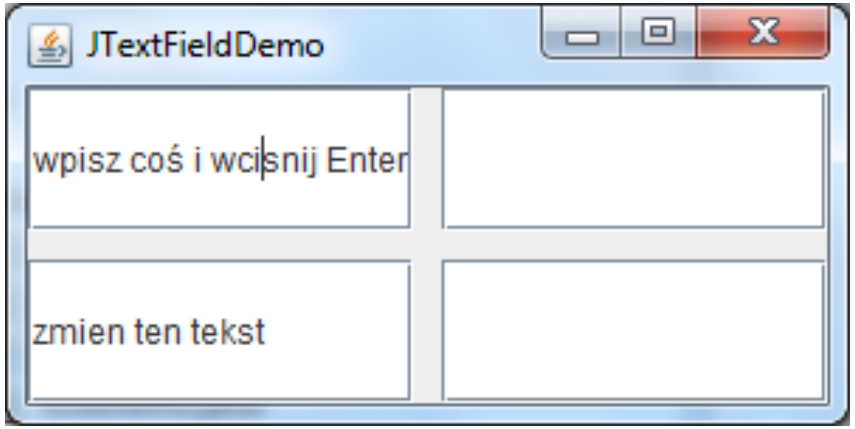
Kontynuacja wprowadzenia do SWING

Przykłady implementacji
wybranych komponentów

JTextFieldDemo.java




```
JTextField pole2 = new JTextField(„wpisz cos i wcisnij Enter");
JTextField pole2 = new JTextField(20);
JTextField pole3 = new JTextField("zmien ten tekst");
JTextField pole4 = new JTextField(20);

pole1.addActionListener(pole1Listener);
pole3.addKeyListener(pole3Listener);
//pole3.addKeyListener(pole3Adapter);
add(pole1);
add(pole2);
add(pole3);
add(pole4);
```

JTextFieldDemo.java

```
ActionListener pole1Listener = new ActionListener() {
            public void actionPerformed(ActionEvent e)
            {
                  pole2.setText( pole1.getText()) ;
            }
   };



  KeyListener pole3Listener = new KeyListener() {
            @Override
            public void keyTyped(KeyEvent e) {
            }

            @Override
            public void keyReleased(KeyEvent e) {
                    pole4.setText( pole3.getText()) ;
            }

            @Override
            public void keyPressed(KeyEvent e) {
            }
   };
```

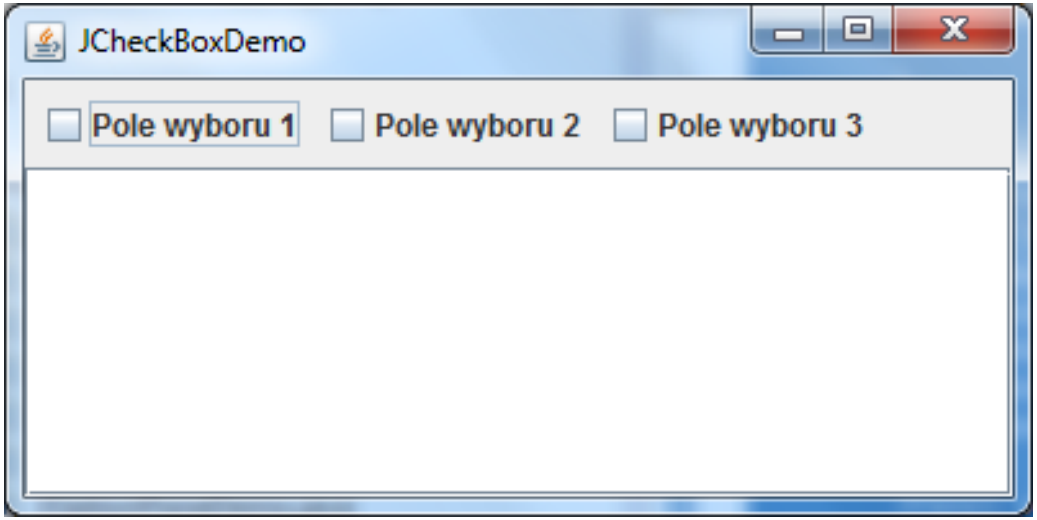
JTextFieldDemo.java

```
// Jesli nie wszystkie metody z KeyListener sa wykorzystane
// bardziej przejrzystej jest korzystanie z KeyAdaptera
// implementujacego wybrane metody:
KeyListener pole3Adapter = new KeyAdapter() {
        public void keyReleased(KeyEvent e) {
                pole4.setText( pole3.getText()) ;
        }
};
```

Inne przykłady adapterów:
MouseAdapter (zamiast MouseListener)
MouseMotionAdapter (zamiast MouseMotionListener)
WindowAdapter (zamiast WindowListener)

JCheckBoxDemo.java

```
JCheckBox cb1 = new JCheckBox("Pole wyboru 1");
JCheckBox cb2 = new JCheckBox("Pole wyboru 2");
JCheckBox cb3 = new JCheckBox("Pole wyboru 3");

cb1.addActionListener(cbListener);
cb2.addActionListener(cbListener);
cb3.addActionListener(cbListener);

Panel panelCheckBox = new JPanel();
panelCheckBox.setLayout(new FlowLayout(FlowLayout.LEFT));

panelCheckBox.add(cb1);
panelCheckBox.add(cb2);
panelCheckBox.add(cb3);
```



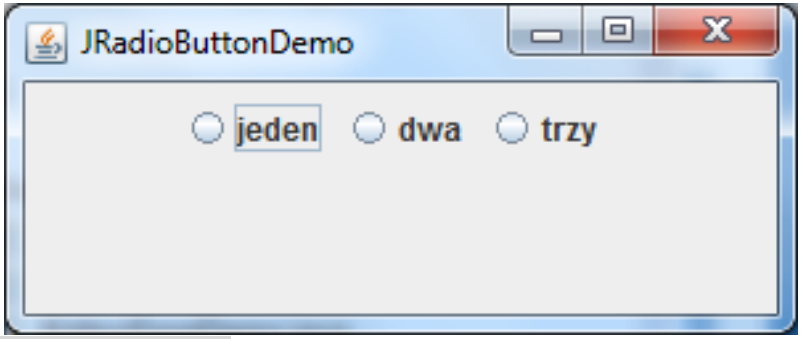
JCheckBoxDemo.java

```
ActionListener cbListener = new ActionListener() {

      public void actionPerformed(ActionEvent e)
      {
            Object obj = e.getSource();

            if (obj == cb1)
              if (cb1.isSelected()) t.append("Pole 1 ustawione\n");
                else t.append("Pole 1 wyczyszczone\n");

            if (obj == cb2)
              if (cb2.isSelected()) t.append("Pole 2 ustawione\n");
                else t.append("Pole 2 wyczyszczone\n");

            if (obj == cb3)
              if (cb3.isSelected()) t.append("Pole 3 ustawione\n");
                else t.append("Pole 3 wyczyszczone\n");

      }
};
```

Przykład obsługi kilku CheckBox'ów w jednym interfejsie…

JRadioButtonDemo.java

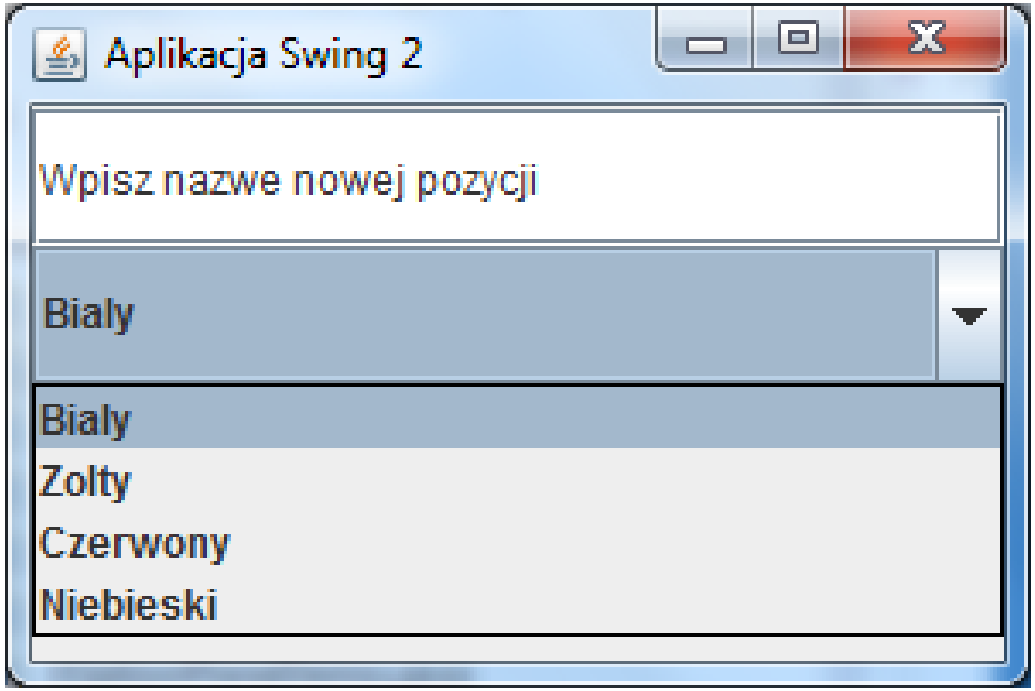
```
    ButtonGroup grupa = new ButtonGroup();
    JRadioButton
      rb1 = new JRadioButton("jeden", false),
      rb2 = new JRadioButton("dwa", false),
      rb3 = new JRadioButton("trzy", false);

// Grupowanie obektow JRadioButton do ButtonGroup
//- tylko jeden moze byc zaznaczony
    grupa.add(rb1); grupa.add(rb2); grupa.add(rb3);
    rb1.addActionListener(listener);
    rb2.addActionListener(listener);
    rb3.addActionListener(listener);

    add(rb1); add(rb2); add(rb3);
```




```
   ActionListener listener = new ActionListener() {
     public void actionPerformed(ActionEvent e) {
       etykieta.setText("Przycisk wyboru " +
         ((JRadioButton)e.getSource()).getText());
     }
   };
```

JComboBoxDemo.java

```
String[] description = { "Bialy", "Zolty",
            "Czerwony", "Niebieski", };
JTextField poleTekstowe = new JTextField("Wpisz nazwe nowej pozycji");
JComboBox comboBox = new JComboBox(description);
JButton przycisk = new JButton("Dodaj pozycje");
JLabel etykieta = new JLabel();
```



JComboBoxDemo.java

```
String[] description = { "Bialy", "Zolty",
            "Czerwony", "Niebieski", };
JTextField poleTekstowe = new JTextField("Wpisz nazwe nowej pozycji");
JComboBox comboBox = new JComboBox(description);
JButton przycisk = new JButton("Dodaj pozycje");
JLabel etykieta = new JLabel();

comboBox.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e){
          etykieta.setText("indeks: "+ comboBox.getSelectedIndex()
            + "    " + comboBox.getSelectedItem());
          if (comboBox.getSelectedItem().equals("Bialy"))
          etykieta.setText("Wybrano kolor Bialy");

        }

add(comboBox);
```

JComboBoxDemo.java

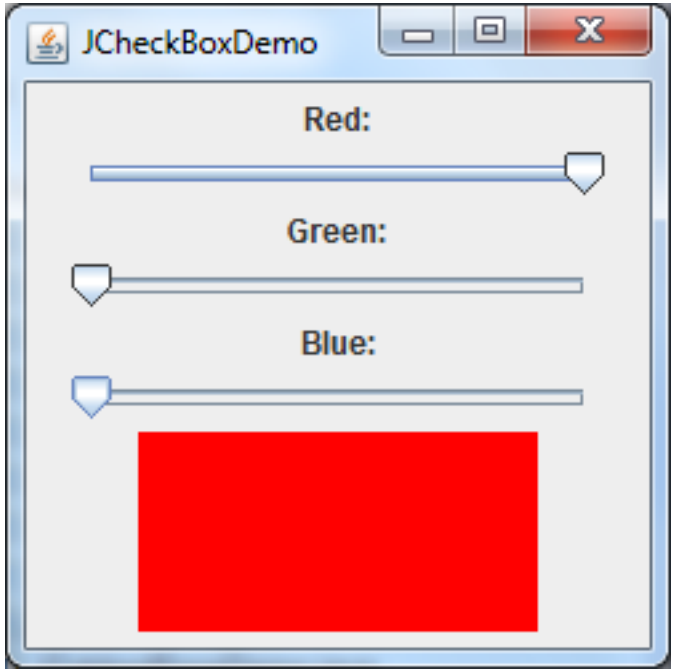
```
String[] description = { "Bialy", "Zolty",
             "Czerwony", "Niebieski", };
JTextField poleTekstowe = new JTextField("Wpisz nazwe nowej pozycji");
JComboBox comboBox = new JComboBox(description);
JButton przycisk = new JButton("Dodaj pozycje");
JLabel etykieta = new JLabel();

 comboBox.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e){
          etykieta.setText("indeks: "+ comboBox.getSelectedIndex()
            + "   " + comboBox.getSelectedItem());
          if (comboBox.getSelectedItem().equals("Bialy"))
           etykieta.setText("Wybrano kolor Bialy");
        }

add(comboBox);

przycisk.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e){
         if (poleTekstowe.getText() != "")
        comboBox.addItem(poleTekstowe.getText());
          etykieta.setText("Dodano: " + poleTekstowe.getText());
         }
      });
```

JSliderDemo.java

```
JSlider redSlider,greenSlider,blueSlider;

redSlider = new JSlider();
redSlider.setMinimum(0);
redSlider.setMaximum(255);
redSlider.setValue(127);

greenSlider = new JSlider(0,255,127);

blueSlider = new JSlider(JSlider.HORIZONTAL, 0, 255, 127);

redSlider.addChangeListener(slidersListener);
greenSlider.addChangeListener(slidersListener);
blueSlider.addChangeListener(slidersListener);

add(new JLabel("Red:"));
add(redSlider);
add(new JLabel("Green:"));
add(greenSlider);
add(new JLabel("Blue:"));
add(blueSlider);
```

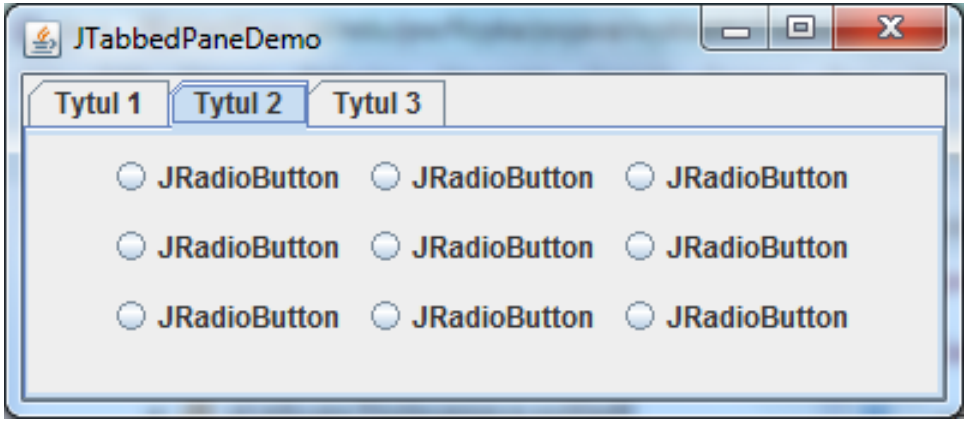
JSliderDemo.java

```
JPanel panel = new JPanel();

ChangeListener slidersListener = new ChangeListener() {

            @Override
    public void stateChanged(ChangeEvent e) {
            int red = redSlider.getValue();
            int green = greenSlider.getValue();
            int blue = blueSlider.getValue();
            panel.setBackground(new Color(red, green, blue));
    }
};
```



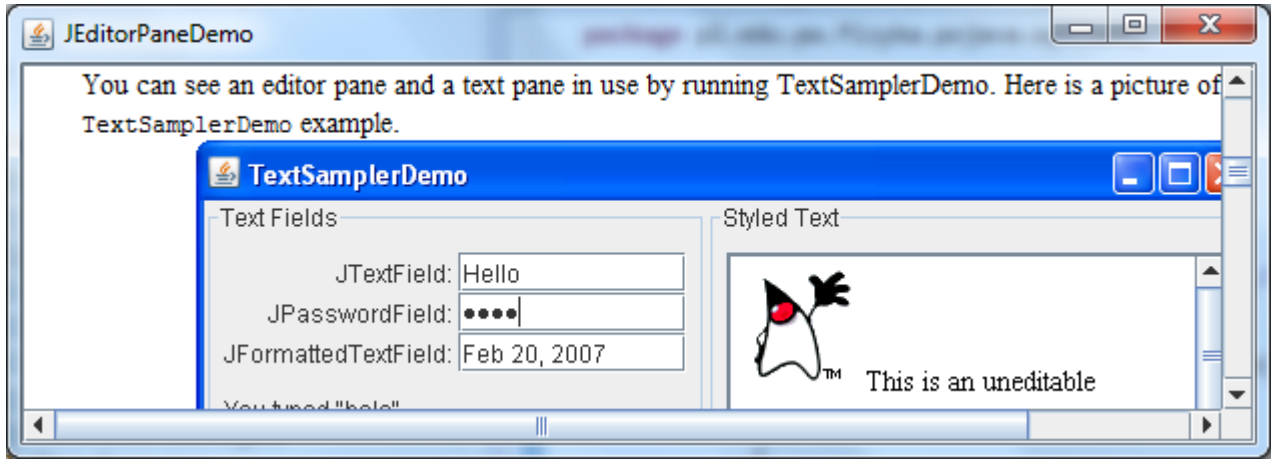
JTabbedPaneDemo.java

```
JTabbedPane tabbedPane = new JTabbedPane();

JPanel panel1 = new JPanel(new FlowLayout());
JPanel panel2 = new JPanel(new FlowLayout());
JPanel panel3 = new JPanel(new FlowLayout());

tabbedPane.addTab("Tytul 1", panel1);
tabbedPane.addTab("Tytul 2", panel2);
tabbedPane.addTab("Tytul 3", panel3);
add(tabbedPane);
// tabbedPane.remove(panel2);
```

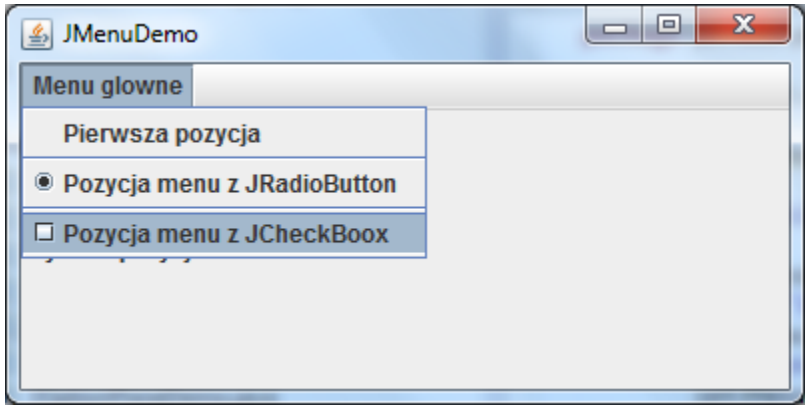


JEditorPaneDemo.java

```
JEditorPane edytor = new JEditorPane();
edytor.setEditable(false);
    try
    {   URL link = new   URL("http://jakasstrona.html");
        edytor.setPage(link);
    }
    catch(IOException e)
    {
        edytor.setText("Wyjatek:"+e);
    }

add(new JScrollPane(edytor));
```

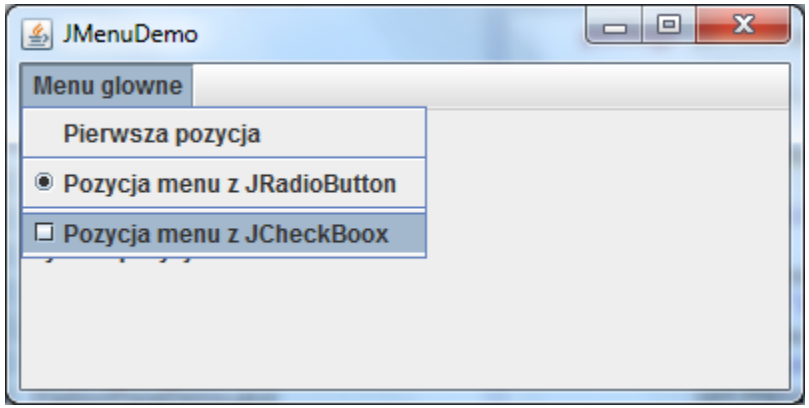


JMenuDemo.java




```
JMenuBar menuBar;
JMenu menu;

JMenuItem menuItem;
final JRadioButtonMenuItem rbMenuItem;
final JCheckBoxMenuItem cbMenuItem
// Tworzenie paska menu
menuBar = new JMenuBar();
//Dodawanie menu:
menu = new JMenu("Menu glowne");
menuBar.add(menu);

menuItem = new JMenuItem("Pierwsza pozycja");
menu.add(menuItem);

rbMenuItem = new JRadioButtonMenuItem("Pozycja menu z JRadioButton");
rbMenuItem.setSelected(true);
menu.add(rbMenuItem);

cbMenuItem = new JCheckBoxMenuItem("Pozycja menu z JCheckBoox");
menu.add(cbMenuItem);

setJMenuBar(menuBar);
```

JMenuDemo.java




```
JMenuBar menuBar;
JMenu menu;

JMenuItem menuItem;
final JRadioButtonMenuItem rbMenuItem;
final JCheckBoxMenuItem cbMenuItem
// Tworzenie paska menu
menuBar = new JMenuBar();
//Dodawanie menu:
menu = new JMenu("Menu glowne");
menuBar.add(menu);

menuItem = new JMenuItem("Pierwsza pozycja");
menu.add(menuItem);

rbMenuItem = new JRadioButtonMenuItem("Pozycja menu z JRadioButton");
rbMenuItem.setSelected(true);
menu.add(rbMenuItem);

cbMenuItem = new JCheckBoxMenuItem("Pozycja menu z JCheckBoox");
menu.add(cbMenuItem);

setJMenuBar(menuBar);
```

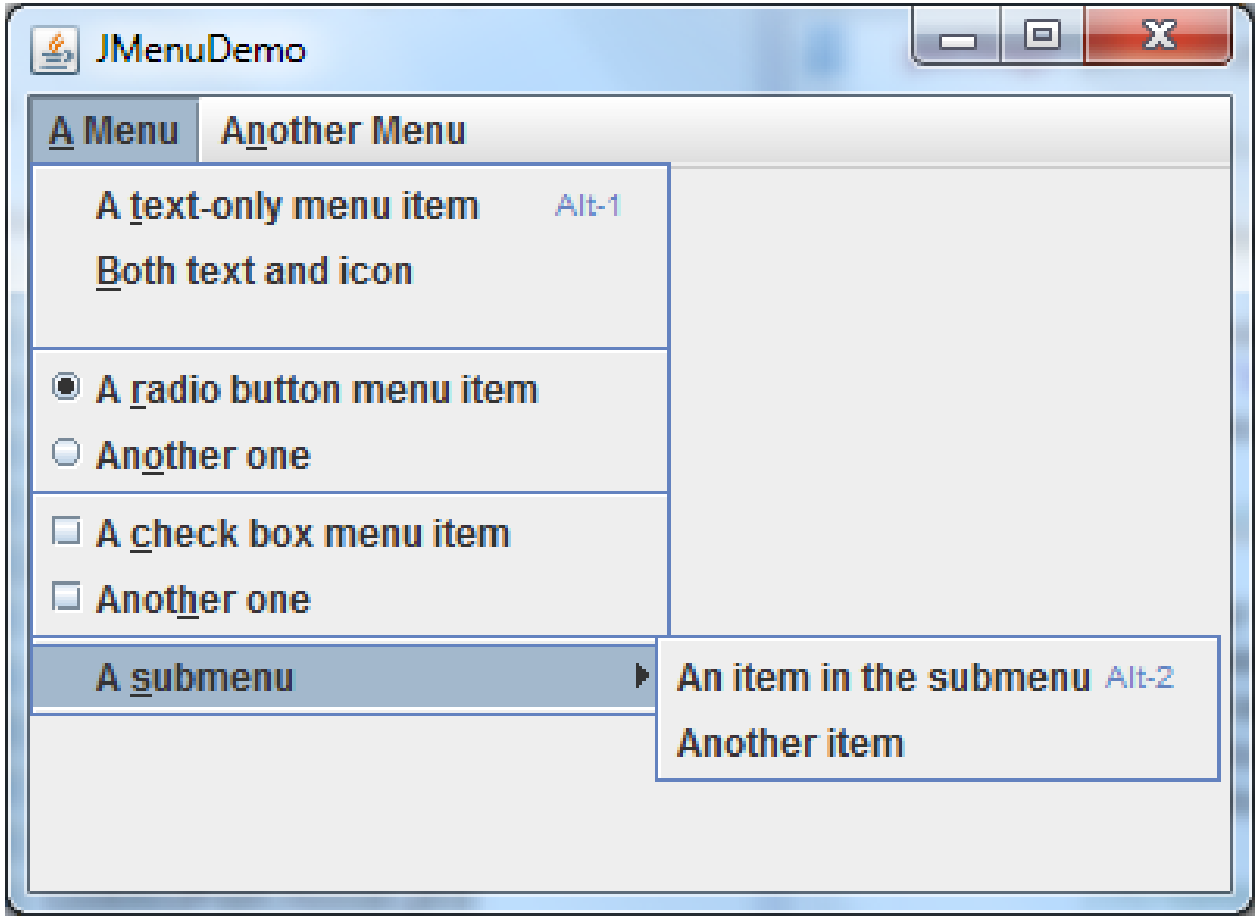
JMenuDemo.java

```
// dodawanie akcji do elementow menu analogicznie jak dla komponentow:

menuItem.addActionListener(new ActionListener() {
   // Przyklad wykorzystania html w etykiecie Jlabel
   // niemal każdy komponent Swing wyświetlający String można formatować html
     public void actionPerformed(ActionEvent e) {
            etykieta.setText(
                    "<html>\n" +
                    "Wybrano pierwsza pozycje menu " +
                   "<ul>\n" +
                   "<li><font color=red>Stan JRadioButton'a: </font>\n" +
                    rbMenuItem.isSelected() +
                     "<li><font color=blue>Stan JCheckBox'a: </font>\n" +
                    cbMenuItem.isSelected()  +
                   "</ul>\n"+
                     "W tej etykiecie JLabel wykorzystano kod HTML"+
                     "</html>");
                    }
            });
```

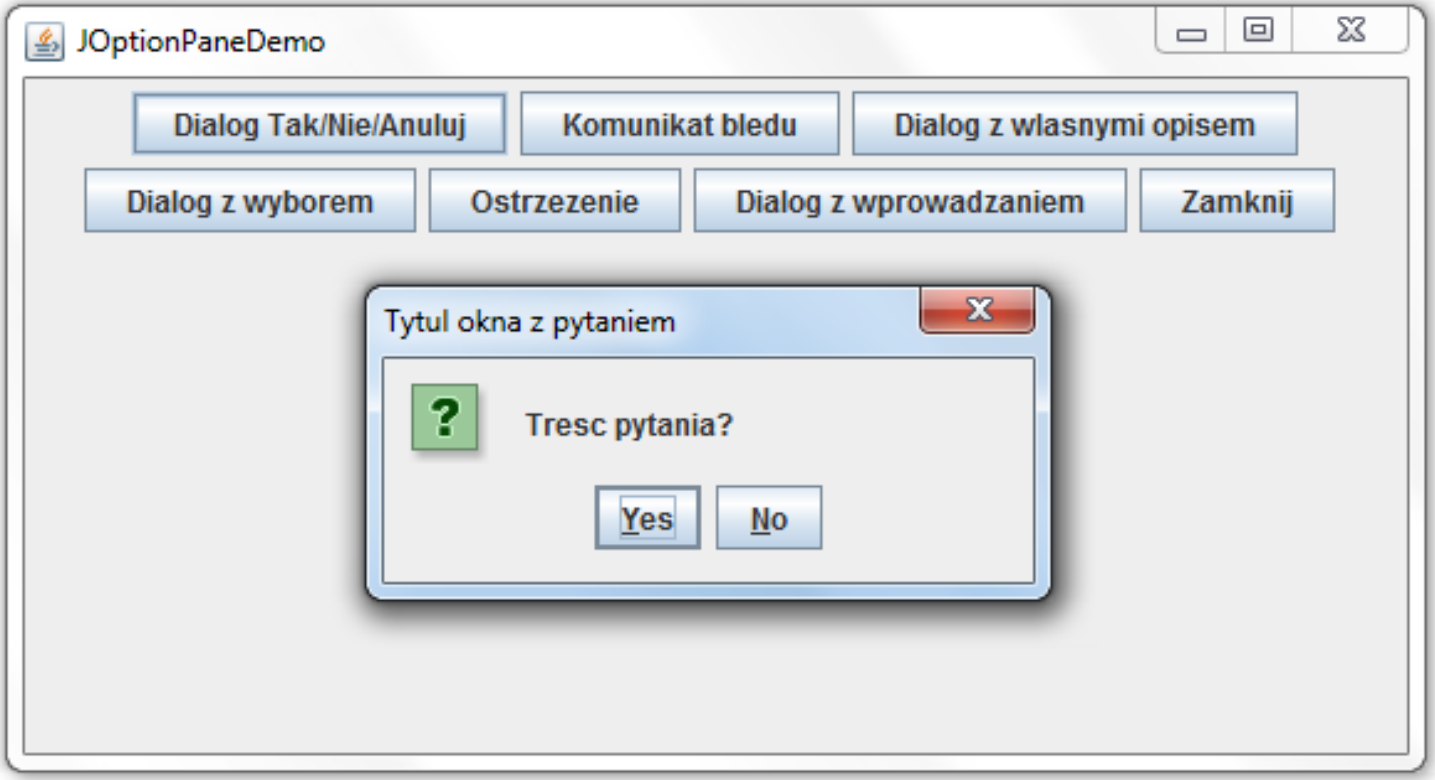
JMenuDemo.java

```
rbMenuItem.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
                etykieta.setText( "Zmieniono stan JRadioButton'a: " +
                                rbMenuItem.isSelected() );
        }
});



cbMenuItem.addItemListener(new ItemListener() {
        @Override
        public void itemStateChanged(ItemEvent e) {
                etykieta.setText( "Zmieniono stan JCheckBox'a: " +
                        cbMenuItem.isSelected() );
        }

});
```

JMenuDemo2.java

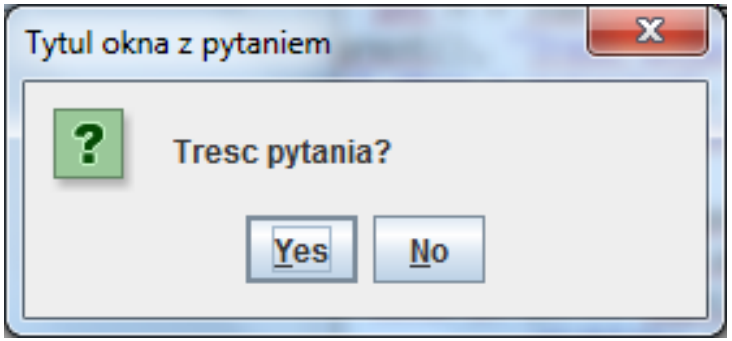


JOptionPaneDemo.java



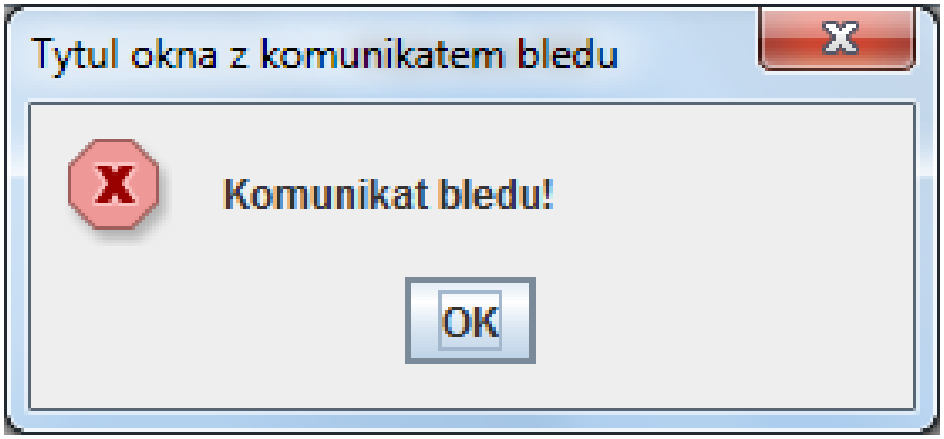
JOptionPaneDemo.java

```
b1.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                    int n = JOptionPane.showConfirmDialog(
                getParent(), "Tresc pytania?",
                "Tytul okna z pytaniem",
                JOptionPane.YES_NO_OPTION);

                    if (n == JOptionPane.YES_OPTION) {
                            etykieta.setText("Wybrano TAK");}
                    else if (n == JOptionPane.NO_OPTION) {
                            etykieta.setText("Wybrano NIE");
    }
                    else {
            etykieta.setText("Nic nie wybrano - zamknieto okno dialogu");
    }
             }
    });
```



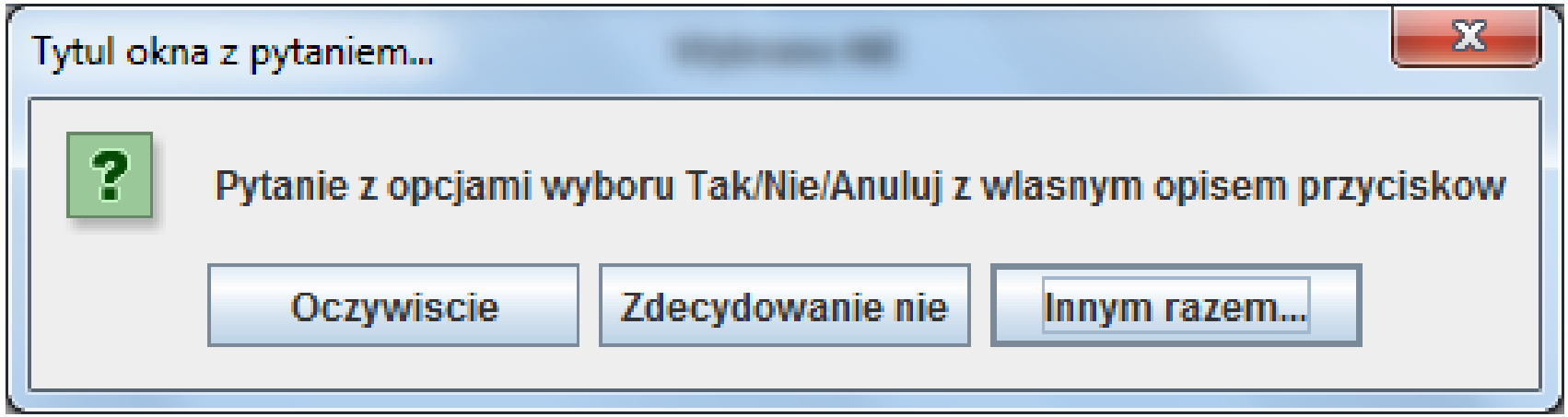
JOptionPaneDemo.java

```
b2.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {
                    JOptionPane.showMessageDialog(
                                f, "Komunikat bledu!",
                                "Tytul okna z komunikatem bledu",
                                JOptionPane.ERROR_MESSAGE);

        }
 });
```



JOptionPaneDemo.java

```
Object[] options = {"Oczywiscie", Zdecydowanie nie", "Innym razem..."};
int n = JOptionPane.showOptionDialog(f,
          "Pytanie z opcjami wyboru Tak/Nie/Anuluj "
          + "z wlasnym opisem przyciskow",
          "Tytul okna z pytaniem...",
           JOptionPane.YES_NO_CANCEL_OPTION,
           JOptionPane.QUESTION_MESSAGE,
           null,
           options,
           options[2]);
```

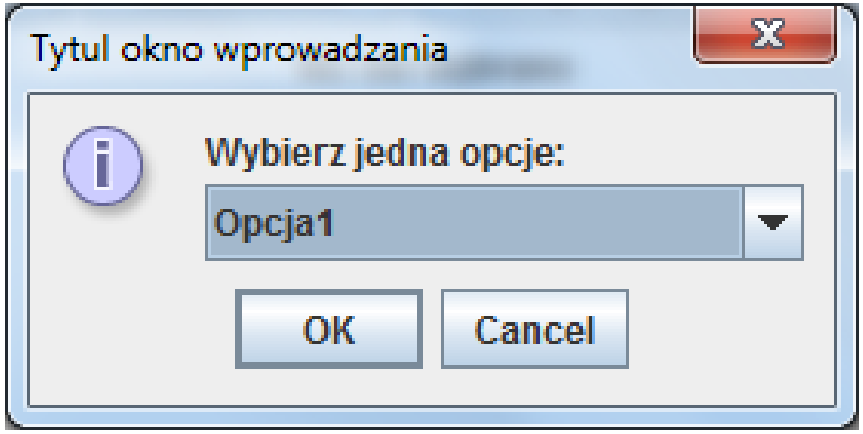


JOptionPaneDemo.java

```
if (n == JOptionPane.YES_OPTION) {
                etykieta.setText("Wybrano: Oczywiscie");
                } else if (n == JOptionPane.NO_OPTION) {
                etykieta.setText("Wybreano: Zdecydowanie nie");
                } else if (n == JOptionPane.CANCEL_OPTION) {
                etykieta.setText("Wybrano: Innym razem...");
                } else {
                etykieta.setText("Nic nie wybrano");
                }
```

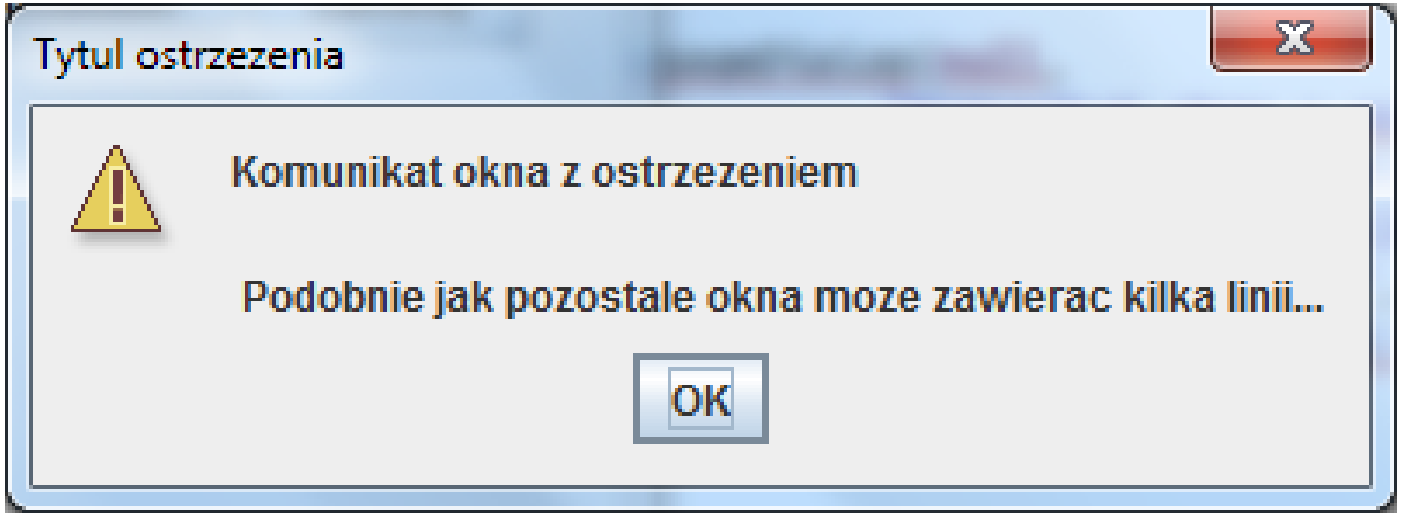
JOptionPaneDemo.java

```
Object[] wartosciWyboru = { "Opcja1", "Opcja2", "Opcja3" };

Object selectedValue = JOptionPane.showInputDialog(
                                    JOptionPaneDemo.this,
                                    "Wybierz jedna opcje:",
                                    "Tytul okno wprowadzania",
                                    JOptionPane.INFORMATION_MESSAGE,
                                    null,
                                    wartosciWyboru,
                                    wartosciWyboru[0]);

etykieta.setText((String)selectedValue);
```



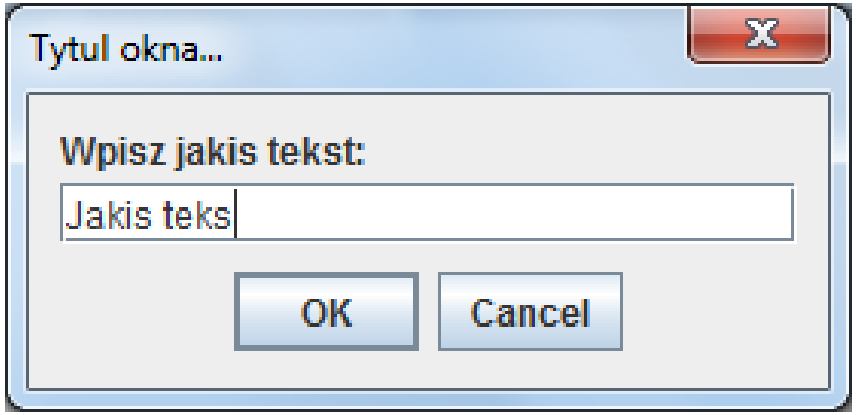
JOptionPaneDemo.java

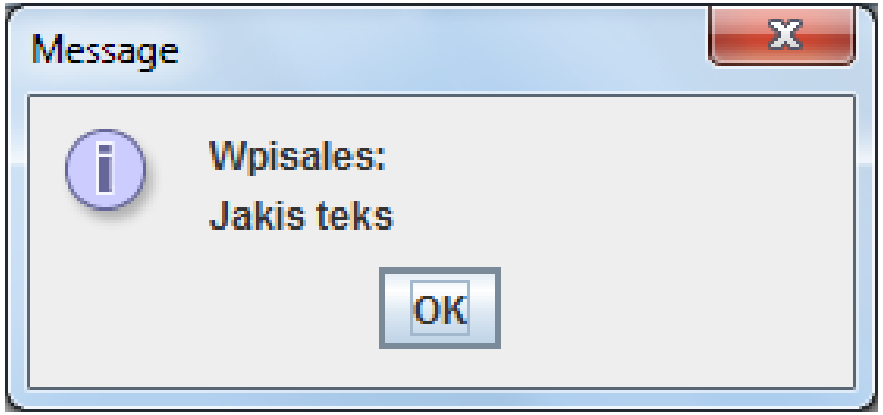
```
JOptionPane.showMessageDialog(null,
                              "Komunikat okna z ostrzezeniem" +
                               "\n\n Podobnie jak pozostale okna" +
                               " moze zawierac kilka linii... " ,
                              "Tytul ostrzezenia",
                              JOptionPane.WARNING_MESSAGE);
```



JOptionPaneDemo.java

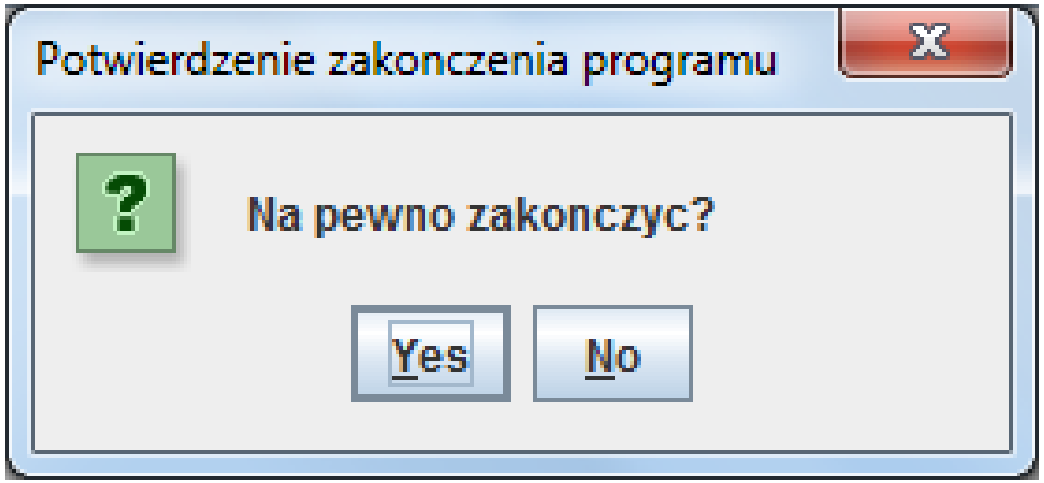
```
String s = (String)JOptionPane.showInputDialog(
                    f,
                    "Wpisz jakis tekst:\n",
                    "Tytul okna...",
                    JOptionPane.PLAIN_MESSAGE,
                    null,
                    null,
                    "tekst domyslny");

JOptionPane.showMessageDialog(f, "Wpisales: \n" + s);
```





JOptionPaneDemo.java

```
int n = JOptionPane.showConfirmDialog(
                    f, "Na pewno zakonczyc?",
                    "Potwierdzenie zakonczenia programu",
                    JOptionPane.YES_NO_OPTION);

if (n == JOptionPane.YES_OPTION)
                            dispose();
```

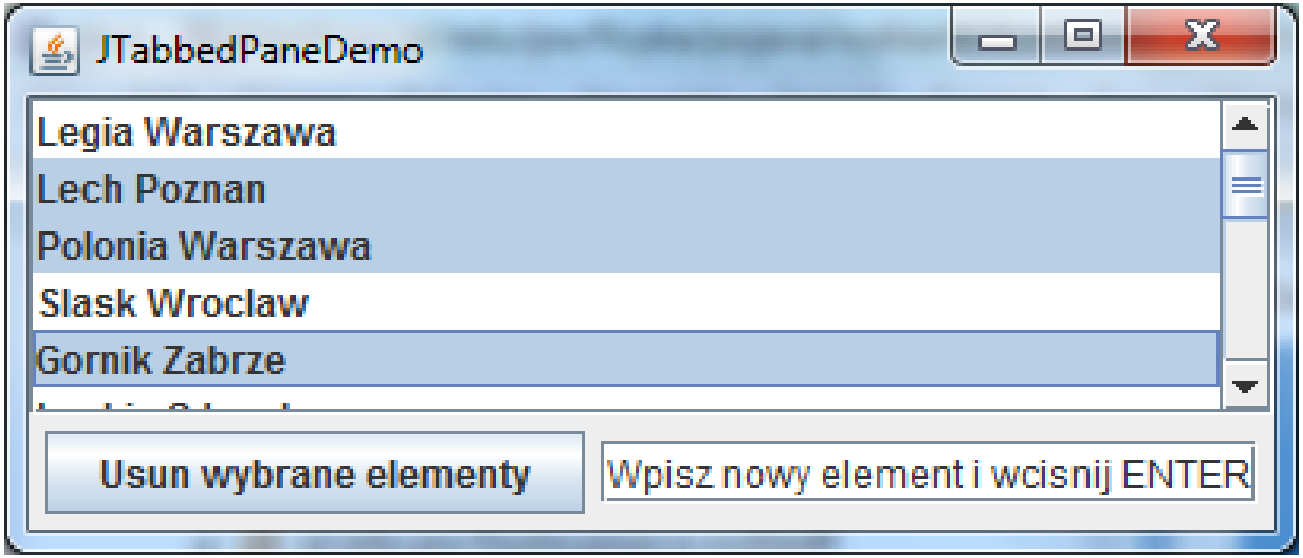


JListDemo.java

```
String[] nazwyDruzyn = { "Legia Warszawa",  "Lech Poznan",  "Polonia Warszawa",
                         "Slask Wroclaw",  "Gornik Zabrze",  "Lechia Gdansk",
                        "Zaglebie Lubin",  "Piast Gliwice",  "Wisla Krakow",
                        "Jagiellonia Bialystok", "Korona Kielce",
                         "Widzew Lodz", "Pogon Szczecin", "Ruch Chorzow",
                         "Podbeskidzie Bielsko-Biala", "GKS Belchatow"};

        DefaultListModel listaElementy = new DefaultListModel();
        JList lista = new JList(listaElementy);


        for (int i=0; i < nazwyDruzyn.length; i++)
                listaElementy.addElement(nazwyDruzyn[i]);

        lista.addListSelectionListener(listaListener);
        lista.setVisibleRowCount(5);

         //Dodawanie JScrollPane:
         JScrollPane listScrollPane = new JScrollPane(lista);
         listScrollPane.setPreferredSize( new Dimension(300,100));

        //add(lista, BorderLayout.CENTER);
         add(listScrollPane, BorderLayout.CENTER);
```

JListDemo.java

```
ListSelectionListener listaListener = new ListSelectionListener() {

        public void valueChanged(ListSelectionEvent e) {
                if (e.getValueIsAdjusting()) return;
                System.out.println("Zaznaczone elementy listy:");
                for (Object wybrane : lista.getSelectedValuesList())
                        System.out.println(wybrane);
                }
};
```



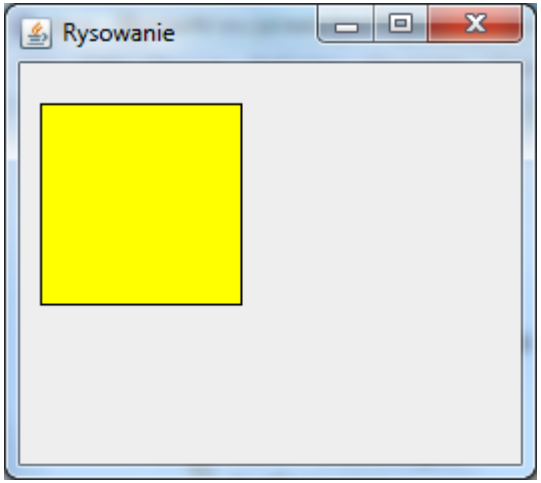
JListDemo.java

```
JButton usunElementy = new JButton("Usun wybrane elementy");
JTextField poleDodawania = new JTextField("Wpisz nowy element i wcisnij ENTER");

usunElementy.addActionListener(usuwanieElementow);
poleDodawania.addActionListener(dodawanieElementow);

ActionListener usuwanieElementow = new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                for (Object wybrane : lista.getSelectedValuesList())
                        listaElementy.removeElement(wybrane);
        }
};

ActionListener dodawanieElementow = new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                listaElementy.addElement(poleDodawania.getText());
        }
};
```

Rysowanie.java

Każdy komponent posiada metodę paintComponent(Graphics g), którą można zmodyfikować w klasach pochodnych, np.:

```
class MyPanel extends JPanel {
    private int squareX = 10;
    private int squareY = 20;
    private int squareW = 100;
    private int squareH = 100;

     public Dimension getPreferredSize() {
             return new Dimension(250,200);
     }


    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.YELLOW);
        g.fillRect(squareX,squareY,squareW,squareH);
        g.setColor(Color.BLACK);
        g.drawRect(squareX,squareY,squareW,squareH);
    }
}
```



Rysowanie.java

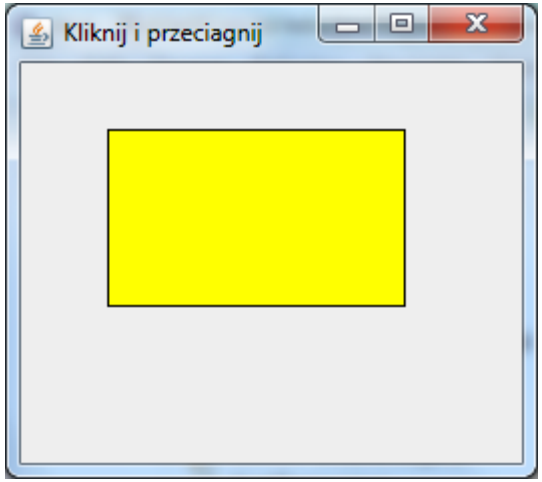
```
public Dimension getPreferredSize() {
        return new Dimension(250,200);
}
```

- Metoda getPreferredSize() została przedefioniowana, żeby ustalać preferowany rozmiar tworzonego panelu rysowania (wykorzystywane przez niektórych zarządców rozmieszczenia komponentów – l*ayout managers*)

RysowanieMysz.java

- Dodając do tak stworzonego panelu interfejs obsługi zdarzeń myszy można rysować prostokąty o rozmiarach definiowanych myszą

```
public MyPanel() {
    addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {
            squareX = e.getX();
            squareY = e.getY();
             repaint();
        }
     }
 }
```



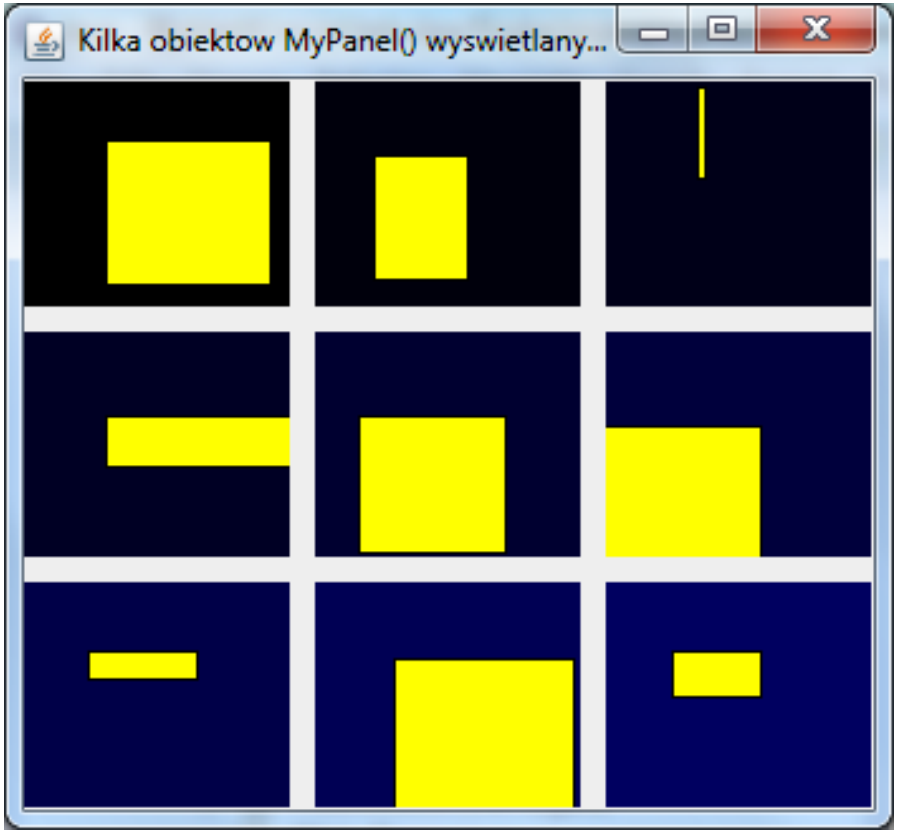
RysowanieMysz.java

- Dodając obsługę „puszczenia” klawisza myszy można definiować rozmiar komponentu:

```
addMouseListener(new MouseAdapter() {
    public void mousePressed(MouseEvent e) {
        squareX = e.getX();
        squareY = e.getY();
    }

    public void mouseReleased(MouseEvent e) {
        if (e.getX() > squareX) squareW = e.getX() - squareX;
                else {
                        squareW = squareX - e.getX();
                        squareX = e.getX();
                    }
        if (e.getY() > squareY) squareH = e.getY() - squareY;
                        else {
                        squareH = squareY - e.getY();
                        squareY = e.getY();
                        }
                    repaint();
                }
});
```

RysowanieMysz2.java

- Tak utworzony panel z opcją rysowania przy pomocy myszy jest traktowany jak każdy inny komponent Swing, może być wielokrotnie wykorzystany:



RysowanieMysz2.java

```
        JFrame f = new JFrame("Kilka obiektow MyPanel() wyswietlanych
jednoczesnie");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout(new GridLayout(3,3,10,10));

        MyPanel[] panele = new MyPanel[9];

        for (int i=0; i<9; i++){
        panele[i] = new MyPanel();
        panele[i].setBackground(new Color(i*12));
        f.add(panele[i]);
        }

        f.pack();
```

RysowanieMysz3.java

Poprzedni przykład można zmodyfikować tak, aby rysowane było kilka elementów, których współrzędne są przechowywane w tablicach:

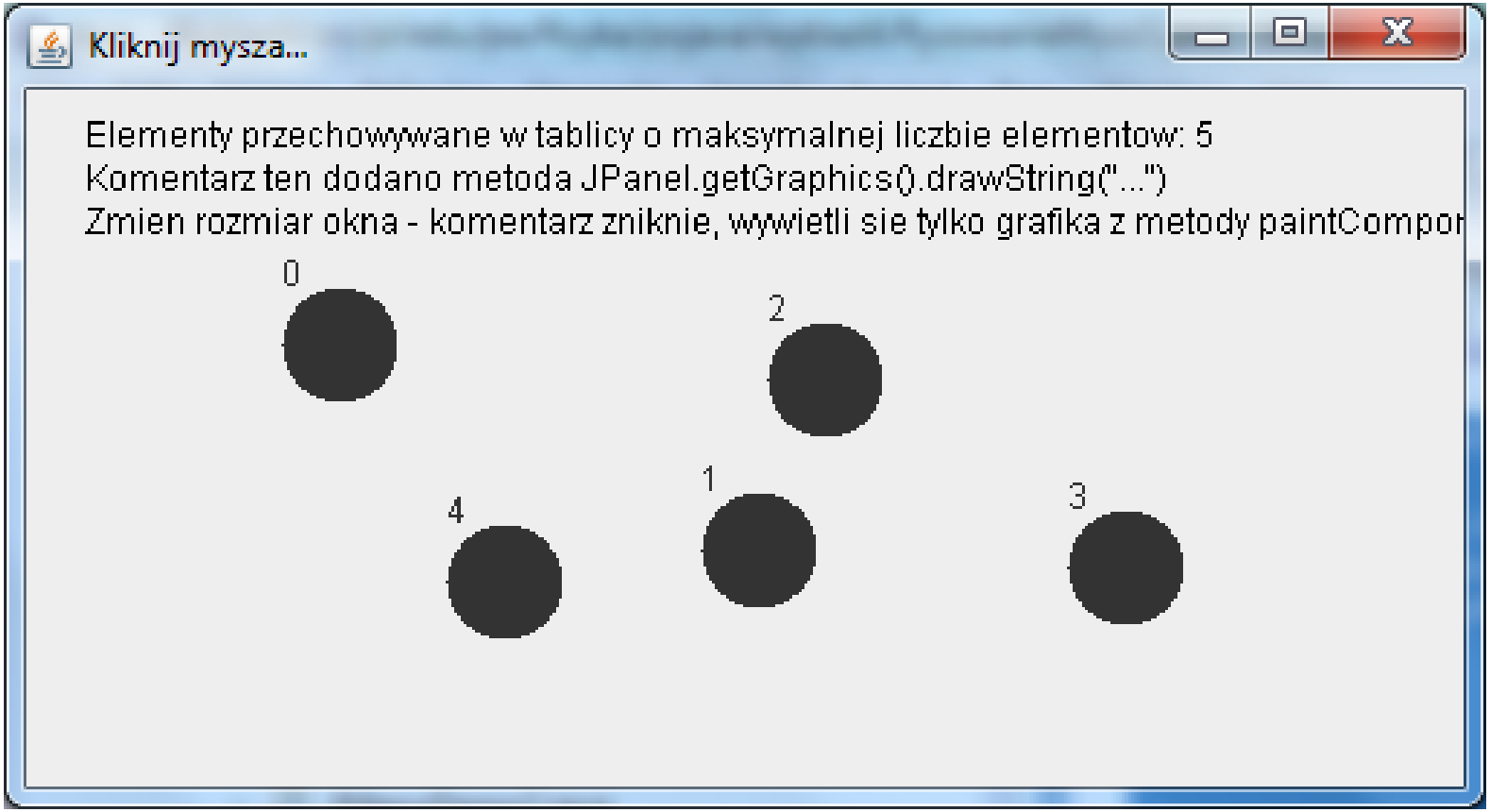
```
class MyPanel3 extends JPanel {
    private int MAKSYMALNA_LICZBA_ELEMENTOW = 5;
        private int[] x = new int [MAKSYMALNA_LICZBA_ELEMENTOW];
        private int[] y = new int [MAKSYMALNA_LICZBA_ELEMENTOW];
        private int licznikKlikniec = 0;

   // (... konstrukor i inne metody...)

    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
       for (int i = 0; i<licznikKlikniec; i++){
            g.fillOval(x[i], y[i], 40, 40);
            g.drawString(""+i, x[i], y[i]);
        }
    }

}
```

RysowanieMysz3.java

Poprzedni przykład można zmodyfikować tak, aby rysowane było kilka elementów, których współrzędne są przechowywane w tablicach:

```
public MyPanel3() {

        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                if (licznikKlikniec < MAKSYMALNA_LICZBA_ELEMENTOW){
                        x[licznikKlikniec] = e.getX();
                        y[licznikKlikniec] = e.getY();
                        licznikKlikniec++;
                        repaint();
                }
                else{
                        //Przekroczona dopuszczalna ilosc klikniec..
                    }

            if (e.getButton() == MouseEvent.BUTTON3) wyczyscElementy();
                //obsluga prawego klawisza myszy
            }
         });
 }
```

RysowanieMysz3.java



Podejście obiektowe…

Stwórzmy klasę Prostokąt, która będzie miała zdefiniowane rozmiary i położenie i „sama” będzie odpowiedzialna za rysowanie

```
public class Prostokat {
        int xPos = 0, yPos = 0;
        int width = 0, height = 0;
        Color color = Color.BLACK;

        public void paint(Graphics g){
        g.setColor(color);
        g.fillRect(xPos,yPos,width,height);
    }
}
```

Metoda paint(Graphics g) otrzyma referencje do obiektu Graphics pewnego komponentu, na który będzie „nadrysowany” prostokąt.

Podejście obiektowe…

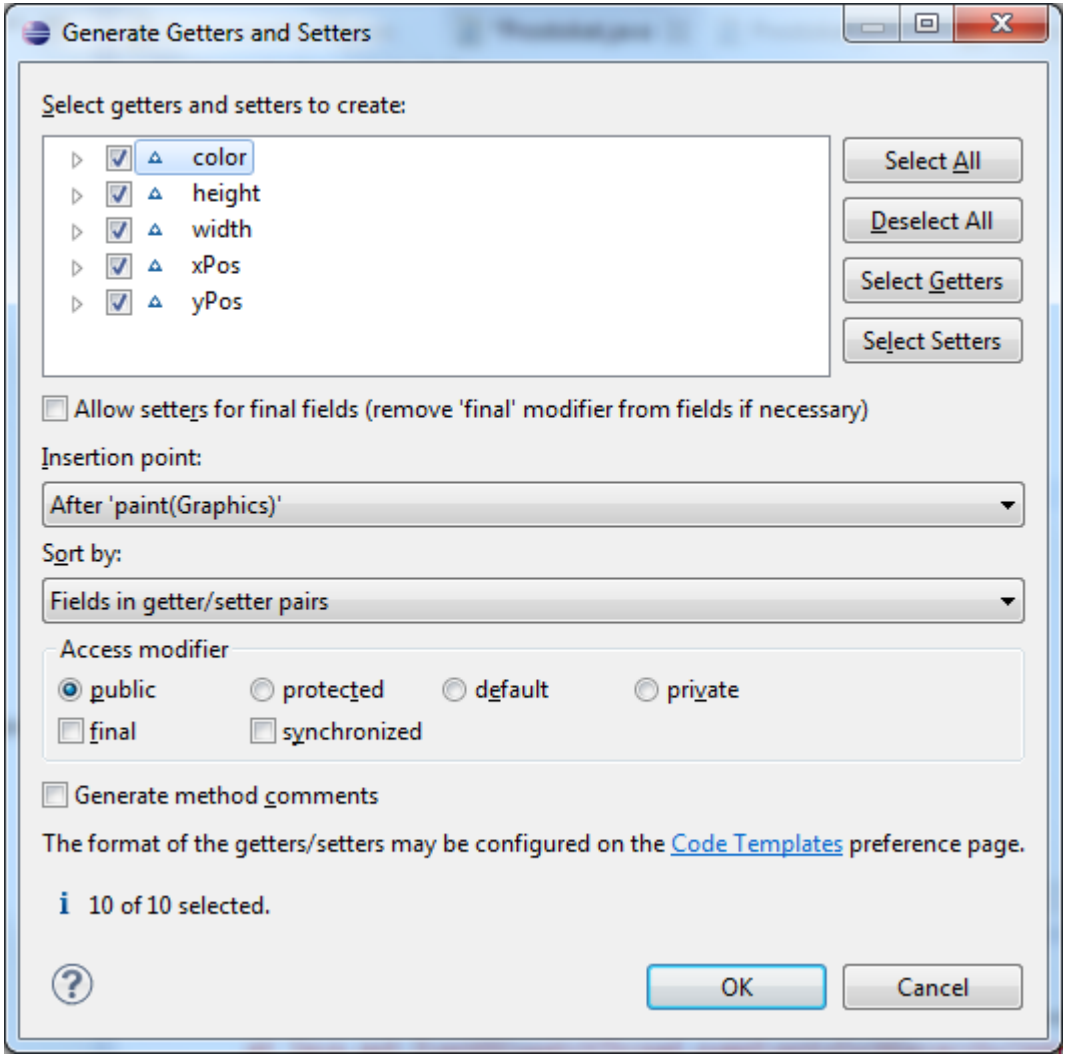
```
public class Prostokat {
        //(…) deklaracja pól

        public void paint(Graphics g){
        g.setColor(color);
        g.fillRect(xPos,yPos,width,height);
    }
}
```

Przykład tworzenia i rysowania obiektu Prostokat w innym komponencie:

```
public void paintComponent(Graphics g) {
    super.paintComponent(g);
      Prostokat prostokat = new Prostokat();
            prostokat.setX( e.getX() );
            prostokat.setY( e.getY() );
            prostokat.setWidth(50);
            prostokat.setHeight(50);
            prostokat.paint(g);
}
```

Podejście obiektowe…

Dodajmy do klasy Prostokąt metody ustawiające i pobierające wartości pól – w Eclipse: menu Source a potem : „*Generate getters and setters*”:



Podejście obiektowe…

Dodajmy do klasy Prostokąt metody ustawiające i pobierające wartości pól – w Eclipse: menu Source a potem : „*Generate getters and setters*”:

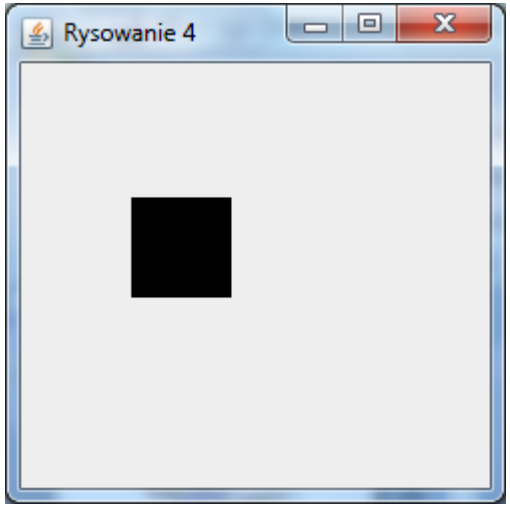
Dodane zostały automatycznie metody:

```
public int getxPos() {
        return xPos;
}
public void setxPos(int xPos) {
        this.xPos = xPos;
}
public int getyPos() {
        return yPos;
}
public void setyPos(int yPos) {
        this.yPos = yPos;
}
public int getWidth() {
        return width;
}
```

```
public void setWidth(int width) {
        this.width = width;
}
public int getHeight() {
        return height;
}
public void setHeight(int height) {
        this.height = height;
}
public Color getColor() {
        return color;
}
public void setColor(Color color) {
        this.color = color;
}
```

RysowanieMysz4.java

```
class MyPanel4 extends JPanel {
    Prostokat prostokat = new Prostokat();
    public MyPanel4() {
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent e) {
                prostokat.setX( e.getX() );
                prostokat.setY( e.getY() );
                prostokat.setWidth(50);
                prostokat.setHeight(50);
                  repaint();
                        }
        });
    }
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        prostokat.paint(g);
    }
}
```



RysowanieMysz5.java

```
addMouseListener(new MouseAdapter() {
        public void mousePressed(MouseEvent e) {

            if (e.getButton() == MouseEvent.BUTTON3) licznikKlikniec=0;

            tablicaProstokatow[licznikKlikniec] = new Prostokat();

            tablicaProstokatow[licznikKlikniec].setX( e.getX() );
            tablicaProstokatow[licznikKlikniec].setY( e.getY() );
            tablicaProstokatow[licznikKlikniec].setWidth(50);
            tablicaProstokatow[licznikKlikniec].setHeight(50);

            tablicaProstokatow[licznikKlikniec].setColor(
 new Color((float)Math.random(), (float)Math.random(),(float)Math.random()));

            licznikKlikniec++;
            repaint();

        }
 });
```

RysowanieMysz5.java

Można utworzyć tablicę elementów klasy Prostokąt, do której po kliknięciu myszy dodawany byłby nowy element:

```
class MyPanel5 extends JPanel {
    private int MAKSYMALNA_LICZBA_ELEMENTOW = 5;
    Prostokat[] tablicaProstokatow =
                          new Prostokat[MAKSYMALNA_LICZBA_ELEMENTOW];
    private int licznikKlikniec = 0;

    // (obsluga zdarzen myszy....)

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        for(int i = 0; i<licznikKlikniec; i++){
                tablicaProstokatow[i].paint(g);
        }

    }
}
```

RysowanieMysz5.java

Dodawanie obiektów do tablicy ma pewną istotną wadę – trzeba znać z góry maksymalną dopuszczalną liczbę elementów.

Ograniczenie to można ominąć stosując tzw. Kolekcje, o których na następnym wykładzie…