

Przykłady zadań

1. Polimorfizm czyli korzystanie z metod zdefiniowanych z identycznym nagłówkiem (nazwa, parametry, wynik zwracany przez return). Podaj kody funkcji `podaj_typ()` i `pokaz()` w klasie `Towar1` i `Towar2`.

```
import java.lang.*;
import java.util.*;
interface Dane
{ String produkt = "Typ produktu: ";
public int podaj_typ(); }

class Towar1 implements Dane
{ int cena;
String nazwa;
Towar1(int cena_, String nazwa_)
{ cena=cena_; nazwa=new String (nazwa_); }
public String toString()
{ return ("cena:"+cena+", nazwa:"+nazwa+", "
+produkt+"."+podaj_typ()+"\n"); }
//zwraca łańcuch zawierający nazwę, cenę oraz typ produktu
public int podaj_typ(){return 1; } //zwraca typ produktu 1
}
class Towar2 extends Towar1
{ int podatek;

//zdefiniuj konstruktor inicjujący atrybuty obiektu, (zastosuj
//słowo super) i przypisujący podatek_ do składowej podatek
//oraz oraz podaj_typ ()
//metodę: toString: zwraca łańcuch z danymi obiektu klasy
//Towar2 oraz /Towar1; wykorzystaj słowo super w celu
//dziedziczenia metody toString z klasy Towar1
//metoda podaj_typ: zwraca typ produktu 2
}
public class Produkty
```

```
{ static Vector <Towar1> towary=new Vector<Towar1>();
static void wyprowadz(/*podać typ parametru p*/)
{ String s="";
towary.add(p);// dodaj do pojemnika towary nowy produkt
Iterator iterator = towary.iterator();
while(/* ...*/) // wstaw metodę iteratora,
{ Towar1 pom=(Towar1)iterator.next();
s+=/*....*/; } //wstaw metodę pobranego obiektu pom z
//kolekcji, która zwraca łańcuch z danymi kolejnego produktu
System.out.println(s); }

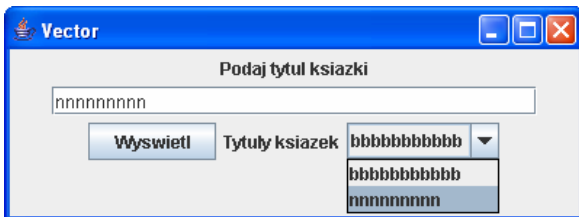
public static void main (String[] args)
{ Towar2 p2 = new Towar2(20,"ksiązka",5);
wyprowadz(p2);
//5
Towar1 p1 = new Towar1(15, "kalendarz");
wyprowadz(p1);
}
```

2. Napisz program, który umożliwi obsługę zdarzenia typu Action – zdefiniuj metody zarządcy zdarzeń obsługujące zdarzenie oraz przycisk1 typu `JButton` generujący zdarzenia – np. umieszczania tekstu z pola wiersz1 w liście tekst1

```
public class Aplikacja extends JFrame
implements ActionListener
{ JButton przycisk1 = new JButton(/*Podać parametry */);
JTextField wiersz1 = new JTextField(/*Podać parametry */);
JComboBox tekst1=new JComboBox();
String dana = new String ();
public Aplikacja()
{ //zdefiniować ciało konstruktora }
```

```
public void actionPerformed (ActionEvent evt)
{/*zdefiniować ciało metody obsługującej zdarzenie od
przycisków przycisk1, przycisk2*/}
public static void main(String[] arg) throws Exception
{ Aplikacja pr= new Aplikacja();
pr.show();}
}
```

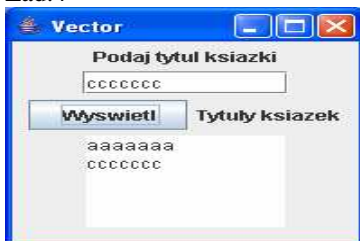
Zad.3.



3. Napisz program, który:

- 1) tworzy okno pokazane z lewej strony
- 1) wstawia z pola typu `JTextField` (Podaj tytuł książki) dane typu `String` do wektora dane typu `Vector` metodą `add` po naciśnięciu klawisza `Enter`
- 2) po naciśnięciu klawisza `Wyświetl` pobiera dane z wektora i umieszcza w liście typu `JComboBox` (Tytuły książek)

Zad.4



3. Napisz program, który:

- 1) tworzy okno pokazane z lewej strony
- 3) wstawia z pola typu `JTextField` (Podaj tytuł książki) dane typu `String` do wektora dane typu `Vector` metodą `add` po naciśnięciu klawisza `Enter`
- 4) po naciśnięciu klawisza `Wyświetl` pobiera dane z wektora i umieszcza w obszarze typu `JTextArea` (Tytuły książek)