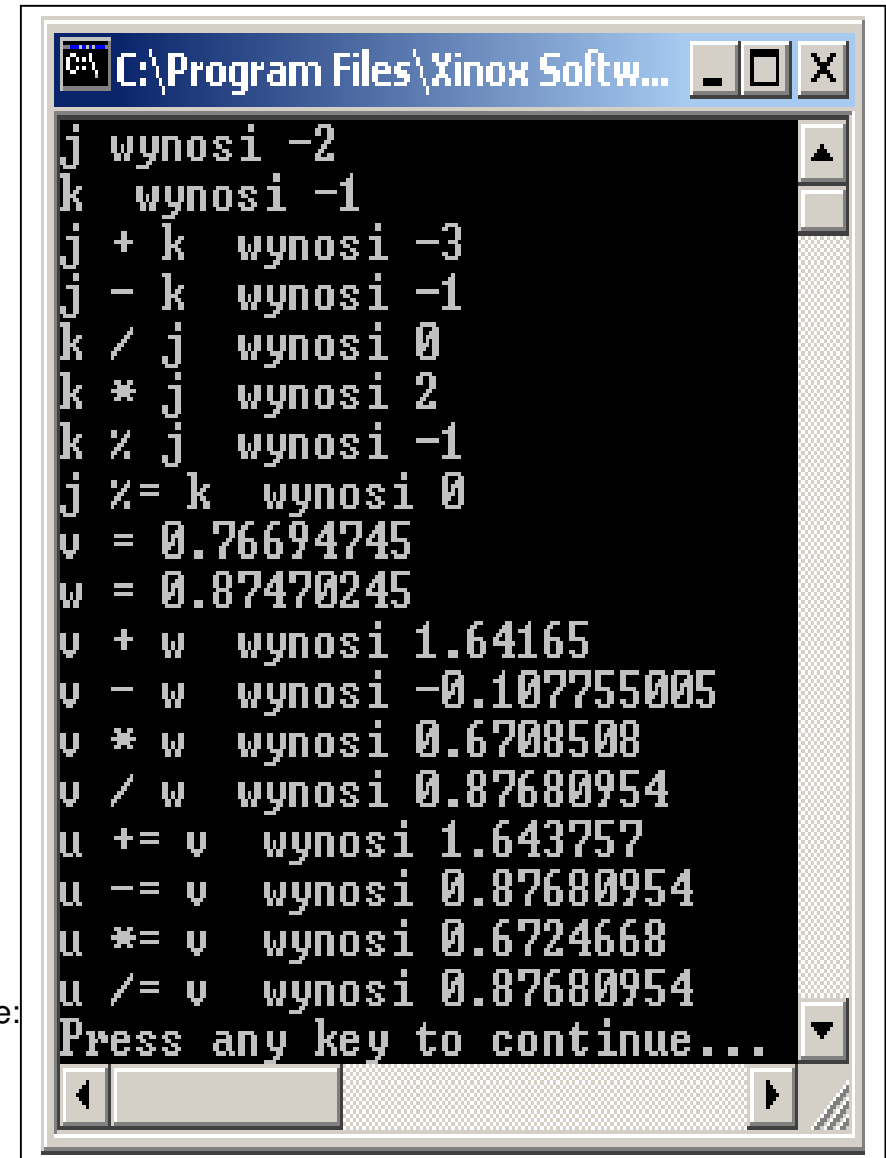
PODSTAWOWE ELEMENTY JEZYKA JAVA – WYRAŻENIA, OPERATORY, INSTRUKCJE

1. Operatory arytmetyczne +, -, /, *, %

Przykład 1 programu z interfejsem konsolowym

```
import java.util.*;

public class Lab2_1
{ public static void main(String[] args)
  { // Tworzy generator liczb losowych, zainicjowany czasem systemowym
    Random rand = new Random();
    int i, j, k; String s;
    // '%' ogranicza wartość do 9: losowanie kolejnej wartości całkowitej
    j = rand.nextInt() % 10; k = rand.nextInt() % 10;
    s = "j wynosi " + j; System.out.println(s);
    s = "k wynosi " + k; System.out.println(s);
    i = j + k; s = "j + k wynosi " + i; System.out.println(s);
    i = j - k; s = "j - k wynosi " + i; System.out.println(s);
    i = k / j; s = "k / j wynosi " + i; System.out.println(s);
    i = k * j; s = "k * j wynosi " + i; System.out.println(s);
    i = k % j; s = "k % j wynosi " + i; System.out.println(s);
    j %= k; s = "j %= k wynosi " + j; System.out.println(s);
    // Operacje na argumentach zmiennoprzecinkowych u,v,w
    float u, v, w; //losowanie kolejnej wartości rzeczywistej
    v = rand.nextFloat(); w = rand.nextFloat();
    s = "v = " + v; System.out.println(s);
    s = "w = " + w; System.out.println(s);
    u = v + w; s = "v + w wynosi " + u; System.out.println(s);
    u = v - w; s = "v - w wynosi " + u; System.out.println(s);
    u = v * w; s = "v * w wynosi " + u; System.out.println(s);
    u = v / w; s = "v / w wynosi " + u; System.out.println(s);
    // następne wyrażenia są realizowane dla char, byte, short, int, long, and double:
    u += v; s = "u += v wynosi " + u; System.out.println(s);
    u -= v; s = "u -= v wynosi " + u; System.out.println(s);
    u *= v; s = "u *= v wynosi " + u; System.out.println(s);
    u /= v; s = "u /= v wynosi " + u; System.out.println(s); } }
```



```
j wynosi -2
k wynosi -1
j + k wynosi -3
j - k wynosi -1
k / j wynosi 0
k * j wynosi 2
k % j wynosi -1
j %= k wynosi 0
v = 0.76694745
w = 0.87470245
v + w wynosi 1.64165
v - w wynosi -0.107755005
v * w wynosi 0.6708508
v / w wynosi 0.87680954
u += v wynosi 1.643757
u -= v wynosi 0.87680954
u *= v wynosi 0.6724668
u /= v wynosi 0.87680954
Press any key to continue...
```

Przykład 1 programu z graficznym interfejsem użytkownika

```
import javax.swing.*;
import java.util.*;

public class Lab2_2
{
    public static void main(String args[])
    {
        //definicja zmiennych całkowitych i, j, k oraz łańcucha s
        int i, j, k; String s;

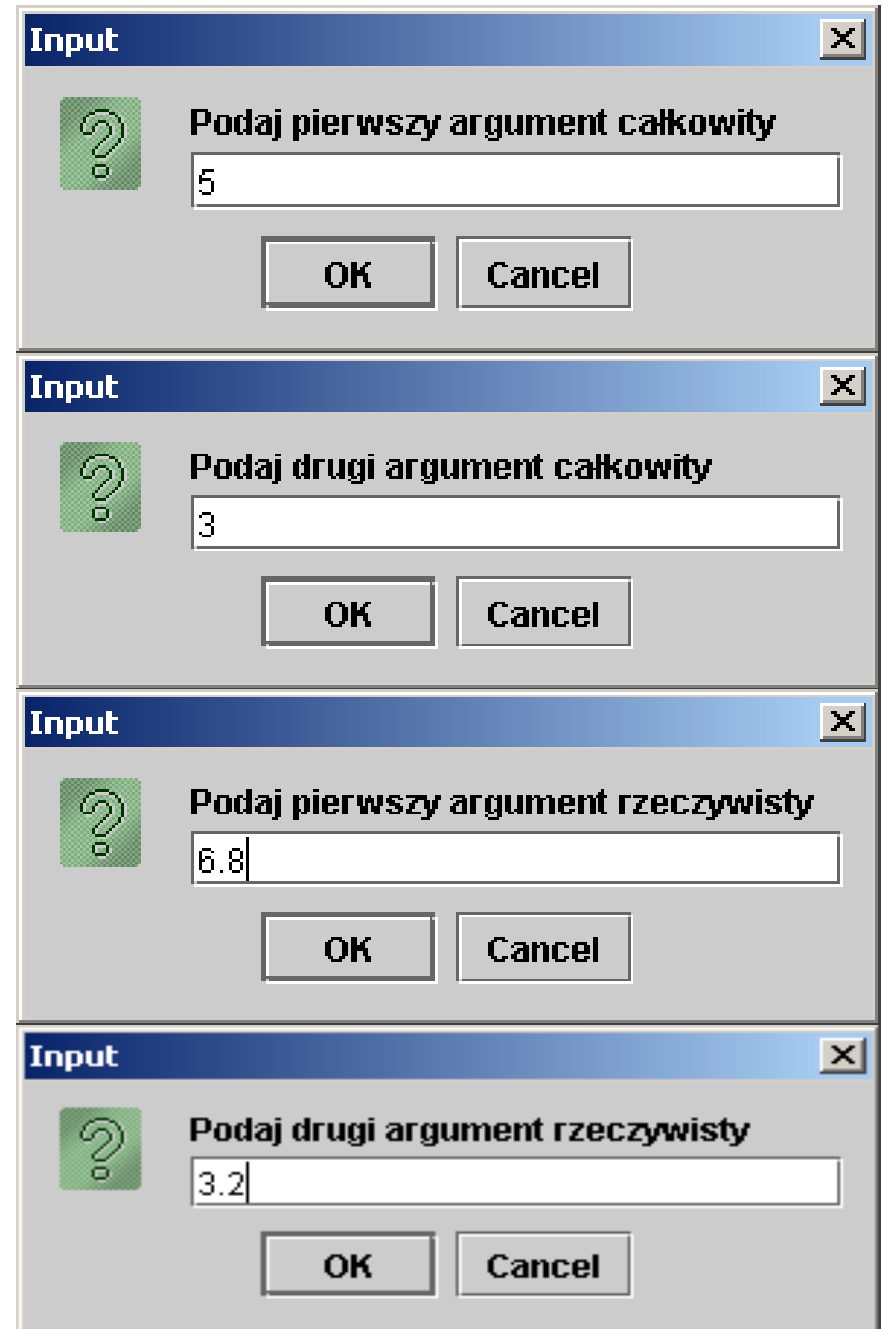
        // pobranie z okienka dialogowego łańcucha 5
        s = JOptionPane.showInputDialog(null,
            "Podaj pierwszy argument całkowity");
        //zamiana łańcucha 5 na liczbę 5
        j = Integer.parseInt(s);

        // pobranie z okienka dialogowego łańcucha 3
        s = JOptionPane.showInputDialog(null,
            "Podaj drugi argument całkowity");
        //zamiana łańcucha 3 na liczbę 3
        k = Integer.parseInt(s);

        //definicja zmiennych rzeczywistych u, v w
        float u, v, w;

        // pobranie z okienka dialogowego łańcucha 6.8
        s=JOptionPane.showInputDialog(null,
            "Podaj pierwszy argument rzeczywisty");
        //zamiana łańcucha 6.8 na liczbę
        v = Float.parseFloat(s);

        // pobranie z okienka dialogowego łańcucha 3.2
        s = JOptionPane.showInputDialog(null,
            "Podaj drugi argument rzeczywisty");
        //zamiana łańcucha 3.2 na liczbę
        w = Float.parseFloat(s);
    }
}
```



```

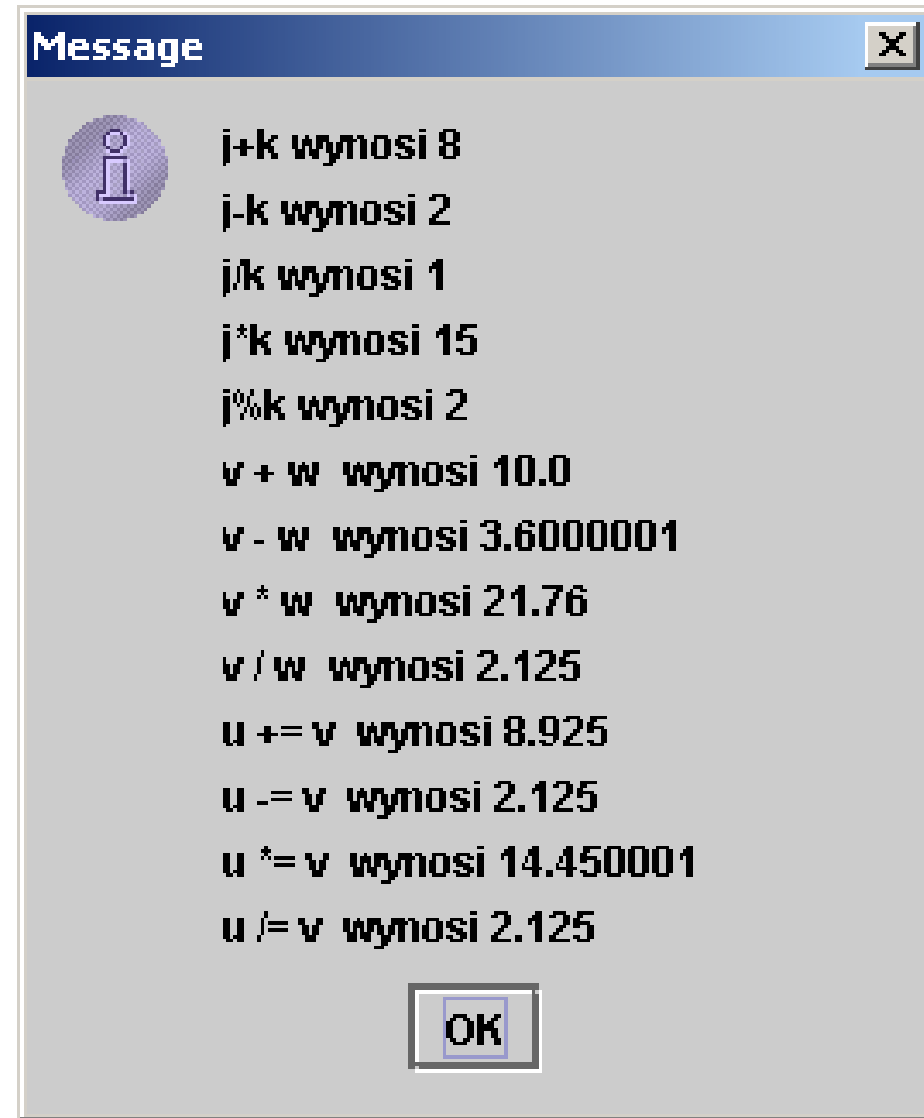
i = j+k;      s="j+k wynosi " + i + "\n";
// dodanie do łańcucha s nowego łańcucha s+= czyli s=s+
i = j - k;    s+="j-k wynosi " + i + "\n";
i = j / k;    s+="j/k wynosi " + i + "\n";
i = j * k;    s+="j*k wynosi " + i + "\n";
i = j % k;    s+="j%k wynosi " + i + "\n";

// Operacje na argumentach zmienneoprzecinkowych
u = v + w;    s += "v + w wynosi " + u + "\n";
u = v - w;    s += "v - w wynosi " + u + "\n";
u = v * w;    s += "v * w wynosi " + u + "\n";
u = v / w;    s += "v / w wynosi " + u + "\n";

// następane wyrażenia są realizowane dla
// char, byte, short, int, long i double:
u += v; s += "u += v wynosi " + u + "\n";
u -= v; s += "u -= v wynosi " + u + "\n";
u *= v; s += "u *= v wynosi " + u + "\n";
u /= v; s += "u /= v wynosi " + u + "\n";

//wyświetlenie łańcucha s
JOptionPane.showMessageDialog(null,s);
System.exit(0);
}
}

```



2. Operatory jednoargumentowe +, -. Operatory inkrementacji przedrostkowej i przyrostkowej

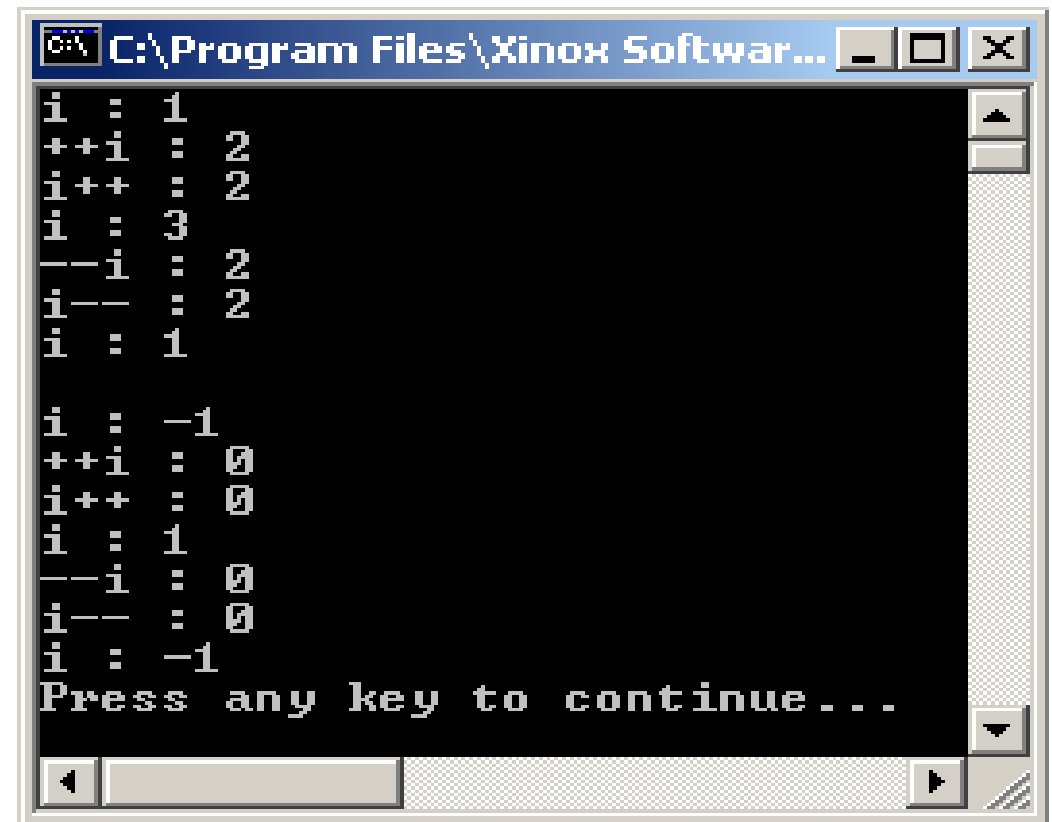
Operatory jednoargumentowe – i plus służą do określenia wartości dodatniej lub ujemnej.

Operatory inkrementacji i dekrementacji przedrostkowej np. ++i lub --i wykonują się najpierw, zanim wykona się wyrażenie, w którym użyto te operatory.

Operatory inkrementacji i dekrementacji przyrostkowej np. i++ lub i-- wykonują się po wykonaniu wyrażenia, w którym użyto te operatory.

Przykład 2 z interfejsem konsolowym

```
public class Lab2_3
{
    public static void main(String[] args)
    { int i = +1;
      System.out.println("i : " + i);
      System.out.println("++i : " + ++i); // Pre-increment
      System.out.println("i++ : " + i++); // Post-increment
      System.out.println("i : " + i);
      System.out.println("--i : " + --i); // Pre-decrement
      System.out.println("i-- : " + i--); // Post-decrement
      System.out.println("i : " + i);
      i = -1;
      System.out.println("\ni : " + i);
      System.out.println("++i : " + ++i); // Pre-increment
      System.out.println("i++ : " + i++); // Post-increment
      System.out.println("i : " + i);
      System.out.println("--i : " + --i); // Pre-decrement
      System.out.println("i-- : " + i--); // Post-decrement
      System.out.println("i : " + i);
    }
}
```

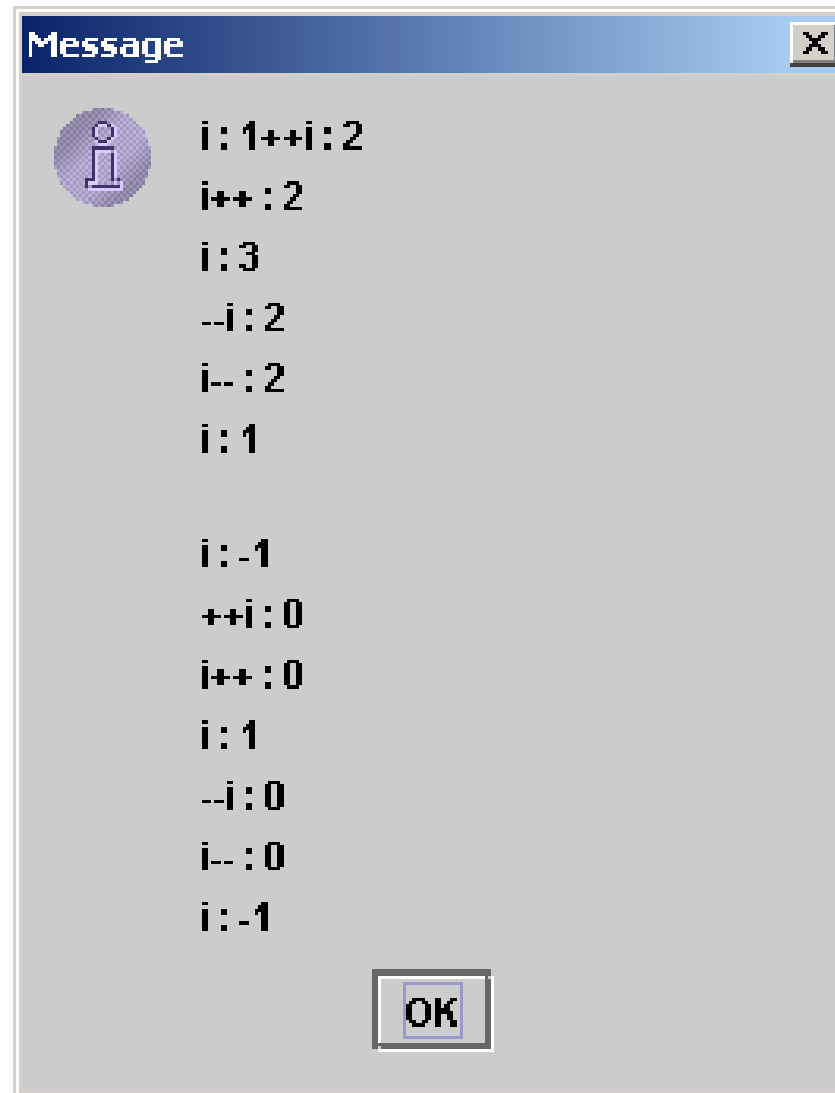


```
C:\Program Files\Xinox Softwar...
i : 1
++i : 2
i++ : 2
i : 3
--i : 2
i-- : 2
i : 1

i : -1
++i : 0
i++ : 0
i : 1
--i : 0
i-- : 0
i : -1
Press any key to continue...
```

Przykład 2 z interfejsem graficznym

```
import javax.swing.*;
import java.util.*;
public class Lab2_4
{
    public static void main(String[] args)
    {
        int i = +1; String s;
        s="i : " + i;
        s+="++i : " + ++i +"\n"; // Pre-increment
        s+="i++ : " + i++ +"\n"; // Post-increment
        s+="i : " + i +"\n";
        s+="--i : " + --i +"\n"; // Pre-decrement
        s+="i-- : " + i-- +"\n"; // Post-decrement
        s+="i : " + i +"\n";
        i = -1;
        s+="\ni : " + i +"\n";
        s+="++i : " + ++i +"\n"; // Pre-increment
        s+="i++ : " + i++ +"\n"; // Post-increment
        s+="i : " + i +"\n";
        s+="--i : " + --i +"\n"; // Pre-decrement
        s+="i-- : " + i-- +"\n"; // Post-decrement
        s+="i : " + i +"\n";
        JOptionPane.showMessageDialog(null,s);
        System.exit(0);
    }
}
```



3. Operatory relacyjne <>, >, <, >=, <=, ==, i logiczne AND (&&), OR (||) and NOT (!) - wynik działania operatorów jest równy wartości false lub true.

Przykład 3 z interfejsem konsolowym

```
import java.util.*;
```

```
public class Lab2_5 {  
    public static void main(String[] args)  
    { Random rand = new Random();  
      int i = rand.nextInt() % 100;  
      int j = rand.nextInt() % 100;  
      String s;  
      s = "i = " + i + "\n";  
      s += "j = " + j + "\n";  
      s += "i > j is " + (i > j) + "\n";  
      s += "i < j is " + (i < j) + "\n";  
      s += "i >= j is " + (i >= j) + "\n";  
      s += "i <= j is " + (i <= j) + "\n";  
      s += "i == j is " + (i == j) + "\n";  
      s += "i != j is " + (i != j) + "\n";  
  
      // Wartość typu int nie jest wartością logiczną  
      // w Javie  
      //! System.out.println(s) ("i && j is " + (i && j));  
      //! System.out.println ("i || j is " + (i || j));  
      //! System.out.println ("!i is " + !i);  
  
      s += "(i < 10) && (j < 10) is " + ((i < 10) && (j < 10)) + "\n";  
      s += "(i < 10) || (j < 10) is " + ((i < 10) || (j < 10)) + "\n";  
      System.out.println(s);  
    }  
}
```

```
i = -77  
j = -52  
i > j is false  
i < j is true  
i >= j is false  
i <= j is true  
i == j is false  
i != j is true  
(i < 10) && (j < 10) is true  
(i < 10) || (j < 10) is true  
  
Press any key to continue...
```

Przy wyświetlaniu wartość typu logicznego jest równa true lub false.

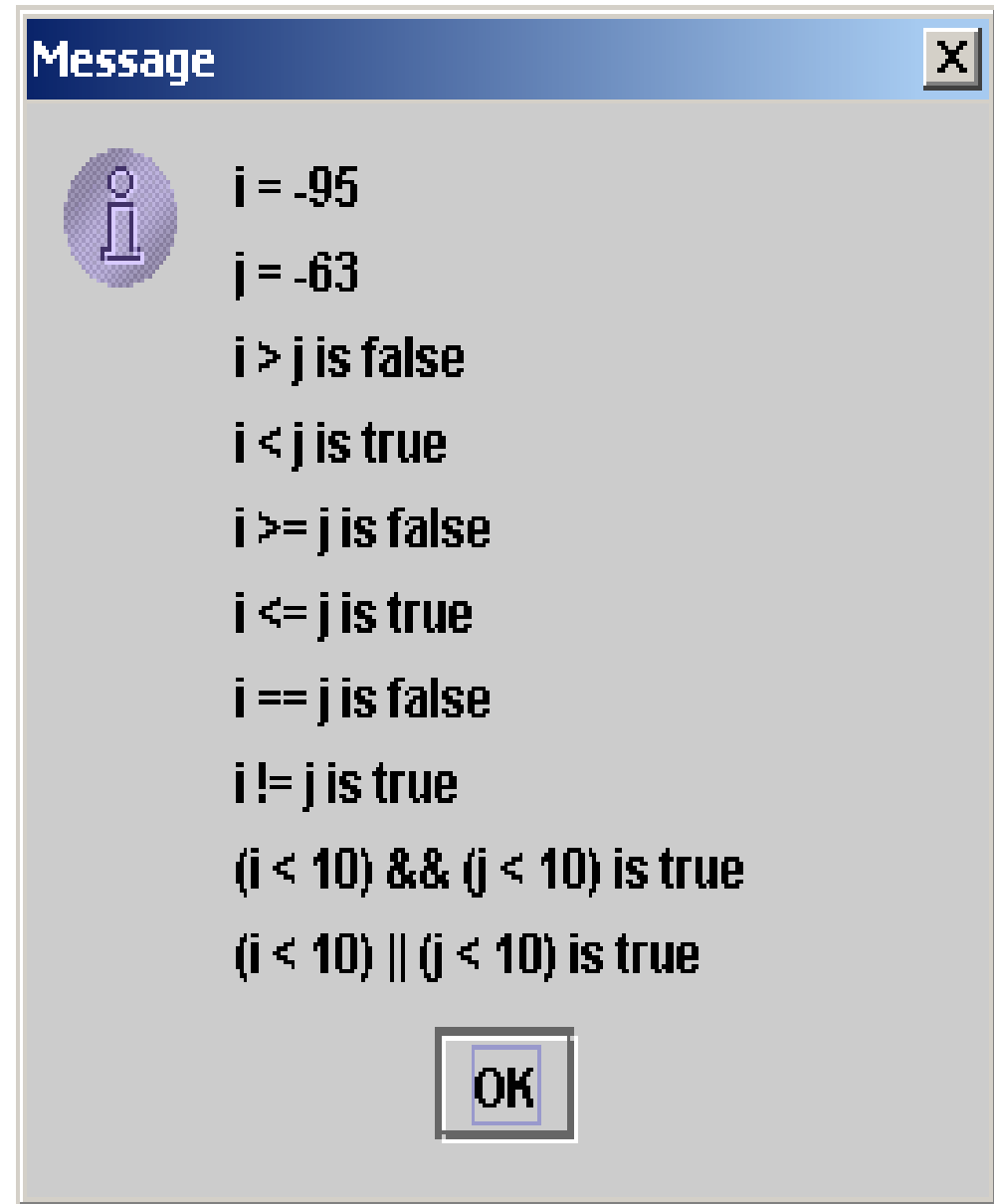
Przykład 3 z interfejsem graficznym

```
import java.util.*;
import javax.swing.*;

public class Lab2_6 {
    public static void main(String[] args)
    {
        Random rand = new Random();
        int i = rand.nextInt() % 100;
        int j = rand.nextInt() % 100;
        String s;
        s= "i = " + i + "\n";
        s+= "j = " + j + "\n";
        s+= "i > j is " + (i > j) + "\n";
        s+= "i < j is " + (i < j) + "\n";
        s+= "i >= j is " + (i >= j) + "\n";
        s+= "i <= j is " + (i <= j) + "\n";
        s+= "i == j is " + (i == j) + "\n";
        s+= "i != j is " + (i != j) + "\n";

        s+="(i < 10) && (j < 10) is " + ((i < 10) && (j < 10)) + "\n";
        s+="(i < 10) || (j < 10) is " + ((i < 10) || (j < 10)) + "\n";

        JOptionPane.showMessageDialog(null,s);
        System.exit(0);
    }
}
```



4. Funkcje statyczne, skrócone obliczanie wartości wyrażeń logicznych

```
public class Lab2_7
{
    static boolean test1(int w)
    {
        System.out.println("test1(" + w + ")");
        System.out.println("wynik: " + (w < 1));
        return w < 1;
    }

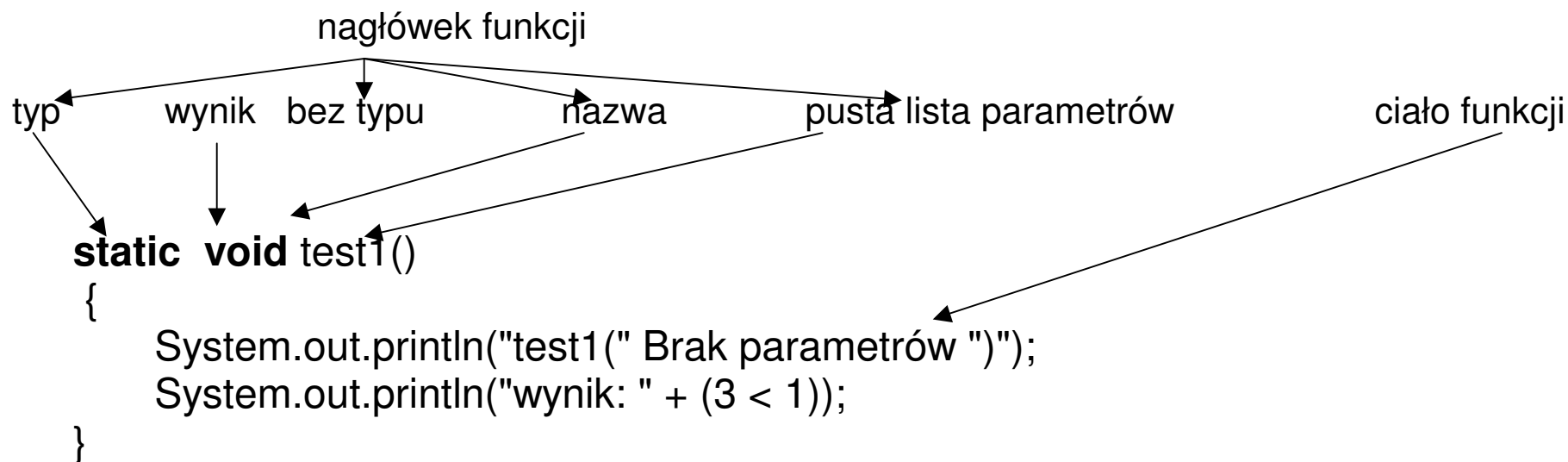
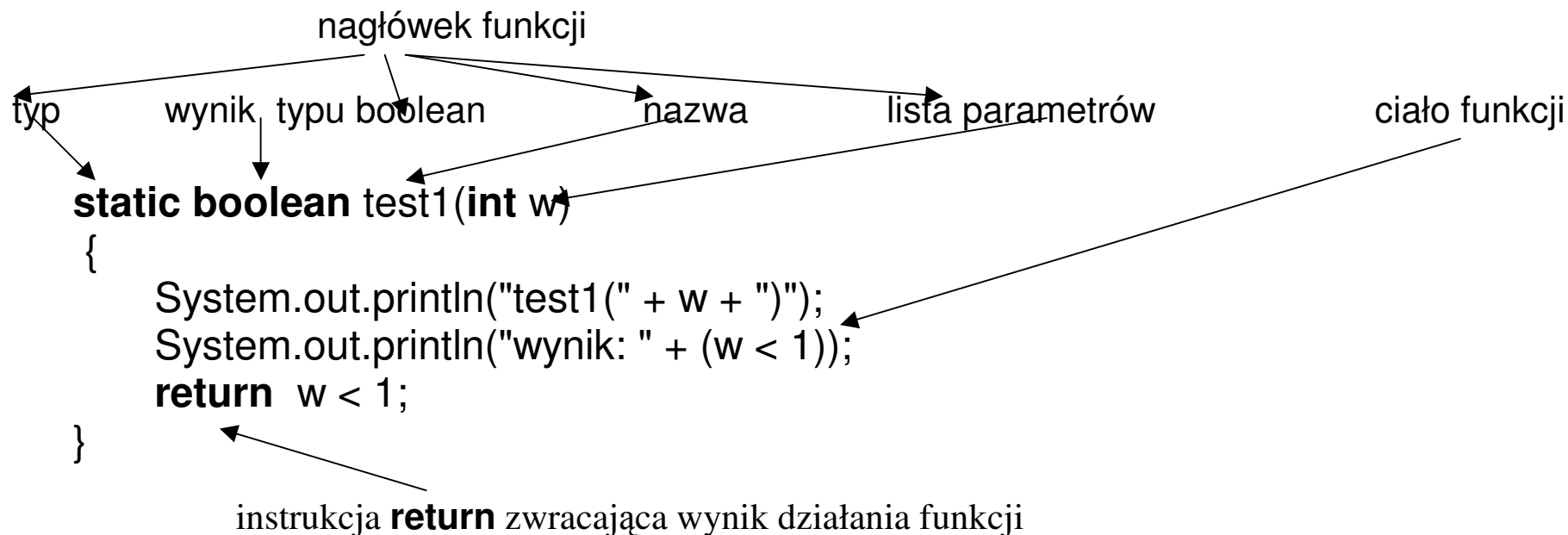
    static boolean test2(int w)
    {
        System.out.println("test2(" + w + ")");
        System.out.println("wynik: " + (w < 2));
        return w < 2;
    }

    static boolean test3(int w)
    {
        System.out.println("test3(" + w + ")");
        System.out.println("wynik: " + (w < 3));
        return w < 3;
    }

    public static void main(String[] args)
    {
        if(test1(0) && test2(1) && test3(2))
            System.out.println("\nwyrzazenie jest prawdziwe\n");
        else
            System.out.println("\nwyrzazenie jet falszywe\n");

        if(test1(0) && test2(2) && test3(2))
            System.out.println("\nwyrzazenie jest prawdziwe\n");
        else
            System.out.println("\nwyrzazenie jet falszywe\n");
    }
}
```

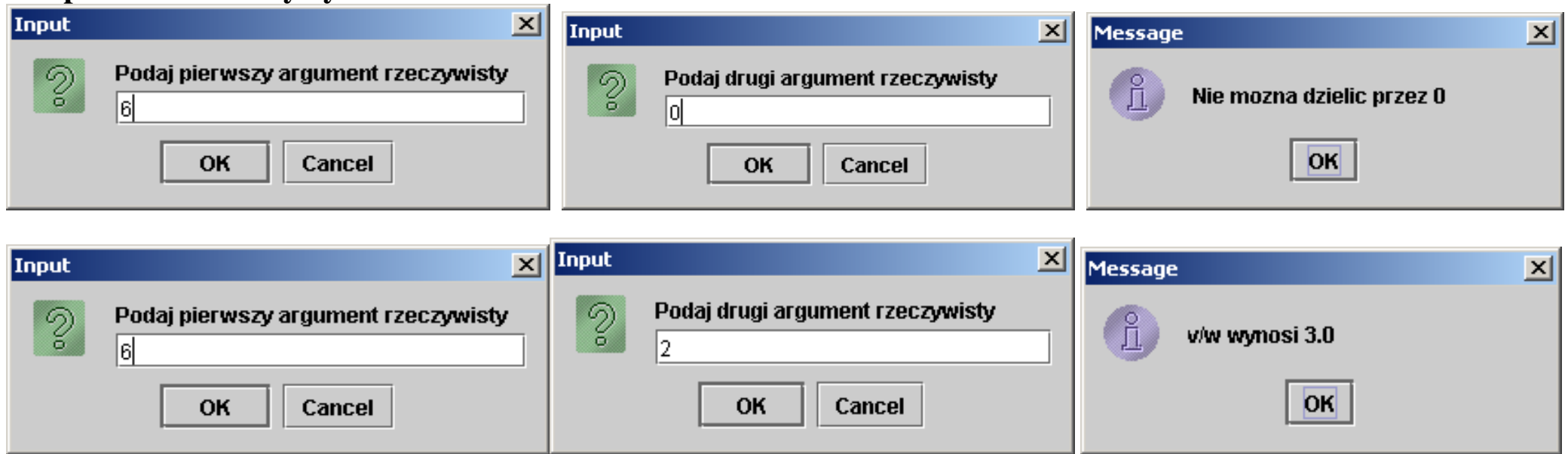
A screenshot of a Java application window. The window title bar shows the path 'C:\Program Files\Xinox Soft...'. The main content area has a black background with white text. The output of the program is displayed in a monospaced font. It shows three test cases: test1(0) with result 'true', test2(1) with result 'true', and test3(2) with result 'true'. A summary line says 'wyrzazenie jest prawdziwe'. Then it shows test1(0) with result 'true' and test2(2) with result 'false', followed by the summary 'wyrzazenie jet falszywe'. At the bottom, it says 'Press any key to continue...'. The window has standard Windows-style window controls (minimize, maximize, close) and a scrollbar on the right side.



void – brak typu

W przypadku obliczania iloczynu logicznego przerywa się obliczanie wartości jeśli jedno z podwyrażeń jest fałszywe.4.

5. Operator alternatywy ? :



```
import javax.swing.*;
import java.util.*;
public class Lab2_8
{
    public static void main(String[] args)
    {
        float v,w; String s;
        s=JOptionPane.showInputDialog(null,
            "Podaj pierwszy argument rzeczywisty");
        v=Float.parseFloat(s);
        s=JOptionPane.showInputDialog(null,
            "Podaj drugi argument rzeczywisty");
        w=Float.parseFloat(s);
        s= w==0 ? " Nie mozna dzielic przez 0" :
            "v/w wynosi" + v/w;
        JOptionPane.showMessageDialog(null,s);
        System.exit(0);
    }
}
```

```
import javax.swing.*;
import java.util.*;
public class Lab2_9
{
    public static void main(String[] args)
    {
        float v,w; String s;
        s=JOptionPane.showInputDialog(null,
            "Podaj pierwszy argument rzeczywisty");
        v=Float.parseFloat(s);
        s=JOptionPane.showInputDialog(null,
            "Podaj drugi argument rzeczywisty");
        w=Float.parseFloat(s);
        if (w==0) s="Nie mozna dzielic przez 0";
        else s="v/w wynosi" + v/w;
        JOptionPane.showMessageDialog(null,s);
        System.exit(0);
    }
}
```

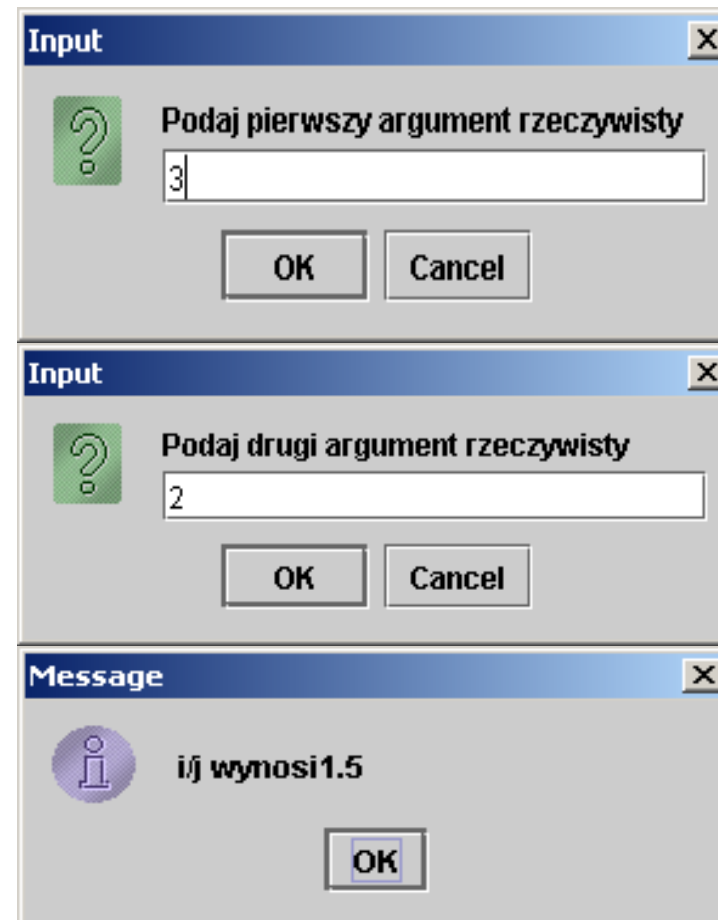
6. Operacje na łańcuchach – operator +, = oraz +=

Patrz przykłady. Zastosowanie w wyrażeniu podłańcucha wymusza konwersję pozostałych elementów na łańcuchy.

```
float v=3, w=2; String s;  
    s="v/w wynosi" + v/w;
```

7. Rzutowanie

```
import javax.swing.*;  
import java.util.*;  
public class Lab2_10  
{  
    public static void main(String[] args)  
    {  
        int i,j; String s;  
        s=JOptionPane.showInputDialog(null,  
            "Podaj pierwszy argument rzeczywisty");  
        i=Integer.parseInt(s);  
        s=JOptionPane.showInputDialog(null,  
            "Podaj drugi argument rzeczywisty");  
        j=Integer.parseInt(s);  
        if (j==0)  
            s = "Nie mozna dzielic przez 0";  
        else //rzutowanie do wartości rzeczywistej,  
            //aby podzielić z resztą  
            s = "i/j wynosi" + i/(float)j;  
  
        JOptionPane.showMessageDialog(null,s);  
        System.exit(0);  
    }  
}
```



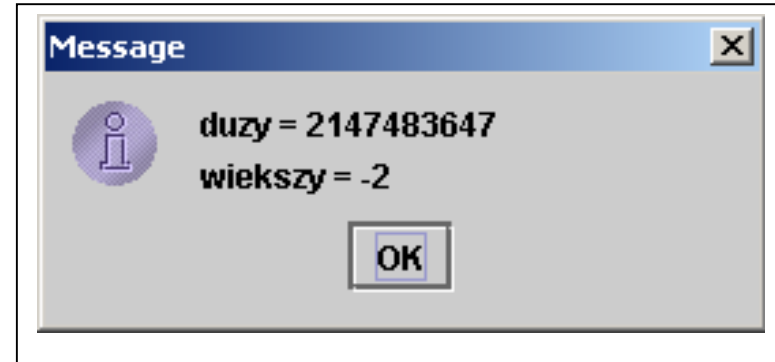
8. Przepelnienie

```
import javax.swing.*.*;
import java.util.*.*;
```

```
public class Lab2_11
{
    public static void main(String[] args)
    {
        int duzy = 0x7fffffff; // maksymalna wartość int
        int wiekszy = duzy * 2;
        String s = "duzy = " + duzy + "\n";
        s += "wiekszy = " + wiekszy + "\n";
        JOptionPane.showMessageDialog(null,s);
    }
}
```

/* Podczas kompilacji zgłaszany jest błąd przepelnienia dla liczb rzeczywistych

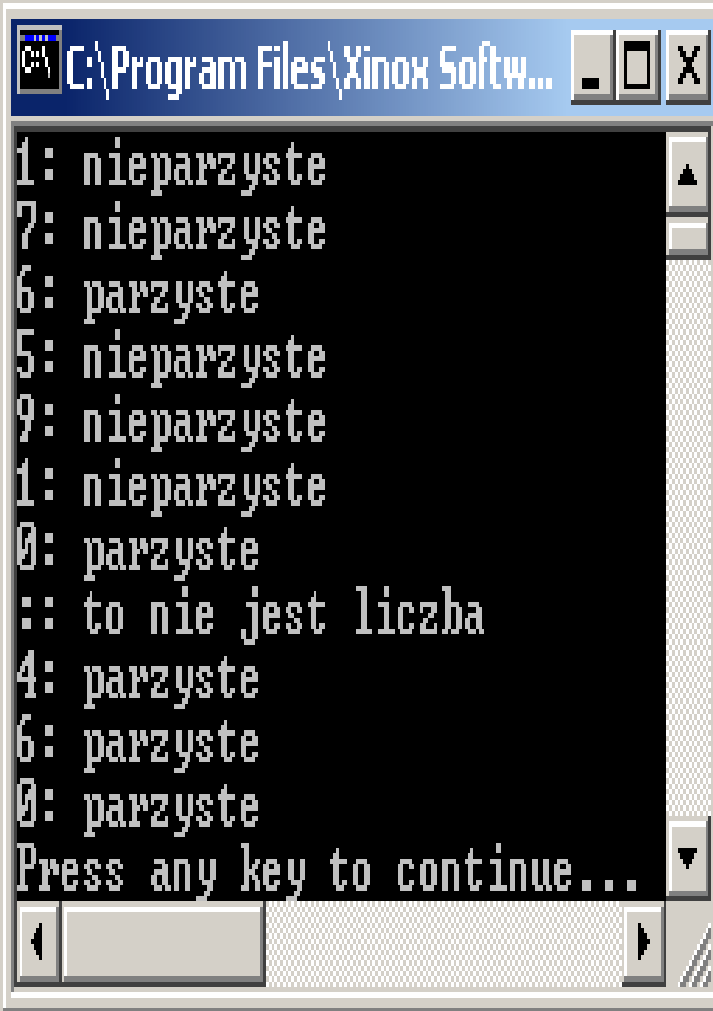
```
float wielki = 3.4E+38; // maksymalna wartość float
System.out.println("wielki = " + wielki);
int jeszcze_wiekszy= wielki * 4;
System.out.println ("jeszcze_wiekszy = " + jeszcze_wiekszy);
*/
System.exit(0);
}
```



9. Instrukcja switch

```
public class Lab2_12 {  
    public static void main(String[] args)  
    { int i=1;  
      do //losowanie kodu ASCII cyfry  
      { char c = (char)(Math.random() * 11 + '0');  
        System.out.print(c + ": ");  
        switch(c)  
        {case '0': i++;  
         case '2':  
         case '4':  
         case '6':  
         case '8':  
             System.out.println("parzyste");  
             break;  
         case '1':  
         case '3':  
         case '5':  
         case '7':  
         case '9':  
             System.out.println(  
                 "nieparzyste");  
             break;  
         default:  
             System.out.println("to nie jest liczba");  
         }  
      }while (i!=3);  
    }  
}
```

- **break** przerywa instrukcję **switch**.
- w przypadku braku **break** instrukcja przechodzi do następnego **case** po wybranym wcześniej bez słowa **break**.
- **default** jest wybierane, gdy wartość zmiennej **c** nie jest równa żadnej wartości przy **case**



```
C:\Program Files\Xinox Softw...  
1: nieparzyste  
7: nieparzyste  
6: parzyste  
5: nieparzyste  
9: nieparzyste  
1: nieparzyste  
0: parzyste  
:: to nie jest liczba  
4: parzyste  
6: parzyste  
0: parzyste  
Press any key to continue...
```