



Рис. 12.4. Простейший текстовый редактор с классами-адаптерами

Конструктор класса **PaintEditor** использует методы **addMouseListener(new PaintMouseListener(this))** и **addMouseMotionListener(new PaintMouseMotionAdapter(this))** для регистрации событий мыши. При создании объекта класса **PaintEditor** эти методы сообщают объекту, что он заинтересован в обработке определенных событий. Однако вместо того, чтобы известить его об этом прямо, конструктор организует посылку ему предупреждений через объекты классов **PaintMouseListener** и **PaintMouseMotionAdapter**. Абстрактный класс **MouseListener** используется для обработки событий, связанных с мышью при создании блока прослушивания, и содержит следующие переопределяемые методы: **mousePressed(MouseEvent e)**, **mouseReleased(MouseEvent e)**. Абстрактный класс **MouseMotionAdapter** используется для обработки событий, связанных с движениями мыши при создании блока прослушивания, и содержит следующие переопределяемые методы: **mouseDragged(MouseEvent e)**, **mouseMoved(MouseEvent e)**.

Класс **PaintEditor** также обрабатывает событие класса **WindowEvent**. Когда объект генерирует событие **WindowEvent**, объект **PaintEditor** анализирует, является ли оно событием **WindowClosing**. Если это не так, объект **PaintEditor** игнорирует его. Если получено ожидаемое событие, в программе запускается процесс завершения ее работы.

Абстрактный класс **WindowAdapter** используется для приема и обработки событий окна при создании объекта прослушивания. Класс содержит методы: **windowActivated(WindowEvent e)**, вызываемый при активизации окна; **windowClosing(WindowEvent e)**, вызываемый при закрытии окна, и др.

Задания к главе 12

1. Создать фрейм с областью для рисования “пером”. Создать меню для выбора цвета и толщины линии.
2. Создать апплет с областью для рисования “пером”. Создать меню для выбора цвета и толщины линии.
3. Создать фрейм с областью для рисования. Добавить кнопки для выбора цвета (каждому цвету соответствует своя кнопка), кнопку для очистки окна. Рисование на панели со скроллингом.
4. Создать апплет с областью для рисования. Добавить кнопки для выбора цвета (каждому цвету соответствует своя кнопка), кнопку для очистки окна. Рисование на панели со скроллингом.

5. Создать простой текстовый редактор, содержащий меню и использующий классы диалогов для открытия и сохранения файлов.
6. Создать апплет, содержащий (**JLabel**) с текстом “Простой апплет”, кнопку и текстовое поле (**TextField**), в которое при каждом нажатии на кнопку выводится по одной строке из текстового файла.
7. Изменить апплет для предыдущей задачи таким образом, чтобы он мог работать и как апплет, и как приложение.
8. Составить программу для управления скоростью движения точки по апплету. Одна кнопка увеличивает скорость, другая – уменьшает. Каждый щелчок изменяет скорость на определенную величину.
9. Изобразить в окне гармонические колебания точки вдоль некоторого горизонтального отрезка. Если длина отрезка равна q , то расстояние от точки до левого конца в момент времени t можно считать равным $q(1 + \cos(wt))/2$, где w – некоторая константа. Предусмотреть поля для ввода указанных величин и кнопку для остановки и пуска процесса.
10. Для предыдущей задачи предусмотреть возможность управления частотой колебаний с помощью двух кнопок. С помощью других двух кнопок (можно клавиш) управлять амплитудой, т.е. величиной q .
11. Построить в апплете ломаную линию по заданным вершинам. Координаты вершин вводятся через текстовое поле и фиксируются нажатием кнопки.
12. Нарисовать в апплете окружность с координатами центра и радиусом, вводимыми через текстовые поля.
13. Создать апплет со строкой, которая движется горизонтально, отражаясь от границ апплета и меняя при этом свой цвет на цвет, выбранный из выпадающего списка.
14. Создать апплет со строкой, движущейся по диагонали. При достижении границ апплета все символы строки случайным образом меняют регистр. При этом шрифт меняется на шрифт, выбранный из списка.

Тестовые задания к главе 12

Вопрос 12.1.

Выбрать необходимое условие принадлежности класса к апплетам.

- 1) класс – наследник класса **Applet** при отсутствии метода **main()** ;
- 2) класс – наследник класса **Applet** или его подкласса;
- 3) класс – наследник класса **Applet** с переопределенным методом **paint()** ;
- 4) класс – наследник класса **Applet** с переопределенным методом **init()** ;
- 5) класс – наследник класса **Applet**, и все его методы объявлены со спецификатором доступа **public**.

Вопрос 12.2.

Дан код:

```
import java.awt.*;
public class Quest2 extends Frame{
```