

## Sprawozdanie 3 z Programowania obiektowego Karol Szatan, gr.4, Air

8.12.2015r

```
import java.applet.Applet;
import java.awt.Button;
import java.awt.Choice;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.awt.Color;

public class Paint extends Applet implements MouseListener, MouseMotionListener,
ActionListener
{
    //definiowanie potrzebnych zmiennych
    private int r ,a ,b,c; //zmienne pomocnicze
    private int x , y; //zmienne okreslajace pozozenie kursora
    private String wyznacznik; //zmienna przechowujaca tekst wyboru
    private int HEIGHT , WIDTH ; //zmienne przechowujace wymiary appletu

    //tworzymy zmienna przycisku typu Button
    private Button olowek;
    private Button pedzel;
    private Button linia;
    private Button okrag;
    private Button gumka;
    private Button wyczysc;

    //zmienne przycisku typu Choice
    private Choice kolor;
    private Choice rozmiar;

    //zmienna typu Color przechowujaca kolor
    private Color kolory;

    //główny program uruchamiajacy się od razu po starcie appletu
    public void init() {
        //Inicjalizacja potrzebnych metod
        super.init();
        this.addMouseListener(this);
        this.addMouseMotionListener(this);
        c=1;//pomoznicza zmienna
        //rozmiar panelu
        HEIGHT=500;
        WIDTH=800;
        //rozmiar przycisków
        x=60;
        y=20;
        //przypisanie rozmiaru appletu
```

```

this.setSize(WIDTH,HEIGHT);
//przypisanie początkowego narzędzia po starcie programu
wyznacznik="ołówek";
//dodanie kolorów do panelu
kolor =new Choice();
kolor.add("Czarny");
kolor.add("Czerwony");
kolor.add("Niebieski");
kolor.add("Zielony");
kolor.add("Zolty");
kolor.add("Bialy");
//dodanie rozmiaru do panelu
rozmiar=new Choice();
rozmiar.add("5");
rozmiar.add("7");
rozmiar.add("10");
rozmiar.add("12");
rozmiar.add("15");
rozmiar.add("20");
rozmiar.add("30");
rozmiar.add("35");
rozmiar.add("40");
rozmiar.add("50");
rozmiar.add("80");
//wybranie domyślnego rozmiaru
rozmiar.select(6);
//menedzer rozmieszczania przyciskow
this.setLayout(new FlowLayout());
setLayout(null); // wyzerowanie Layout Managera dzięki czemu będziemy mogli
recznie ustawić komponenty
//dodanie przyciskow typu choice z wyborem koloru i rozmiaru
this.add(kolor);
this.add(rozmiar);
//utworzenie przyciskow do narzędzi
linia = new Button("Linia");
this.add(linia);
linia.addActionListener(this);
okrag = new Button("Okrag");
this.add(okrag);
okrag.addActionListener(this);
ołówek = new Button("Ołówek");
this.add(ołówek);
ołówek.addActionListener(this);
pedzel = new Button("Pedzel");
this.add(pedzel);
pedzel.addActionListener(this);
gumka = new Button("Gumka");
this.add(gumka);
gumka.addActionListener(this);
wyczysc = new Button("Wyczysc");
this.add(wyczysc);
wyczysc.addActionListener(this);
//rozmiar przycisków i ich położenie
kolor.setBounds(0, 20, x, y);
rozmiar.setBounds(60,20,x,y);
linia.setBounds(120,20,x,y);
okrag.setBounds(180,20,x,y);
ołówek.setBounds(240,20,x,y);
pedzel.setBounds(300,20,x,y);

```

```

        gumka.setBounds(360,20,x,y);
        wyczysc.setBounds(420,20,x,y);
    }
    //opisanie panela menu
    public void paint(Graphics g) {
        g.drawString("Kolory", 0, 15);
        g.drawString("Rozmiar", 60, 15);
        g.drawString("Narzędzia", 120, 15);
    }

    @Override
    public void mouseDragged(MouseEvent arg0) {
        // TODO Auto-generated method stub
        System.out.println("MouseDragged: ");
        //stworzenie obiektu klasy Graphics
        Graphics g = getGraphics();
        // przypisanie odpowiedniego koloru
        wyborkoloru();
        g.setColor(kolory);
        // pobiera rozmiar z przycisku rozmiar i przypisuje do zmiennej
        r = Math.round(Float.valueOf(rozmiar.getSelectedItem()));
        // sprawdzenie jaka wartosc ma wyznacznik i przypisanie odpowiedniego
        narzedzia
        switch (wyznacznik)
        {
            //rysowanie okregu o promieniu r i srodku w punkcie klikniecia myszka
            case "okrag": { g.drawOval(arg0.getX()-r/2, arg0.getY()-r/2, r, r);
break; }

            //rysowanie lini z poczatkiem i koncem podanymi dwoma kliknieciami
            case "linia" : { break; }
            //rysowanie olowkiem, czyli punktem
            case "olowek": { g.drawLine(a, b, arg0.getX(), arg0.getY());
a=arg0.getX(); b=arg0.getY(); break; }
            //rysowanie pedzlem czyli kole o promieniu r
            case "pedzel": { g.fillOval(arg0.getX()-r/2, arg0.getY()-r/2, r, r);
break; }

            //gumowanie, gumka jest kwadratem
            case "gumka": { g.clearRect(arg0.getX()-r/2, arg0.getY()-r/2, r, r);
break; }

        }
    }

    @Override
    public void mouseMoved(MouseEvent arg0) {
        // TODO Auto-generated method stub
        System.out.println("MouseMoved");
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("MouseClicked " +e.getX() );
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("mouseEntered");
    }

    @Override
    public void mouseExited(MouseEvent e) {

```

```

        // TODO Auto-generated method stub
    }
    @Override
    public void mousePressed(MouseEvent e) {
        // TODO Auto-generated method stub
        //stworzenie obiektu klasy Graphics
        Graphics g = getGraphics();
        // przypisanie odpowiedniego koloru
        wyborkoloru();
        g.setColor(kolory);
        // pobiera rozmiar z przycisku rozmiar i przypisuje do zmiennej
        r = Math.round(Float.valueOf(rozmiar.getSelectedItem()));
        // sprawdzenie jaka wartosc ma wyznacznik i przypisanie odpowiedniego narzedzia
        switch(wyznacznik) {
            case "linia": {
                //rysowanie lini podzielone jest na dwa etapy, plus trzeci jako
                przygotowanie na powtorzenie akcji
                if(c==0)
                    { g.drawLine(a,b, e.getX(), e.getY()); c=2; }
                if(c==1)
                    { a=e.getX(); b=e.getY(); c=0; }
                if(c==2)
                    { c=1; }
                break;
            }
            case "kolo":
                { g.drawOval(e.getX()-r/2, e.getY()-r/2, r, r); break; }
            case "gumka":
                { g.clearRect(e.getX()-r/2,e.getY()-r/2,r,r); break; }
            case "olowek":
                { a=e.getX(); b=e.getY(); g.fillOval(e.getX()-1, e.getY()-1, 2,
2); break; }
            case "pedzel":
                { g.fillOval(e.getX()-r/2, e.getY()-r/2, r, r); break; }
        }
    }
    @Override
    public void mouseReleased(MouseEvent e) {
        // TODO Auto-generated method stub
        System.out.println("mouseReleased"+e.getX());
    }
    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        Object sprawdz = arg0.getSource();// tworzymy obiekt przechowujacy dane z
        pobranego argumentu ActionEvent
        //zmiana rodzaju narzedzia w zaleznosci od kliknietego przycisku

        if(sprawdz==wyczysc)
            { this.getGraphics().clearRect(0, 0, getWidth(), getHeight()); }
        if(sprawdz==okrag)
            { wyznacznik="okrag"; }
        if(sprawdz==linia)
            { wyznacznik="linia"; }
        if(sprawdz==olowek)
            { wyznacznik="olowek"; }
        if(sprawdz==pedzel)
            { wyznacznik="pedzel"; }
    }

```

```

        if(sprawdz==gumka)
            { wyznacznik="gumka"; }
    }

    //sprawdzenie jaki kolor wybraliśmy i przypisanie go do zmiennej koloru
    public void wyborkoloru() {
        switch (kolor.getSelectedItem()) {
            case "Czarny":
                { kolory=Color.black; break; }
            case "Czerwony":
                { kolory=Color.red; break; }
            case "Niebieski":
                { kolory=Color.blue; break; }
            case "Zielony":
                { kolory=Color.green; break; }
            case "Zolty":
                { kolory=Color.yellow; break; }
            case "Bialy":
                { kolory=Color.white; break; }
        }
    }
}

```

