1. Dziedziczenie metod bez przedefiniowania i z przedefiniowaniem: rola słowa kluczowego super; przeciążanie metod.
Zdefiniuj metody *pokaz()* w klasach Osoba1 (bez parametrów) i Osoba2 (z parametrem), tak, aby poprawnie działała funkcja *main* w klasie *Osoba oraz konstruktor w Osoba2*.

```
class Osoba1
  { int wiek;
    String nazwisko;
    Osoba1(int wiek_, String nazwisko_)
     {    wiek=wiek_;     nazwisko=new String (nazwisko_);}
    void pokaz()        { /*...*/  }        //na ekranie nazwisko i
wiek
   boolean porownaj(Osoba1 os) {} // true, gdy osoby są tego
samego typu
  }  //nie wolno stosowac operatora  instanceof
class Osoba2 extends Osoba1
 { int pobory;
    /*zdefiniuj konstruktor, który dziedziczy po Osoba1 i
przypisuje pobory_ do składowej pobory oraz  metode pokaz,
która na ekranie wyświetla  dane: wiek, nazwisko za pomocą
dziedziczonej metody pokaz() oraz pobory wraz przekazanym
przez nagłówek łańcuchem zawierającym menu- wykorzystaj
słowo super*/
}
public class Osoba
{  public static void main (String[] args)
 {  Osoba2 p2 = new Osoba2(20,"Kowalski",5);
    Osoba1 p1 = new Osoba1(15, "Nowak");
    p2.pokaz();        //Kowalski, 20
    p2.pokaz("nazwisko, wiek, pobory: ");
                       // nazwisko, wiek, pobory: Kowalski, 20
                          5
    p1.pokaz();    //Nowak, 15
    p1.porownaj(p2);   } }  //false
```

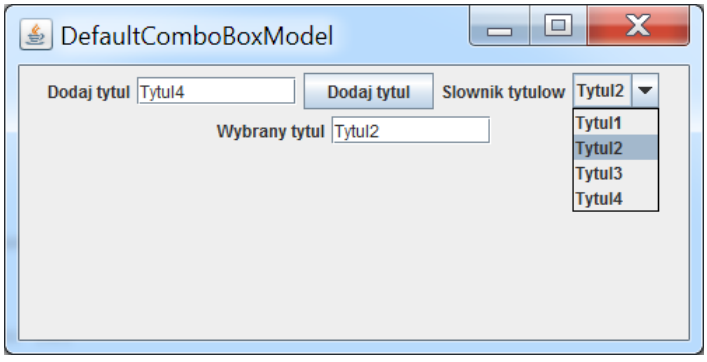
---

2. Polimorfizm czyli korzystanie z metod przedefiniowanych z identycznym nagłówkiem (nazwisko, parametry, wynik zwracany przez return). Podaj kody funkcji podaj_typ(), pokaz(), porównaj(Osoba1) w klasie Osoba1 i podaj_typ(), pokaz(), podaj_pobory() w klasie Osoba2 oraz metody wyprowadz w klasie Osoba. Nagłówki metod podaj_typ(), pokaz() oraz podaj_pobory() są przesłaniane w klasie Osoba2.

```
public interface Dane
 {  String produkt ="Typ osoby:  ";
    public  void podaj_typ(); }
class Osoba1 implements  Dane
  { int wiek;
    String nazwisko;
    Osoba1(int wiek_, String nazwisko_)
       { wiek=wiek_;      nazwisko=new String (nazwisko_); }
    void pokaz()      { /*...*/  }    //wyświetla nazwisko i wiek
    int podaj_pobory () {return 0;}
    public void podaj_typ ()   {/*...*/ } //wyświetla typ osoby
   boolean porownaj(Osoba1 os) {} /* true, gdy osoby mają te
same wartości atrybutów; należy zastąpić sprawdzanie atrybutu
pobory sprawdzaniem wyniku zwracanego przez metodę
podaj_pobory();nie wolno stosować operatora  instanceof */
  }
class Osoba2  extends Osoba1
 { int pobory;
/*zdefiniuj konstruktor inicjujący atrybuty obiektu, (zastosuj
słowo super) i  przypisuje pobory_ do składowej pobory i dwie
metody: pokaz(), podaj_typ () oraz podaj_pobory().
Pierwsza wyświetla na ekranie dane: wiek, nazwisko, za pomocą
dziedziczonej metody pokaz oraz pobory - do definicji
wykorzystaj słowo super; druga podaje typ osoby
(implementacja metody podaj_typ(); trzecia zwraca wartość
poborów  */
```

```
pp public class Osoba
  { ArrayList dane = new ArrayList(); // popraw w stylu Javy1.5
  static void wyprowadz(/*podać typ parametru p*/)
    {/* wstaw do pojemnika dane i za pomocą iteratora wyświetl
dane w wektorze za pomocą metody pokaz i podaj_typ  obiektów
wstawionych do //wektora*/
  }
public static void main (String[] args)
  {  Osoba2 p2 = new Osoba2(20,"Kowalski",5);
     Osoba1 p1 = new Osoba1(15, "Nowak");
      wyprowadz(p2);          //Kowalski, 20
                               // Typ  osoby 2
                              //5
      wyprowadz(p1);          //Kowalski, 20
                               // Typ  osoby 2
                               5
                               // Nowak 15
                              // Typ osoby 1
     p1.porownaj(p2);      //false
     p2.porownaj(p1);      //false
     p2.porownaj(p2);      //true
     p1.porownaj(p1);      //true
     }
```

Zad.3.



3. Napisz szkielet programu (utworzenie okna i wykonanie obsługi zdarzeń), który:
1) tworzy okno pokazane z lewej strony
1) wstawia z pola typu JTextField („Podaj tytul") dane typu String do kolekcji dane typu ArrayList metodą add po naciśnięciu klawisza „Dodaj tytul" i automatycznie wyświetla aktualną zawartość kolekcji w komponencie „Slownik tytulow" typu JComboBox
2) po naciśnięciu na wybrana pozycję listy komponentu „Slownik tytulow" wyswietla jej zawartość w polu „Wybrany tytul"