

Freie Positionierung von GUI-Komponenten?

→ Für Größe und Position ist der
LayoutManager verantwortlich!

→ freie Positionierung also nur ohne (oder
eigenen) LayoutManager!

```
JPanel jp = new JPanel(null);
```

```
JFrame jf = new ...
```

```
jf.setLayout(null);
```

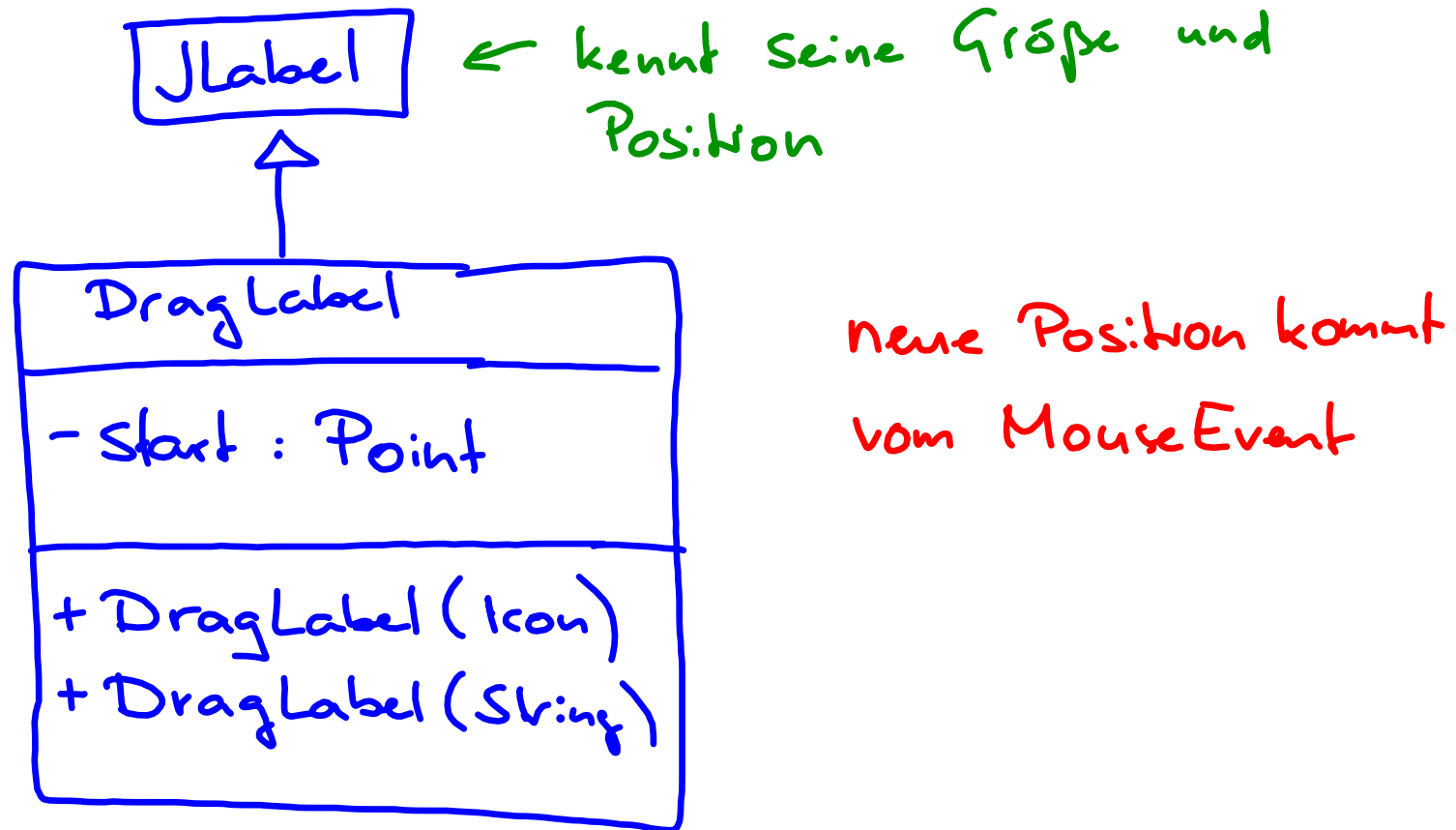
↑
„NullLayout“

Aufgabe:

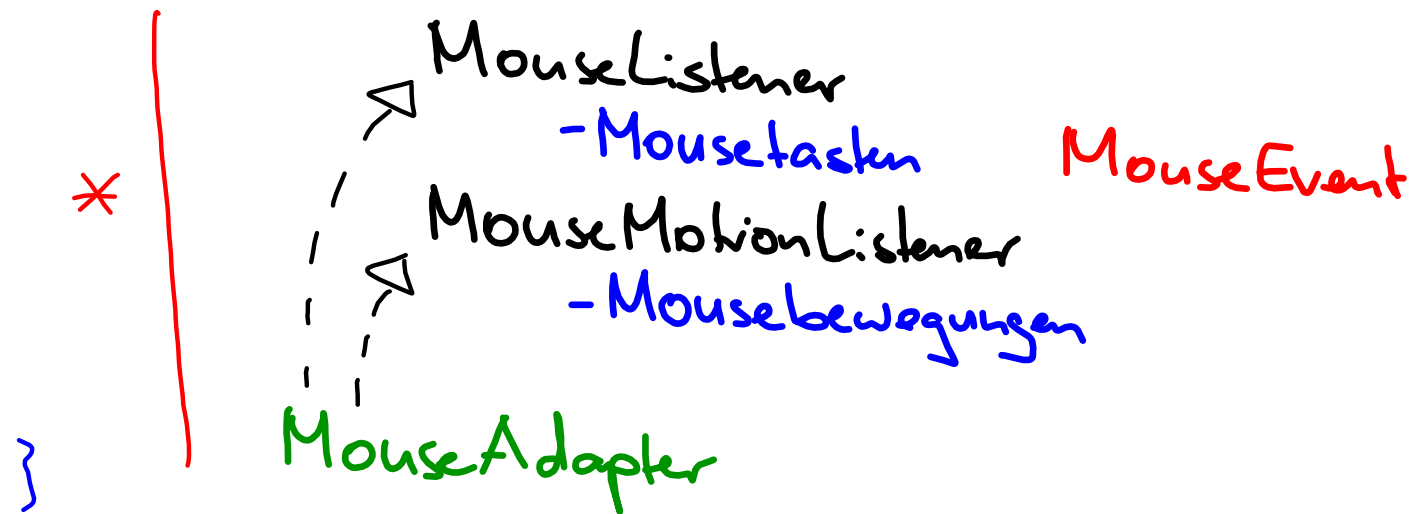
Schreiben Sie ein Programm, das in einem Fenster ein JLabel frei positioniert darstellt.

Hinweis: Wenn Sie das „NullLayout“ verwenden, sind Sie selbst für Größe und Position aller anzuzugrunden GUI-Komponenten im Kontainer verantwortlich.

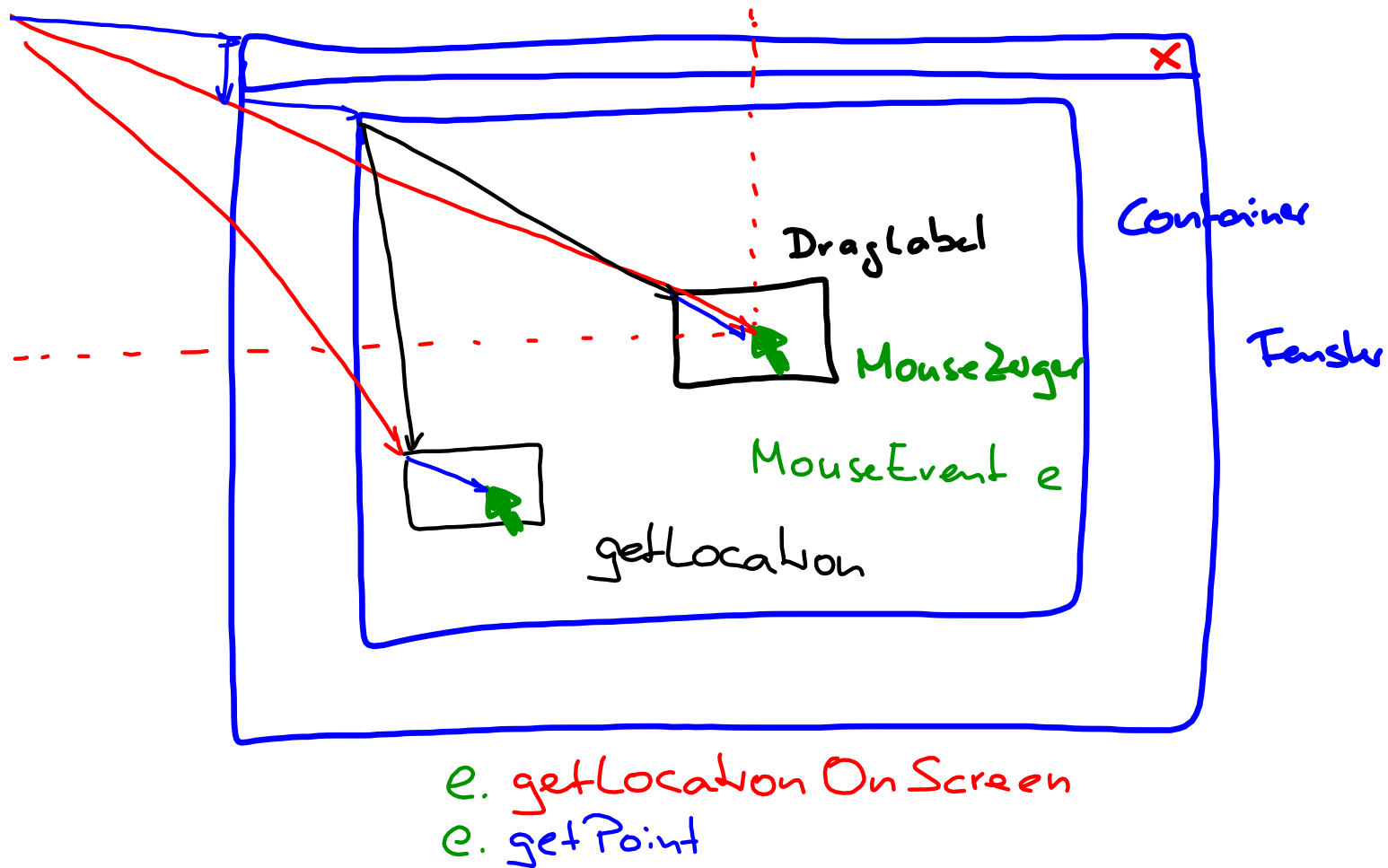
```
pSvm (...) {  
    JFrame jf = new JFrame("NullLayout Test");  
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    jf.setLayout(null); DragLabel  
    JLabel minion = new JLabel(new ImageIcon("minion.png"));  
    minion.setSize(minion.getPreferredSize());  
    minion.setLocation(50, 70);  
    jf.add(minion);  
  
    jf.setSize(500, 500);  
    jf.setVisible(true);  
}
```



```
public class DragLabel extends JLabel {  
    private Point start;  
  
    public DragLabel (Icon icon) {  
        super(icon);  
    }  
}
```

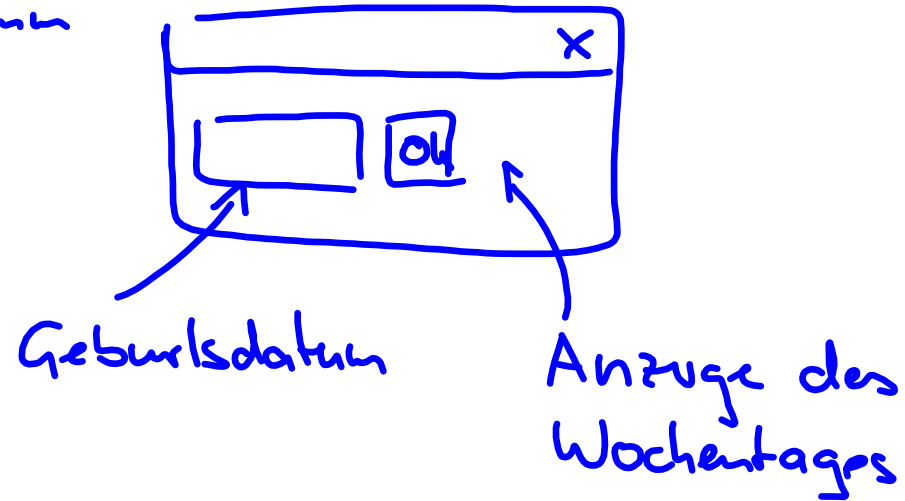


```
* MouseAdapter ma = new MouseAdapter() {  
    public void mousePressed(MouseEvent e) {  
        start = new Point(e.getLocationOnScreen().x -  
            getLocation().x, e.getLocationOnScreen().y -  
            getLocation().y);  
    }  
    public void mouseDragged(MouseEvent e) {  
        setLocation(e.getLocationOnScreen().x - start.x,  
            e.getLocationOnScreen().y - start.y);  
    }  
};  
addMouseListener(ma);  
addMouseMotionListener(ma);
```



Vertiefungsaufgabe:

Programm



Calendar

Schreiben Sie ein Programm, das in einer GUI das Geburtsdatum erfragt und nach dem Klick auf ein JButton in einem JLabel den Wochentag (Mo.-So.) anzeigt.