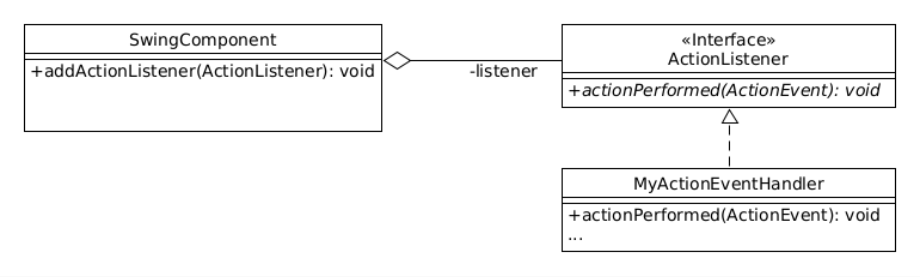
Swing Events

Carsten Gips (HSBI)

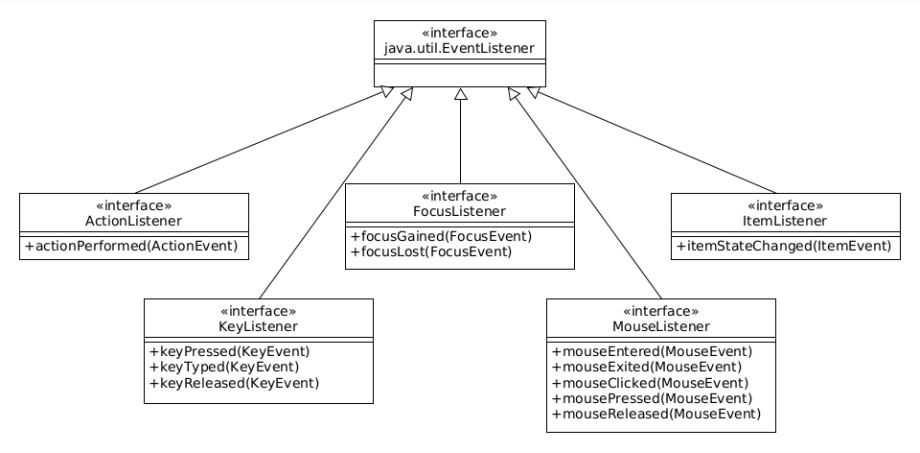
Reaktion auf Events: Anwendung Observer-Pattern



=> Observer aus dem Observer-Pattern!

```
component.addActionListener(ActionListener);
component.addMouseListener(MouseListener);
```

Arten von Events



Details zu Listenern

- Ein Listener kann bei mehreren Observables registriert sein:

```
Handler single = new Handler();
singleButton.addActionListener(single);
multiButton.addActionListener(single);
```

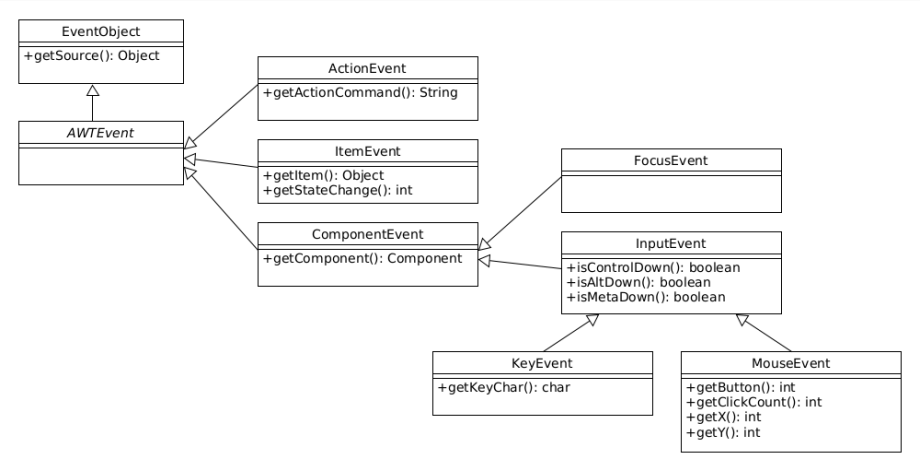
- Ein Observable kann mehrere Listener bedienen:

```
multiButton.addActionListener(new Handler());
multiButton.addActionListener(new Handler());
```

- Sequentielles Abarbeiten der Events bzw. Benachrichtigung der Observer

Demo: events.ListenerDemo

Wie komme ich an die Daten eines Events?



Event-Objekte: Quelle des Events plus aufgetretene Daten

Demo: events.MouseListenerDemo

Listener vs. Adapter

=> Bei Nutzung eines Event-Listeners müssen immer **alle** Methoden implementiert werden

Abhilfe: **Adapter**-Klassen:

- Für viele Event-Listener-Interfaces existieren Adapter-Klassen
- Implementieren jeweils ein Interface
- Alle Methoden mit **leerem** Body vorhanden

=> Nur benötigte Listener-Methoden überschreiben.

Demo: events.MouseAdapterDemo

Wrap-Up

Observer-Pattern in Swing-Komponenten:

- Events: Enthalten Source-Objekt und Informationen
- Event-Listener: Interfaces mit Methoden zur Reaktion
- Adapter: Listener mit leeren Methodenrümpfen

LICENSE