Обработка действий мыши

Событие MouseEvent возникает в компоненте по любой из семи причин:

- нажатие кнопки мыши — идентификатор MOUSE_PRESSED;
- отпускание кнопки мыши — идентификатор MOUSE_RELEASED;
- щелчок кнопкой мыши — идентификатор MOUSE_CLICKED (нажатие и отпускание не
- различаются);
- перемещение мыши — идентификатор MOUSE_MOVED;
- перемещение мыши с нажатой кнопкой — идентификатор MOUSE_DRAGGED;
- появление курсора мыши в компоненте — идентификатор MOUSE_ENTERED;
- выход курсора мыши из компонента — идентификатор MOUSE_EXITED.

Для их обработки есть семь методов в двух интерфейсах:

```
public interface MouseListener extends EventListener{
public void mouseClicked(MouseEvent e);
public void mousePressed(MouseEvent e);
public void mouseReleased(MouseEvent e);
public void mouseEntered(MouseEvent e);
public void mouseExited(MouseEvent e);
}
public interface MouseMotionListener extends EventListener{
public void mouseDragged(MouseEvent e);
public void mouseMoved(MouseEvent e);
}
```

Эти методы могут получить от параметра e координаты курсора мыши в системе координатомпонента методами e.getX(), e.getY() или одним методом e.getPoint(), возвращающим экземпляр класса Point. Координаты курсора мыши относительно всего экрана,определяемые объектом класса GraphicsConfiguration, можно получить методами e.getXOnScreen(), e.getYOnScreen() илие.getLocationOnScreen().
Двойной щелчок кнопкой мыши можно отследить методом e.getClickCount(), возвращающим количество щелчков. При перемещении мыши возвращается 0.

Узнать, какая кнопка была нажата, можно с помощью метода e.getModifiers() класса InputEvent сравнением со следующими статическими константами класса InputEvent:

- BUTTON1_MASK — нажата первая кнопка, обычно левая;
- BUTTON2_MASK — нажата вторая кнопка, обычно средняя, или одновременно нажаты
- обе кнопки на двухкнопочной мыши;
- BUTTON3_MASK — нажата третья кнопка, обычно правая.

Пример простейшего варианта "рисовалки" — класс Scribble. При нажатии первой кнопки мыши методоммousePressed() запоминаются координаты курсора мыши. При протаскивании мыши вычерчиваются отрезки прямых между текущим и предыдущим положением курсора мыши методом mouseDragged().

При создании класса-слушателя Scribble и реализации интерфейсов MouseListener и MouseMotionListener пришлось реализовать все их семь методов, хотя мы отслеживали только нажатие кнопки и перемещение мыши, и нам нужны были лишь методы mousePressed() и mouseDragged(). Для остальных методов мы задали пустые реализации. Чтобы облегчить задачу реализации интерфейсов, имеющих более одного метода, созданы классы-адаптеры.

Классы-адаптеры представляют собой пустую реализацию интерфейсов-слушателей, имеющих более одного метода. Их имена составляются из имени события и слова "Adapter".
Например, для действий с мышью есть два класса-адаптера. Выглядятониоченьпросто:

```
public abstract class MouseAdapter implements MouseListener{
public void mouseClicked(MouseEvent e){}
public void mousePressed(MouseEvent e){}
public void mouseReleased(MouseEvent e){}
public void mouseEntered(MouseEvent e){}
public void mouseExited(MouseEvent e){}
public void mouseMoved(MouseEvent e){}
public void mouseDragged(MouseEvent e){}
public void mouseWheelMoved(MouseEvent e){}
}
public abstract class MouseMotionAdapter implements MouseMotionListener{
public void mouseDragged(MouseEvent e){}
public void mouseMoved(MouseEvent e){}
}
```

Вместо того чтобы реализовать интерфейс, можно расширять эти классы.

Классов-адаптеров всего семь. Кроме уже упомянутых трех классов, это классы ComponentAdapter, ContainerAdapter, FocusAdapterиKeyAdapter.