Sistemas Multimedia

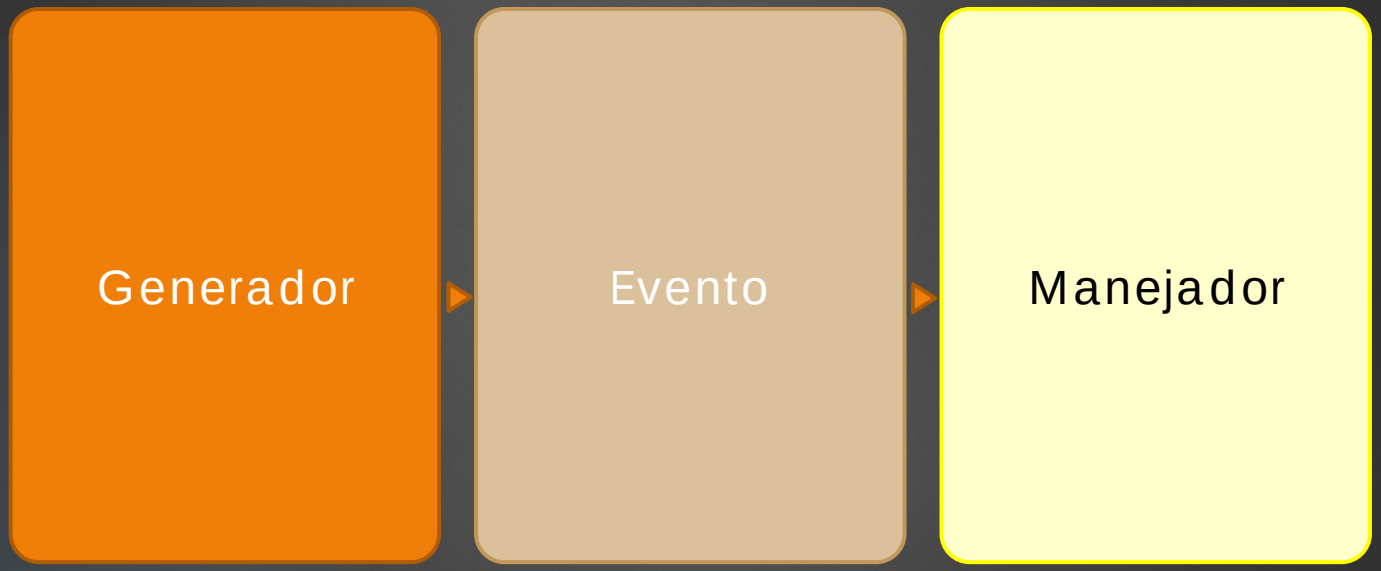
Eventos

Índice

- Roles
- Eventos
- Generadores de eventos
- Manejadores de eventos

Roles

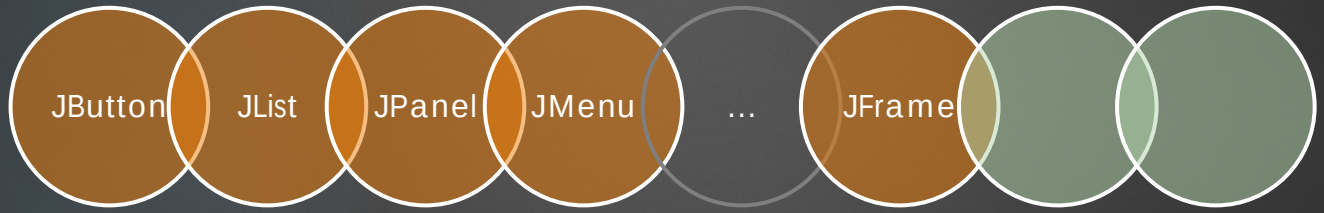
- En el proceso de gestión de eventos intervienen tres objetos: generador, evento y manejador



¿Existen clases predeterminadas? ¿Son suficientes o tenemos que crear clases propias?

Generadores de eventos

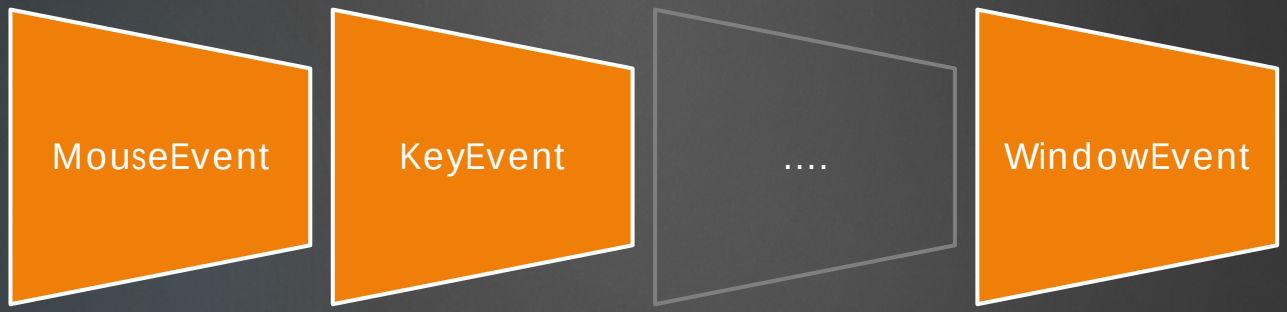
- Objetos que “generan” eventos “por el motivo que sea” (p.e., un botón al pulsarlo). El “motivo” determinará la clase del evento
- Muchas clases de Java son generadoras. En particular, todos los elementos IU de la Swing (componentes, contenedores, etc.) generan eventos



- Se pueden crear clases propias que sean generadoras (si bien en nuestras prácticas serán suficientes las clases ya existentes)

Eventos

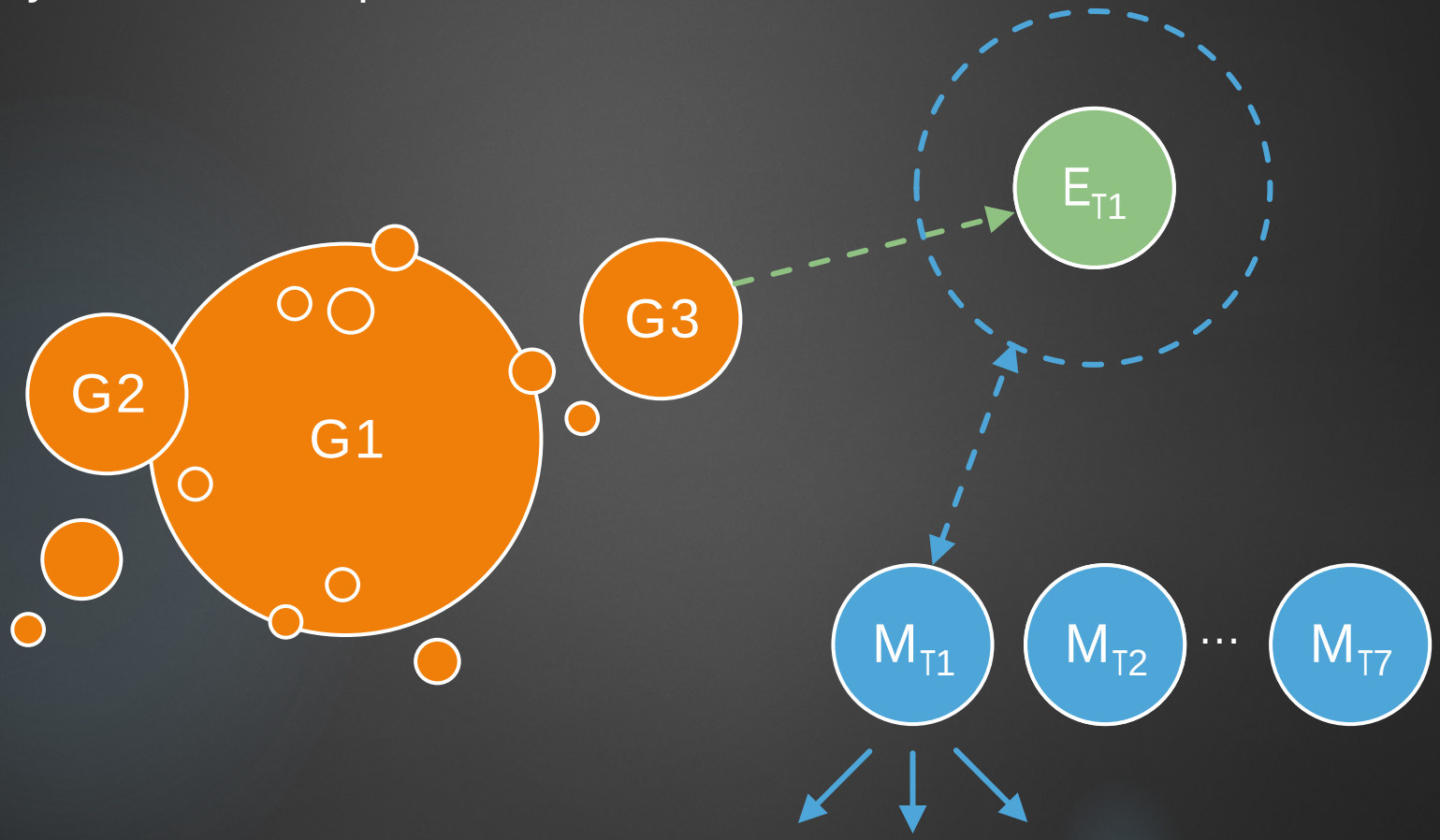
- Objeto que representa “que algo ha pasado” y contiene información sobre “lo que ha pasado”
- Existen clases predeterminadas para muchos tipos de eventos (todas heredando de *Event*)



- Se pueden crear clases propias de eventos (si bien en nuestras prácticas serán suficientes las clases ya existentes)

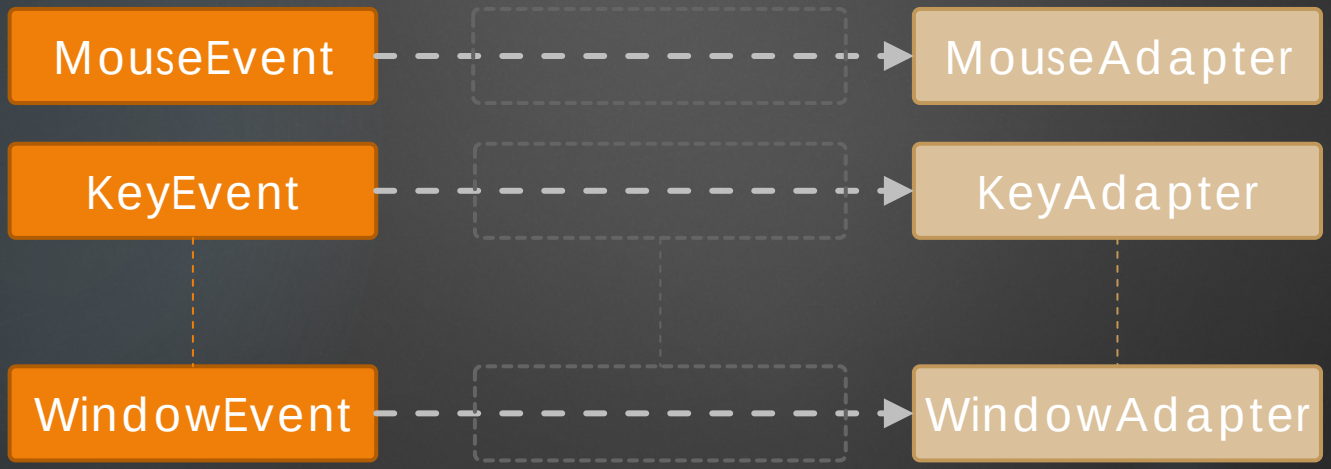
Manejadores eventos

- Objetos que “capturan” un evento “lo procesan” y dan una respuesta/acción



Manejadores eventos

- Objetos que “capturan” un evento “lo procesan” y dan una respuesta/acción
- Existen clases predeterminadas (una por cada tipo de evento), pero es necesario crear clases manejadoras propias (que hereden de las clases predeterminadas)



Manejadores eventos

- En la clase manejadora habrá métodos asociados a los diferentes “motivos” que han podido generar el evento. Cuando el manejador “capture” el evento, ejecutará el método de su clase asociado al motivo que provocó el evento

```
public abstract class MouseAdapter ... {
  public void mouseClicked(MouseEvent e) {}
  public void mousePressed (MouseEvent e) {}
  public void mouseReleased (MouseEvent e) {}
  public void mouseEntered (MouseEvent e) {}
  public void mouseExit(MouseEvent e) {}
  ...
}
```

```
public class MiManejadorRatón extends MouseAdapter {
  public void mouseClicked(MouseEvent e) {
     //Código a ejecutar cuando se capture un clic
  }
}
```

Manejadores eventos

1. Definir la clase manejadora
   - Heredando de la clase asociada al tipo de evento
   - Sobrecargar los métodos que interesen
2. Crear objeto manejador de eventos

```
MiManejadorRatón manejador = new MiManejadorRatón();
```

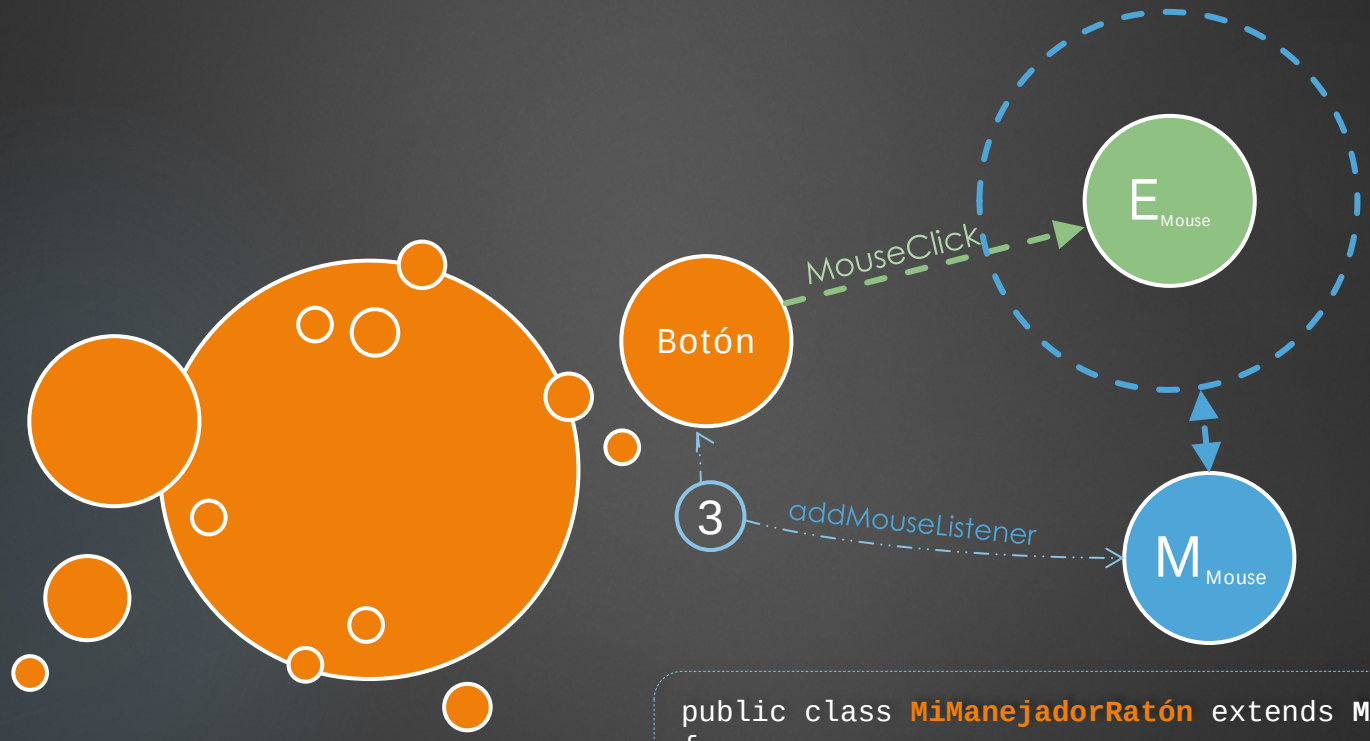
3. Enlazar generador y manejador

```
generador.addMouseListener(manejador);
```

```
public class MiManejadorRatón extends MouseAdapter {
  public void mouseClicked(MouseEvent e) {
     //Código a ejecutar cuando se capture un clic
  }
}
```

Manejadores eventos

- Ejemplo “mouseClick”



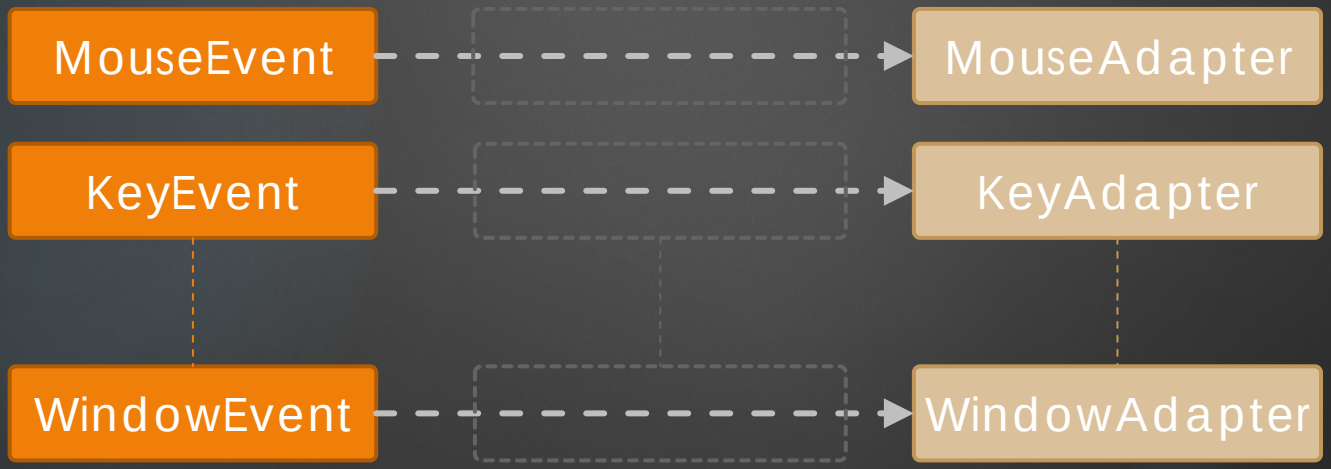
1

```
public class MiManejadorRatón extends MouseAdapter
{
  public void mouseClicked(MouseEvent e) {
     //Código a ejecutar
  }
}
```

2 new ...

Manejadores eventos

- Objetos que “capturan” un evento “lo procesan” y dan una respuesta/acción
- Existen clases predeterminadas (una por cada tipo de evento), pero es necesario crear clases manejadoras propias (que hereden de las clases predeterminadas)



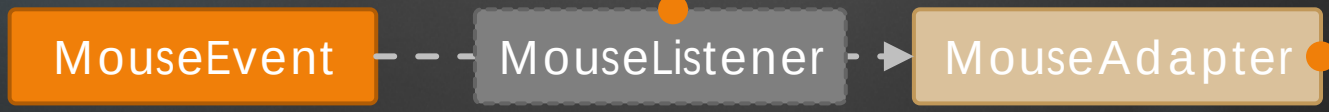
Manejadores eventos

- Objetos que “capturan” un evento “lo procesan” y dan una respuesta/acción
- Existen clases predeterminadas (una por cada tipo de evento), pero es necesario crear clases manejadoras propias (que hereden de las clases predeterminadas)

Manejadores eventos

```
public class MiManejadorRatón extends MouseAdapter {
  public void mouseClicked(MouseEvent e) {
     //Código a ejecutar cuando se capture un clic
  }
}
```

```
public class MiManejadorRatón implements MouseListener {
  public void mouseClicked(MouseEvent e) {
     //Código a ejecutar cuando se capture un clic
  }
  public void mouseClicked(MouseEvent e) {
  }
  public void mousePressed (MouseEvent e) {
  }
  public void mouseReleased (MouseEvent e) {
  }
  public void mouseEntered (MouseEvent e) {
  }
  public void mouseExit(MouseEvent e) {
  }
}
```



Manejadores eventos

```
public abstract class MouseAdapter implements MouseListener,
                                               MouseWheelListener,
                                               MouseMotionListener {

  public void mouseClicked(MouseEvent e) {}
  public void mousePressed (MouseEvent e) {}
  public void mouseReleased (MouseEvent e) {}
  public void mouseEntered (MouseEvent e) {}
  public void mouseExit(MouseEvent e) {}
  void mouseWheelMoved(MouseWheelEvent e) {}
  void mouseDragged(MouseEvent e) {}
  void mouseMoved(MouseEvent e){}

}
```