

1. Arbeiten mit der Maus

Mausereignisse werden durch Benutzeraktionen ausgelöst, die sich auf die Computermaus beziehen. Leider werden in den unterschiedlichen Java-Versionen unterschiedliche Modelle zur Behandlung von Mausereignissen benutzt. U.a. führt das zu der unschönen Situation, dass es verschiedene Methoden zur Behandlung desselben Ereignisses gibt, die – wenn man versehentlich eine „falsche“ Bezeichnung wählt - anstandslos übersetzt werden, ohne zum gewünschten Ergebnis zu führen. (Bsp: es gibt die Methoden *mouseDrag()* und *mouseDragged()*) Wir werden hier zuerst das veraltete erste Modell benutzen, weil es so schön einfach ist. (Auch wenn die Java-Autoren das „missbilligen“: die Methoden sind *deprecated*.)

Ereignisse werden von Java in einem Ereignis der Klasse *Event* gekapselt, das genauere Informationen über die Art des ausgelösten Ereignissen enthält. Wir müssen es in der Parameterliste der benutzten Methoden angeben, werden es aber (noch nicht) benutzen. Zusätzlich enthalten die veralteten Maus-Methoden zwei ganze Zahlen *x* und *y*, die den Ort des Mausereignisses auf dem Bildschirm angeben. Diese werden wir auswerten.

Bei der Maus kann man die Maustaste drücken und wieder loslassen.

```
public boolean mouseDown(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}

public boolean mouseUp(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}
```

Weiterhin kann die Maus mit oder ohne gedrückter Maustaste bewegt werden:

```
public boolean mouseDrag(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}

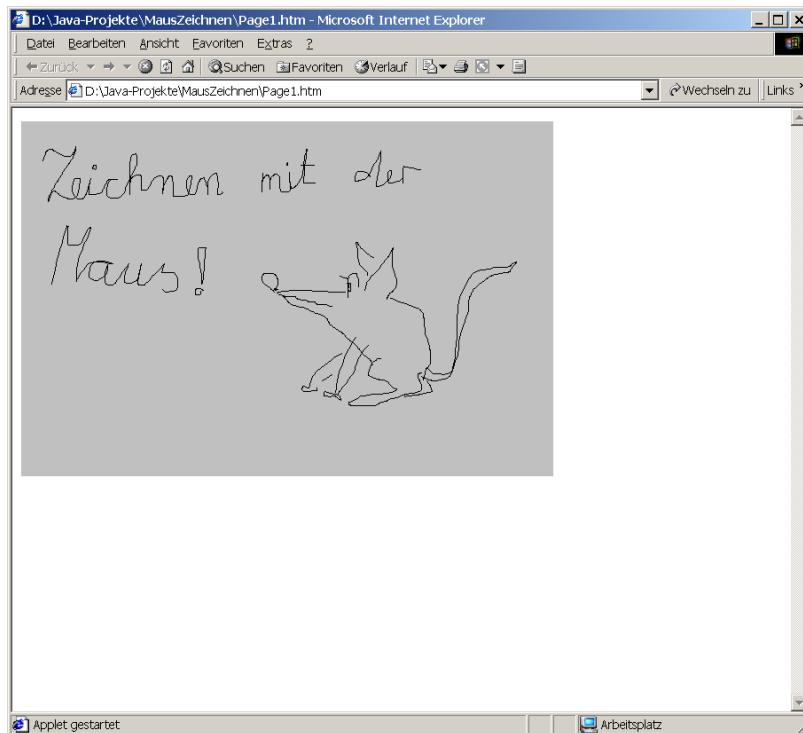
public boolean mouseMove(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}
```

Und zuletzt kann die Maus einen Bereich oder ein Bildelement „betreten“ (darüber bewegt werden) oder es „verlassen“.

```
public boolean mouseEnter(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}

public boolean mouseExit(Event e, int x, int y)
{
    // hier stehen geeignete Anweisungen
    return true;
}
```

Wir wollen die Maus zum Zeichnen benutzen. Dazu merken wir uns jeweils die Koordinaten des Punktes, an dem die Maustaste gedrückt wurde und benutzen diese als ersten Anfangspunkt einer Linie, die gezeichnet wird, wenn die Maus bei gedrückter Maustaste bewegt wird. Nachdem eine Linie gezeichnet wurde, merken wir uns deren Endpunkt als neuen Anfangspunkt.



```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
```

```
public class Applet1 extends Applet
{
    int xanf,yanf;

    public boolean mouseDown(Event e, int x, int y)
    {
        xanf=x; yanf=y;
        return true;
    }

    public boolean mouseDrag(Event e, int x, int y)
    {
        Graphics g=getGraphics();
        g.drawLine(xanf,yanf,x,y);
        xanf=x; yanf=y;
        return true;
    }

    public void update(Graphics g)
    {
        paint(g);
    }
}
```

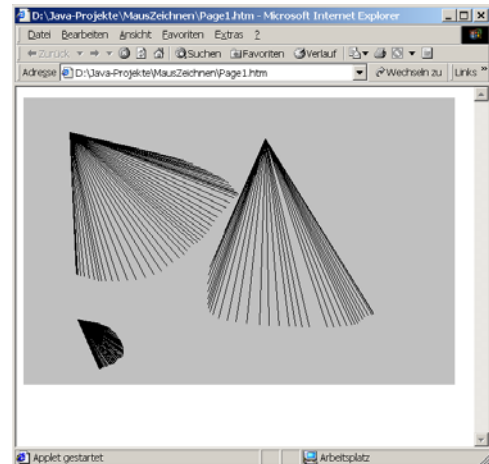
Koordinaten des jeweils ersten Punktes

Beim jeweils ersten Tastendruck Koordinaten merken

Bei Mausbewegungen Linie zeichnen und neuen „Anfangspunkt“ merken

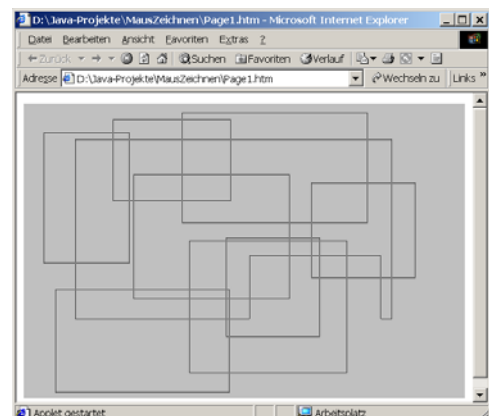
2. Aufgaben

1. a: Verändern Sie die Anfangspunktkoordinaten beim Zeichnen nicht. Erzeugen Sie dann so etwas wie die abgebildeten „Fächer“.
 - b: Lassen Sie die Linienfarben beim Zeichnen zufällig verändern.
 - c: Lassen Sie die Linienfarben zufällig, aber „schön“ verändern, so dass sich „geschmackvolle“ Fächer ergeben.
2. Sparen Sie eine „Leiste“ am oberen Bildrand vom Zeichnen aus. Färben Sie die Leiste ein.



3. a: Führen Sie farbige Flächen („Buttons“) auf der oberen Leiste ein, die aus Mausklicks reagieren, z.B. ihre Farbe ändern oder sichtbar „gedrückt“ werden.
- b: Beschriften Sie die Buttons. Verändern Sie bei Mausklicks auf die Buttons die Zeichenfarbe.

4. a: Lassen Sie mit Hilfe der Maus Rechtecke am Bildschirm zwischen Anfangspunkt und aktuellem Mauspunkt zeichnen.
- b: Zeichnen Sie die Rechtecke im invertierenden Modus. Löschen Sie die gerade gezeichneten Rechtecke nicht zu schnell, sonst sehen Sie gar nichts am Bildschirm, weil die Grafikkarte nicht schnell genug ist!
- c: Füllen Sie die so gezeichneten Rechtecke farblich aus, nachdem die Maustaste losgelassen wurde.
- d: Lassen Sie die Zeichenfarbe mit Hilfe der Buttons wählen.



- e: Schreiben Sie jetzt das Programm „Mondrian“ so um, dass „mondrianische“ Bilder mit der Maus gezeichnet werden können.
5. Benutzen Sie Buttons, um unterschiedliche Formen am Bildschirm erzeugen zu können. Man kann z.B. Kreise „aufziehen“, Polygone zeichnen, Dreiecke „setzen“, ...
 6. Überlegen Sie sich die auftretenden Probleme, wenn man gezeichnete Rechtecke wieder löschen möchte. Ähnliche Aufgaben erwarten Sie, wenn Sie die Rechtecke nachträglich

