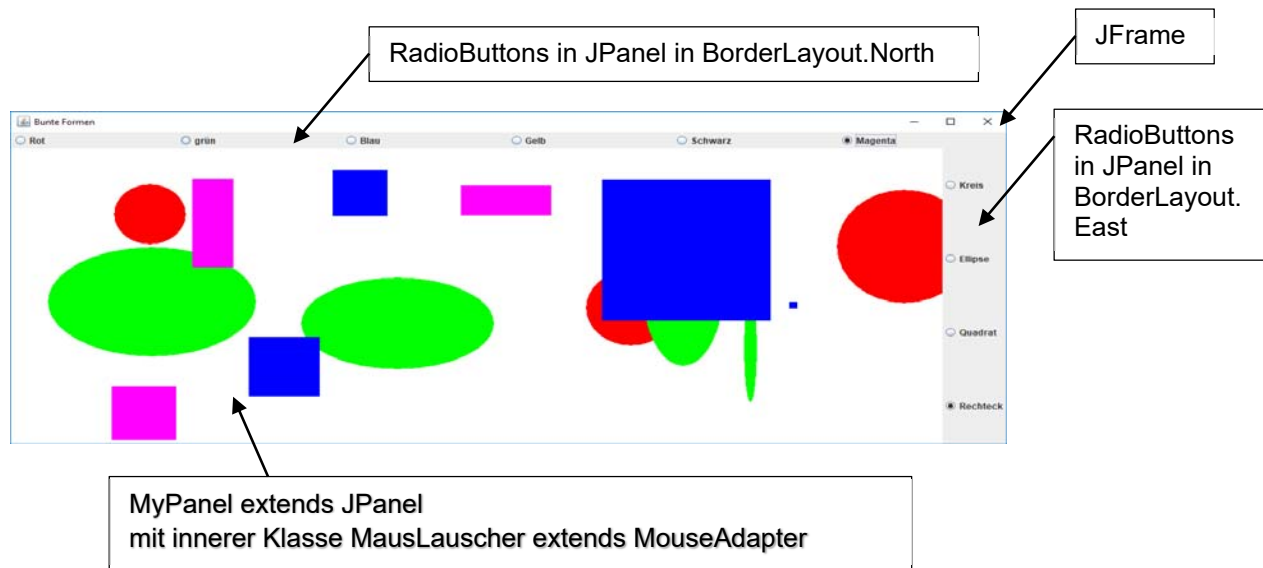


## Einfache Grafik Anwendungen mit Java Graphics Bibliothek

*Ziel: Erstellung einer einfachen Grafikanwendung: Kennenlernen elementarer Methoden der Graphics Klasse..*

1.1. Die zu erstellende Anwendung hat folgende GUI:



- Erstellen Sie eine GUI Anwendung mit dem Window Builder
- Der ContentPane enthält ein BorderLayout
- Im BorderLayout.North befindet sich ein JPanel für die RadioButtons „Rot“, „Grün“, „Blau“, „Gelb“, „Schwarz“ und „Magenta“. Die RadioButtons müssen zu einer ButtonGroup zusammengefasst werden. Dazu markiert man im Design-Fenster alle RadioButton und wählt im Kontextmenü (rechte Maustaste) `setButtonGroup` → `NewStandard`
- Im BorderLayout.East befindet sich ein JPanel für die RadioButtons „Kreis“, „Ellipse“, „Quadrat“ und „Rechteck“, die zusammen auch eine ButtonGroup bilden.
- Im BorderLayout.Center befindet sich die Malfläche: ein Objekt der Klasse `MyPanel`, die von `JPanel` abgeleitet wird. Diese besitzt zwei Attribute `x_pos` und `y_pos` zum Speichern der Koordinaten des letzten Mausklicks.
- Die Klasse `MyPanel` besitzt außerdem eine innere Klasse für den „MausLauscher“. Die `MausLauscher` Klasse implementiert die Methode

```
public void mousePressed(MouseEvent e)
{
    x_pos = e.getX(); // Koordinaten des Mausklicks speichern
    y_pos = e.getY();
    Graphics g = getGraphics(); // Grafikkontext ermitteln
    paint(g); // überschriebene paint Methode aufrufen
}
```
- In der `paint` Methode wird dann abhängig von den ausgewählten RadioButtons eine Form an den Koordinaten `x_pos`, `y_pos` gezeichnet. Die Größe der jeweiligen Form wird durch einen Zufallswert für die Höhe und Breite bestimmt.

```
* @author kds[]
public class Paint_WB
{
    private JFrame frame;
    private JRadioButton rbtnRot;
    private JRadioButton rbtnGruen;
    private JRadioButton rbtnBlau;
    private JRadioButton rbtnGelb;
    private JRadioButton rbtnSchwarz;
    private JRadioButton rbtnMagenta;
    private JRadioButton rbtnKreis;
    private JRadioButton rbtnEllipse;
    private JRadioButton rbtnQuadrat;
    private JRadioButton rbtnRechteck;

    * Launch the application.[]
    public static void main(String[] args)[]

    * Create the application.[]
    public Paint_WB()[]

    * Initialize the contents of the frame.[]
    private void initialize()
    {
        frame = new JFrame();
        frame.setBounds(200, 200, 1250, 500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(new BorderLayout(0, 0));

        JPanel pnlFarbauswahl = new JPanel();
        frame.getContentPane().add(pnlFarbauswahl, BorderLayout.NORTH);
        pnlFarbauswahl.setLayout(new GridLayout(0, 6, 0, 0));

        rbtnRot = new JRadioButton("Rot");
        pnlFarbauswahl.add(rbtnRot);

        // weitere RadioButtons ...

        ButtonGroup farbauswahl = new ButtonGroup();
        farbauswahl.add(rbtnRot);
        farbauswahl.add(rbtnGruen);
        farbauswahl.add(rbtnBlau);
        farbauswahl.add(rbtnGelb);
        farbauswahl.add(rbtnSchwarz);
        farbauswahl.add(rbtnMagenta);

        JPanel pnlFormauswahl = new JPanel();
        frame.getContentPane().add(pnlFormauswahl, BorderLayout.EAST);
        pnlFormauswahl.setLayout(new GridLayout(0, 1, 0, 0));

        rbtnKreis = new JRadioButton("Kreis");
        pnlFormauswahl.add(rbtnKreis);

        rbtnEllipse = new JRadioButton("Ellipse");
        pnlFormauswahl.add(rbtnEllipse);

        rbtnQuadrat = new JRadioButton("Quadrat");
        pnlFormauswahl.add(rbtnQuadrat);

        rbtnRechteck = new JRadioButton("Rechteck");
        pnlFormauswahl.add(rbtnRechteck);

        ButtonGroup formauswahl = new ButtonGroup();
        formauswahl.add(rbtnKreis);
        formauswahl.add(rbtnEllipse);
        formauswahl.add(rbtnQuadrat);
        formauswahl.add(rbtnRechteck);

        // Die Malfläche ist ein Objekt von MyPanel
        frame.getContentPane().add(new MyPanel(), BorderLayout.CENTER);
    }
}
```

```
public MyPanel()
{
    setBackground(Color.WHITE);
    addMouseListener(new Mein_MausAdapter());
}
// innere eigene Lauscherklasse für Mausereignisse
class Mein_MausAdapter extends MouseAdapter
{
    public void mousePressed(MouseEvent e)
    {
        x_pos = e.getX();
        y_pos = e.getY();
        Graphics g = getGraphics();
        //zeichnen(); // Alternative in Kommentar!
        paint(g);
    }
}
//public void zeichnen()
public void paint (Graphics g)
{
    //Graphics g = getGraphics();
    Color farbe = null;
    // w (=Breite) und h (=Höhe) der Form werden mit einer
    // Zufallsfunktion ermittelt
    int w = (int) (Math.random() * 200);
    int h = (int) (Math.random() * 200);
    // welche Farbe wurde ausgewählt?
    if (rbtnRot.isSelected()) // (***)
        farbe = Color.red;
    else if (rbtnGruen.isSelected()) farbe = Color.green;
    else if (rbtnBlau.isSelected()) farbe = Color.blue;
    else if (rbtnGelb.isSelected()) farbe = Color.yellow;
    else if (rbtnSchwarz.isSelected()) farbe = Color.black;
    else if (rbtnMagenta.isSelected()) farbe = Color.magenta;
    g.setColor(farbe);

    // welche Form wurde ausgewählt?
    if (rbtnKreis.isSelected()) // (***)
        g.fillOval(x_pos, y_pos, w, w);
    else if (rbtnEllipse.isSelected()) g.fillOval(x_pos, y_pos, w, h);
    else if (rbtnQuadrat.isSelected()) g.fillRect(x_pos, y_pos, w, w);
    else if (rbtnRechteck.isSelected()) g.fillRect(x_pos, y_pos, w, h);
}
}
```