

## Podejście obiektowe. Tablice obiektów

### Przykład 1 – metody i atrybuty statyczne oraz niestacyjne

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

```
public class Napis4  
{  
    String wynik = "";  
    static int ile_obiektow = 0;
```

- 1) Metoda niestacyjna, oprócz atrybutów niestacyjnych i metod niestacyjnych, może używać atrybutów statycznych i metod statycznych danej klasy.
- 2) Należy utworzyć obiekt, aby wywołać metodę niestacyjną klasy tego obiektu.
- 3) Atrybut statyczny należy do klasy i dlatego jest wspólny dla wszystkich obiektów. Dlatego może być on używany w dowolnym kontekście: statycznym i niestacyjnym
- 4) W metodzie statycznej można używać tylko atrybutów statycznych i metod statycznych danej klasy

```
public void Inicjuj()  
{ ile_obiektow++;  
  wynik+="";  
}
```

- 1) Każdy obiekt, który używa w metodzie statycznej lub niestacyjnej atrybutu statycznego, używa tego atrybutu jako wspólnego:
- 2) Każdy obiekt ma własny atrybut **wynik** oraz wspólny atrybut **ile\_obiektow**

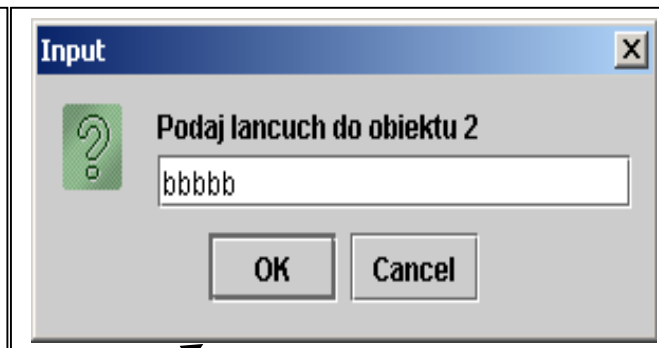
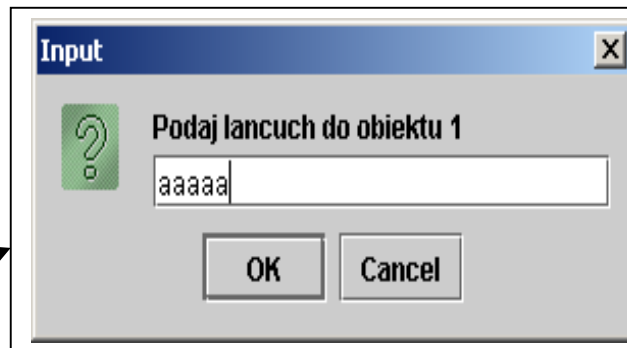
```
public void Dopisz_do_wyniku(String lan)  
{ wynik+=lan+"\n"; }
```

```
public void Rysuj_graficznie()  
{ JOptionPane.showMessageDialog(null, wynik); }
```

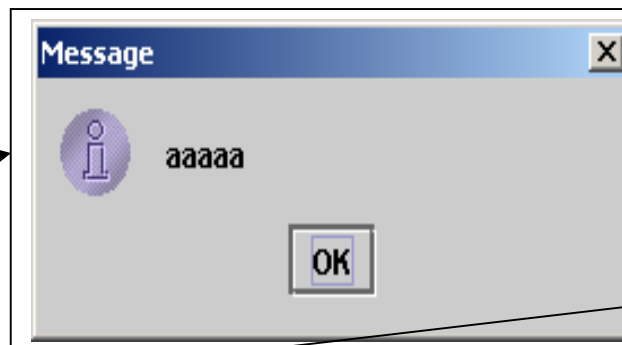
```
public void Rysuj_konsolowo()  
{ System.out.println(wynik); }
```

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

```
{ String s;  
  Napis4 napis1, napis2;  
  napis1=new Napis4();  
  napis2=new Napis4();  
  napis1.Inicjuj();  
  napis2.Inicjuj();  
  s=JOptionPane.showInputDialog(null,"Podaj lancuch do obiektu 1");  
  napis1.Dopisz_do_wyniku(s);  
  s=JOptionPane.showInputDialog(null,"Podaj lancuch do obiektu 2");  
  napis2.Dopisz_do_wyniku(s);
```



```
napis1.Rysuj_graficznie();
```



```
napis2.Rysuj_graficznie();
```

```
napis1.Rysuj_konsolowo();
```

```
napis2.Rysuj_konsolowo();
```

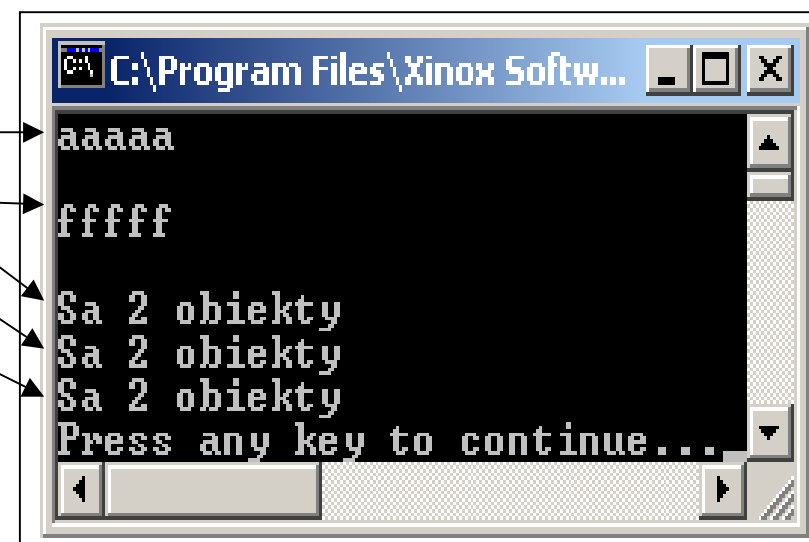
```
System.out.println("Sa "+napis1.ile_obiektow+" objekty");
```

```
System.out.println("Sa "+napis2.ile_obiektow+" objekty");
```

```
System.out.println("Sa "+Napis4.ile_obiektow+" objekty");
```

```
System.exit(0);
```

```
// nazwę obiektów napis1 I napis2 można pominąć przy odwołaniu do atrybutu  
// ile_obiektow, ponieważ jest to statyczny atrybut (wspólny dla obu obiektów)
```



## Przykład 2 Klasy z z atrybutami obiektowymi typu static

Klasa System

Atrybuty:

```
static PrintStream err //The "standard" error output stream.
```

```
static InputStream in //The "standard" input stream.
```

```
static PrintStream out // The "standard" output stream.
```

Metody //.....

```
System.out.println("Proba");
```

**out** jest atrybutem obiektowym statycznym, czyli istnieje jako obiekt, dlatego, można używać jego metod (statycznych i niestacyjnych) bez tworzenia obiektu klasy System. **W metodach innych klasach zawsze do wywoływanych składowych statycznych dodajemy nazwę ich klasy np. System.out**

Klasa PrintStream //metody dotyczące drukowania

```
void print(boolean b) Print a boolean value.
```

```
void print(char c) Print a character.
```

```
void print(char[] s) Print an array of characters.
```

```
void print(double d) Print a double-precision floating-point number.
```

```
void print(float f) Print a floating-point number.
```

```
void print(int i) Print an integer.
```

```
void print(long l) Print a long integer.
```

```
void print(Object obj) Print an object.
```

```
void print(String s) Print a string.
```

```
void println() Terminate the current line by writing the line separator string.
```

```
void println(boolean x) Print a boolean and then terminate the line.
```

```
void println(char x) Print a character and then terminate the line.
```

```
void println(char[] x) Print an array of characters and then terminate the line.
```

```
void println(double x) Print a double and then terminate the line.
```

```
void println(float x) Print a float and then terminate the line.
```

```
void println(int x) Print an integer and then terminate the line.
```

```
void println(long x) Print a long and then terminate the line.
```

```
void println(Object x) Print an Object and then terminate the line.
```

```
void println(String x) Print a String and then terminate the line.
```

### 1. Tablice w aplikacjach typu Windows

# Tablice

## 1. Tablica w Javie jest obiektem

- **Deklarowanie tablicy**

```
String [] nazwy; //równoważne deklaracje zmiennej tablicowej  
String nazwy []; // czyli referencji do obiektu tablicy, deklarujące elementy obiektowe  
int liczby[]; //zamienna tablicowa deklarująca elementy nieobektowe
```

- **Tworzenie obiektu tablicowego**

```
int liczby []= new int [10;]  
// utworzono tablicę 10 elementów typu int  
String nazwy []= new String[10];  
// utworzono tablicę 10 referencji typu String, należy dla każdego elementu tablicy przydzielić pamięć
```

- **Przydział pamięci na elementy obiektowe tablicy**

```
np. nazwy[0]= new String("Pewien wiersz");  
//przydzielono pamięć na obiekt typu String w pierwszym elemencie tablicy (liczby[0])
```

- **Pobranie rozmiaru tablicy**

```
nazwy.length
```

## Przykład 1

```
0
1
2
3
4
lancuch 0
lancuch 1
lancuch 2
lancuch 3
lancuch 4
Press any key to continue...
```

```
import java.lang.*;
public class Tablice
{
    public static void main(String args[])
    { final int N=5;

        int liczby [] = new int [N];
        for (int i=0; i<liczby.length; i++)
        {
            liczby[i]= i;
            System.out.println(liczby[i]);
        }

        String nazwy[]=new String[N];
        for (int i=0; i<nazwy.length; i++)
        {
            nazwy[i]=new String("lancuch "+i);
            System.out.println(nazwy[i]);
        }
    }
}
```

## 2. Tablice zawierające obiekty

### 2.1. Definicja elementu tablicy

#### Przykład 2

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

```
class Osoba1
```

```
{ String nazwisko;
```

```
  float srednia;
```

```
  String uwagi;
```

```
  static int ile_obiektow = 0;
```

```
  public void Inicjuj()
```

```
  {ile_obiektow++;}
```

```
  public void Nadaj_nazwisko(String lan)
```

```
  { nazwisko=lan;}
```

```
  public String Podaj_nazwisko()
```

```
  {return nazwisko;}
```

```
  public void Nadaj_uwagi(String lan)
```

```
  { uwagi=lan;}
```

```
  public String Podaj_uwagi()
```

```
  {return uwagi;}
```

```
  public void Nadaj_srednia(float srednia_)
```

```
  { srednia=srednia_;}
```

```
  public float Podaj_srednia()
```

```
  {return srednia;}
```

```
  public void Wstaw()
```

```
  { String S;
```

```
    Inicjuj();
```

```
    S = JOptionPane.showInputDialog(null, "Podaj nazwisko");
```

```
    Nadaj_nazwisko(S);
```

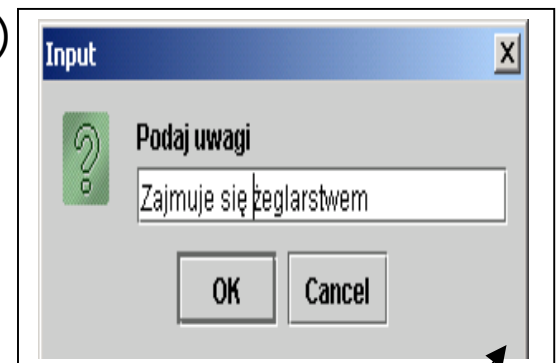
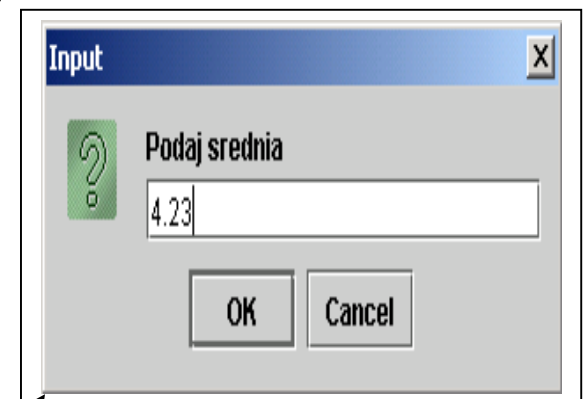
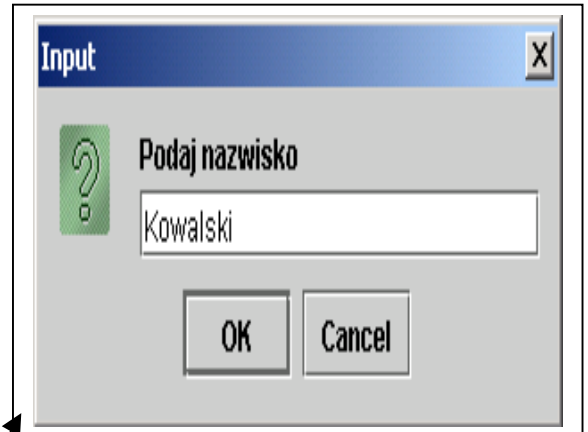
```
    S = JOptionPane.showInputDialog(null, "Podaj srednia");
```

```
    Nadaj_srednia(Float.parseFloat(S));
```

```
    S =JOptionPane.showInputDialog(null, "Podaj uwagi");
```

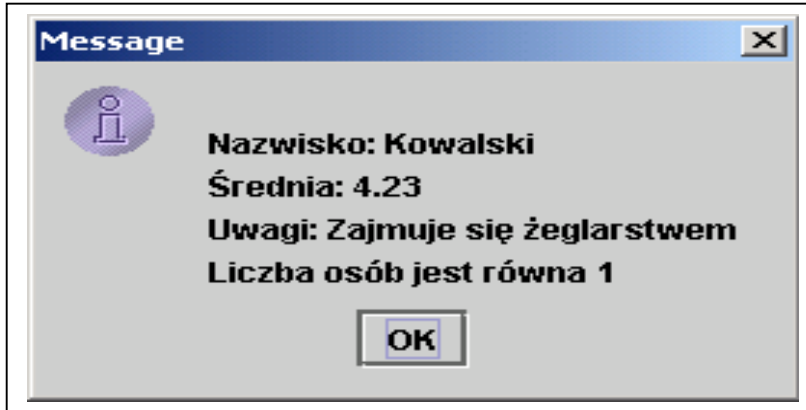
```
    Nadaj_uwagi(S);
```

```
  }
```



```
public void Wyświetl()
```

```
{String napis="";  
  napis+="\n Nazwisko: "+nazwisko;  
  napis+="\n Średnia: "+srednia;  
  napis+="\n Uwagi: "+uwagi;  
  napis+="\n Liczba osób jest równa "+ile_obiektow;  
  JOptionPane.showMessageDialog(null, napis);  
}
```



```
public class Cala
```

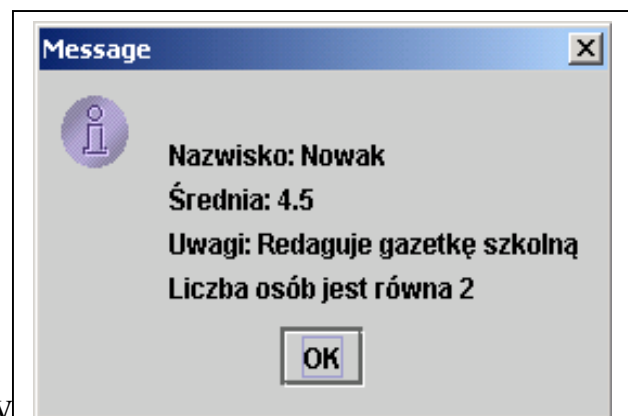
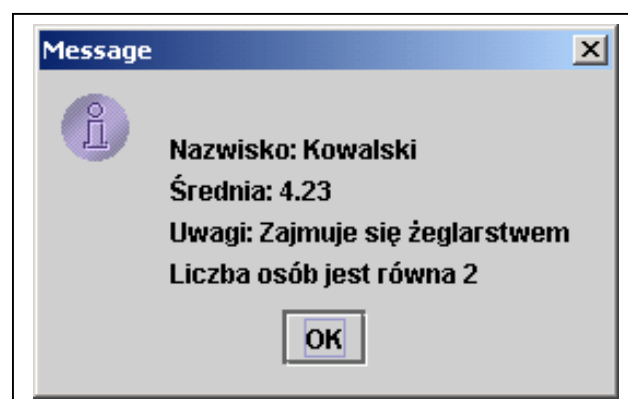
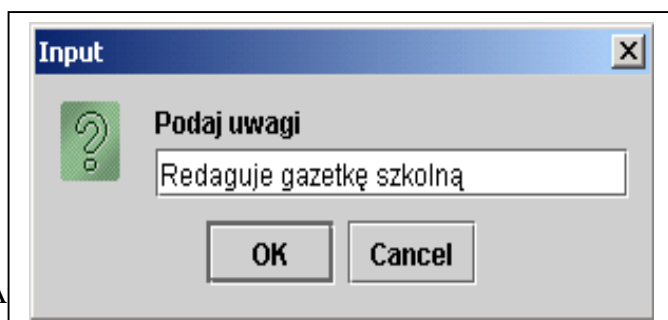
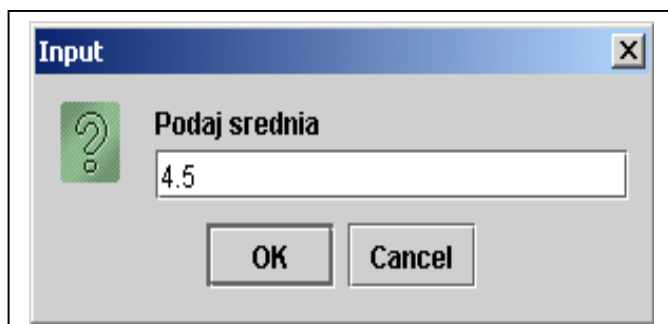
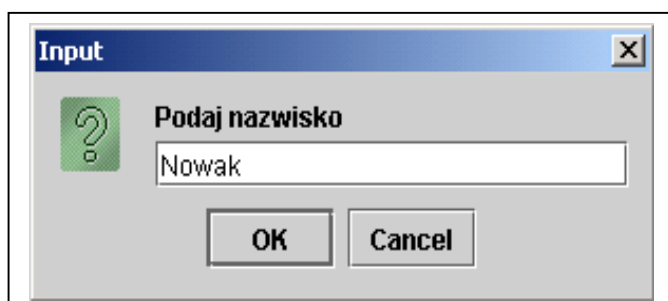
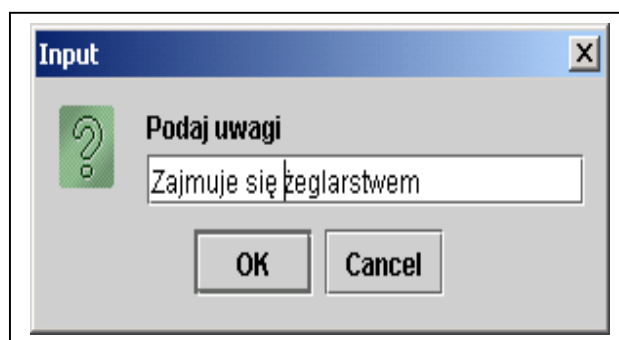
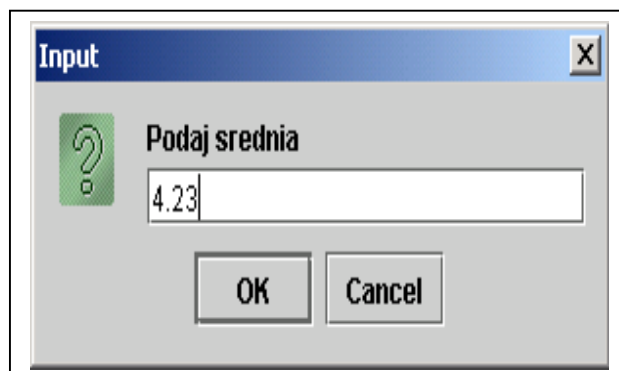
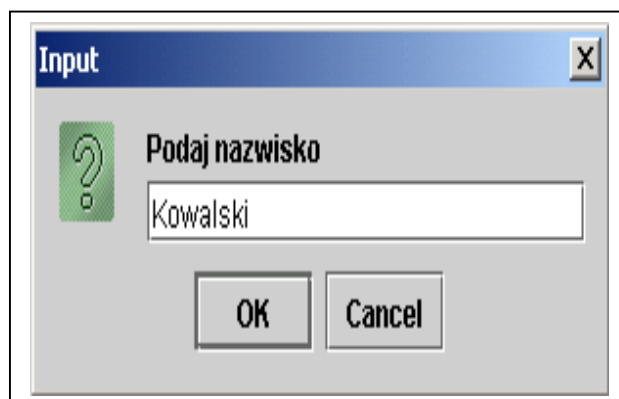
```
{ static public void main(String args[])  
  {Osoba1 os1 = new Osoba1();  
  os1.Wstaw();  
  os1.Wyświetl();  
  System.exit(0);  
  }  
}
```

## 2.2. Tablica obiektów – operacje wstawiania i wyświetlania

### Przykład 3

```
import javax.swing.*;
import java.util.*;
class Osoba1
{ // kod klasy z przykładu 2 zawarty w nawiasach {}
}
```

```
public class Tablica_osob
{ static int N=2;
  static public void main(String args[])
  { Osoba1 Dane[] = new Osoba1[N];
    int ile;
    for (ile=0; ile<Dane.length;ile++)
      { Dane[ile]=new Osoba1();
        Dane[ile].Wstaw();
      }
    for (int i=0; i<ile;i++)
      Dane[i].Wyswietl();
  }
}
```





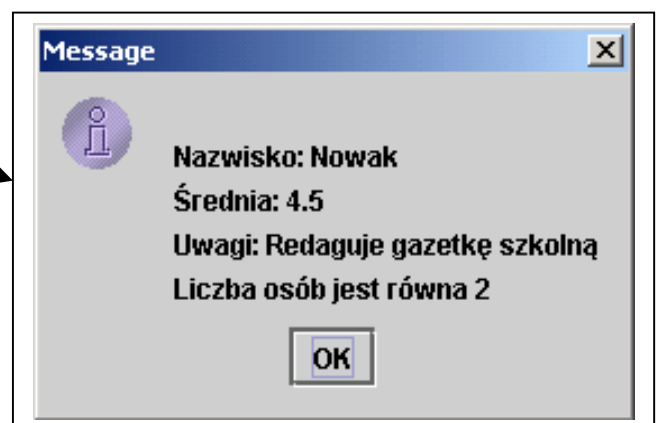
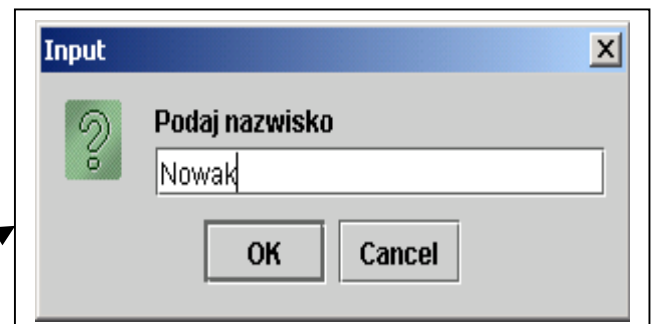
## 2.3. Tablica obiektów –operacje wstawiania, wyświetlania i wyszukiwania

### Przykład 4

```
import javax.swing.*;
import java.util.*;
class Osoba1
{ // kod klasy z przykładu 3 zawarty w nawiasach {}
    public boolean Szukaj(String s)
    {
        return nazwisko.equals(s);
    }
}

public class Tablica_osob_porownanie
{ static int N=2;
  static public void main(String args[])
  {Osoba1 Dane[] = new Osoba1[N];
   int ile;
   for (ile=0; ile<Dane.length;ile++)
   { Dane[ile]=new Osoba1();
     Dane[ile].Wstaw();
   }
   for (int i=0; i<ile;i++)
     Dane[i].Wyswietl();

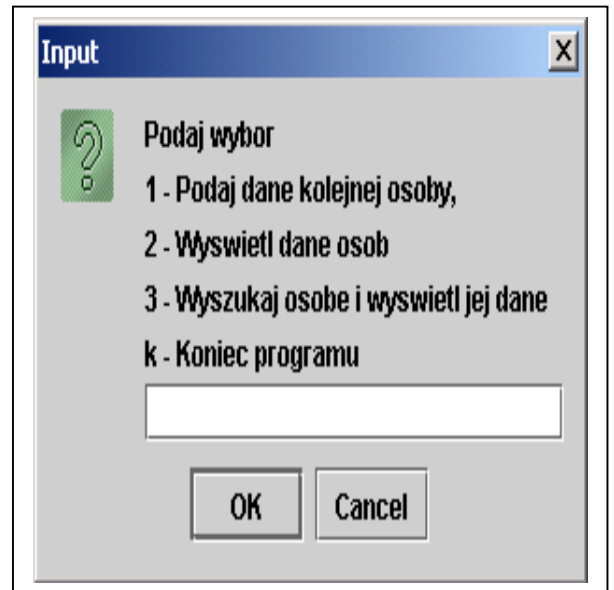
   String s= JOptionPane.showInputDialog(null,"Podaj nazwisko");
   for (int i=0; i<ile;i++)
   { if (Dane[i].Szukaj(s))
     Dane[i].Wyswietl();
   }
   System.exit(0);
}
}
```



## 2.4. Tablica obiektów –wybór wstawiania, wyświetlania i wyszukiwania

### Przykład 5

```
import javax.swing.*;
import java.util.*;
class Osoba1
{ // kod klasy z przykładu 4 zawarty w nawiasach {}
}
public class Tablica_osob_menu
{ static int N=2;
  static public void main(String args[])
  { Osoba1 Dane[] = new Osoba1[N];
    int ile=0;
    String s; char ch;
    do
    { s=JOptionPane.showInputDialog(null, "Podaj wybor"
      +"\n1 - Podaj dane kolejnej osoby,"
      +"\n2 - Wyświetl dane osob"
      +"\n3 - Wyszukaj osobe i wyswietl jej dane"
      +"\nk - Koniec programu");
      ch = s.charAt(0);
      switch(ch)
      { case '1' : for (ile=0; ile<Dane.length;ile++)
                  { Dane[ile]=new Osoba1();
                    Dane[ile].Wstaw();}
                break;
        case '2' : for (int i=0; i<ile;i++) Dane[i].Wyświetl();
                break;
        case '3' : s = JOptionPane.showInputDialog(null,"Podaj nazwisko");
                  for (int i=0; i<ile;i++)
                    { if (Dane[i].Szukaj(s)) Dane[i].Wyświetl();}
                  break;
        case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu");
                break;
        default : JOptionPane.showMessageDialog(null,"Zla opcja");
                }
    }while (ch != 'k') ;
    System.exit(0); }}
```



## 2.5. Tablica obiektów –operacje wyboru wstawiania, wyświetlania i wyszukiwania, zmiana rozmiarów tablicy

### Przykład 6

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

```
class Osoba1
```

```
{ String nazwisko;
```

```
  float srednia;
```

```
  String uwagi;
```

```
  static int ile_obiektow = 0;
```

```
  public void Inicjuj()
```

```
  { ile_obiektow++;}
```

```
  public void Nadaj_nazwisko(String lan)
```

```
  { nazwisko=lan;}
```

```
  public String Podaj_nazwisko()
```

```
  { return nazwisko;}
```

```
  public void Nadaj_uwagi(String lan)
```

```
  { uwagi=lan;}
```

```
  public String Podaj_uwagi()
```

```
  { return uwagi;}
```

```
  public void Nadaj_srednia(float srednia_)
```

```
  { srednia=srednia_;}
```

```
  public float Podaj_srednia()
```

```
  {return srednia;}
```

```
  public void Wstaw()
```

```
  { String S;
```

```
    Inicjuj();
```

```
    S = JOptionPane.showInputDialog(null, "Podaj nazwisko");
```

```
    Nadaj_nazwisko(S);
```

```
    S = JOptionPane.showInputDialog(null, "Podaj srednia");
```

```
    Nadaj_srednia(Float.parseFloat(S));
```

```
    S = JOptionPane.showInputDialog(null, "Podaj uwagi");
```

```
    Nadaj_uwagi(S);
```

```
  }
```

```
public void Wyświetl()
```

```
{String napis="";  
  napis+="\n Nazwisko: "+nazwisko;  
  napis+="\n Średnia: "+srednia;  
  napis+="\n Uwagi: "+uwagi;  
  napis+="\n Liczba osób jest równa "+ile_obiektow;  
  JOptionPane.showMessageDialog(null, napis);  
}
```

```
public boolean Szukaj(String s)
```

```
{ return nazwisko.equals(s);  
}  
}
```

```

public class Tablica_osob_menu
{
static int N; //zmienna typu static, która w programie przechowuje aktualny rozmiar tablicy
static public void main(String args[])
{
    Osoba1 Dane[]=null; //referencja do tablicy, która jest tworzona w opcji 1
    int ile=0;
    String s;
    char ch;
    do
    {
        s=JOptionPane.showInputDialog(null, "Podaj wybor"
            +" \n1 - Podaj dane kolejnej osoby,"
            +" \n2 - Wyświetl dane osob"
            +" \n3 - Wyszukaj osobę i wyświetl jej dane"
            +" \nk - Koniec programu");
        ch = s.charAt(0); //pobranie opcji
        switch(ch)
        {
            case '1' : s=JOptionPane.showInputDialog(null, "Podaj rozmiar tablicy");
                N=Integer.parseInt(s);
                Dane=new Osoba1[N]; //utworzenie tablicy
                for (ile=0; ile<Dane.length;ile++)
                    {
                        Dane[ile]=new Osoba1(); //tworzenie elementów tablicy
                        Dane[ile].Wstaw();}
                break;
            case '2' : if (Dane!=null) //jesli tablica istnieje, to można wyświetlać
                for (int i=0; i<ile;i++)
                    Dane[i].Wyświetl();
                break;
            case '3' : if (Dane==null) break; //jeśli tablica nie istnieje, nie można wyszukiwać
                s = JOptionPane.showInputDialog(null, "Podaj nazwisko");
                for (int i=0; i<ile;i++)
                    {
                        if (Dane[i].Szukaj(s))
                            Dane[i].Wyświetl();
                    }
                break;
            case 'k' : JOptionPane.showMessageDialog(null, "Koniec programu");
                break;
            default : JOptionPane.showMessageDialog(null, "Zła opcja");
        }
    }
    while (ch != 'k') ;
    System.exit(0); }}

```