

## zadania1.txt

Hello.java

```
//główna klasa progr
//główna klasa programu:
public class Hello {
    //główna metoda programu:
    public static void main(String args[]){
        System.out.print("witaj swiecie!"); //ten tekst w nawiasie zostanie
        wyświetlony:
    }
}
```

Program nr 2

Plik zawiera definicje dwóch klas Person oraz Proba . Pierwsza z nich definiuje typ w postaci, w którym będziemy przechowywać dane na temat przykładowych osób Jana Kowalskiego i Anny Nowak. Klasa Proba zawiera przykład zastosowania klasy Person w krótkim programie wpisującym dane osobowe naszych bohaterów Jana i Anny oraz wyświetlającym je w zrozumiałej dla użytkownika formie. Jak widać z przykładu, pomimo tego, że wzorzec jest wspólny dla trzech obiektów (definiacja klasy Person), każdy z nich przechowuje swoje niezależne dane. Powyższy kod jest jedynie przykładem i zastosowano w nim konwencje, których w normalnym programowaniu nie powinno się stosować (np. publiczne atrybuty obiektu).

```
public class Proba {
public static void main(String[] args) {
    Person p, q, r;
    p = new Person();
    q = new Person();
    r = new Person();

    p.firstname = "Jan";
    p.lastname = "Kowalski";
    p.year = 1981;
    p.PESEL = "81111224350";

    q.firstname = "Anna";
    q.lastname = "Nowak";
    q.year = 1975;
    q.PESEL = "75032074926";

    System.out.println( p.firstname + " " + p.lastname + ", " +
                        p.year + ", PESEL: " + p.PESEL );

    System.out.println( q.firstname + " " + q.lastname + ", " +
                        q.year + ", PESEL: " + q.PESEL );

    }
}

class Person
{
    public String firstname, lastname;
    public int year;
    public String PESEL;
}
}
```

Program - prosta grafika

zadania1.txt

```

import java.awt.*;
import javax.swing.*;

public class Rysik
{
    public static void main(String[] args)
    {
        JFrame okno = new JFrame("Okno");
        okno.add(new Plansza());

        okno.setSize(100,100);

        // dzięki tej linii program zakończy się po zamknięciu okna
        okno.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        /*
        * bez poniższej linii kodu nasze okno będzie niewidoczne
        * ustawienie setVisible(false) ukrywa okno, lecz go nie niszczy
        * to znaczy, że możemy ukryć okno, a za chwilę je pokazać
        * bez konieczności ponownego tworzenia go od podstaw
        */
        okno.setVisible(true);
    }
}

class Plansza extends JPanel
{
    Plansza()
    {
        // te informacje wydrukowane zostaną w konsoli/terminalu, a nie w
        oknie programu!!
        System.out.println("szerokość planszy:" + this.getWidth());
        System.out.println("wysokość planszy:" + this.getHeight());
    }

    /*
    * w tej funkcji umieszczamy kod ze wszystkim, co chcemy narysować
    * jest to funkcja, która wywoływana jest automatycznie przez Javę
    * za każdym razem, gdy zachodzi taka potrzeba (np. zmiana wielkości
    * okna przez użytkownika); lepiej nie wywoływać jej na własną rękę
    */
    public void paint(Graphics g)
    {
        /*
        * umieszczenie tych zmiennych jako zmiennych lokalnych
        * funkcji paint() zapewni aktualizację tych zmiennych
        * podczas zmiany wielkości okna przez użytkownika
        */
        int width = this.getWidth();
        int height = this.getHeight();

        // obiekt graficzny g rysuje linię po przekątnej panelu z marginesem
        10 pikseli g.drawLine(10, 10, width - 10, height - 10);
    }
}

```

zadania1.txt

Program nr 3 - auta

```
class Vehicle
{
    private String owner;
    public String model;
    public int type; //1,2,3 - typy samochodow
    private int reg;

    public void setOwner(String s) { owner=s; }
    public String getOwner() { return owner; }

    // public String toString()
    // { return model + ", " + type + "\nReg.No." + reg +
    //   ", wlasnosc: " + owner + "\n"; }

    public String toString()
    { return model + ", " + typeToString() + "\nNr " + reg +
      ", wlasnosc: " + owner + "\n"; }

    private static int count=1000; /* poniewaz jest static, z kazdego
      obiektu nastepuje odwołanie do tej
      samej komorki pamieci */

    Vehicle() { reg = count++; } // konstruktor klasy

    private String typeToString()
    {
        if (type == 1) return "rower";
        else if (type == 2) return "motocykl";
        else if (type == 3) return "auto";
        else return "NIEZNANY!";
    }
}

class CrashTest2
{
    public static void main(String[] args)
    {
        Vehicle v1 = new Vehicle();
        Vehicle v2 = new Vehicle();
        Vehicle v3 = new Vehicle();

        v1.model = "Syrena 105 Turbo";
        v1.type = 3;
        v1.setOwner("Jan Kowalski");
        System.out.println(v1);

        v2.model = "Harley D., 1965";
        v2.type = 2;
        v2.setOwner("Crazy Jackill");
        System.out.println(v2);

        v3.model = "Romet wigry 3";
        v3.type = 1;
        v3.setOwner("John Brown");
        System.out.println(v3);
    }
}
```