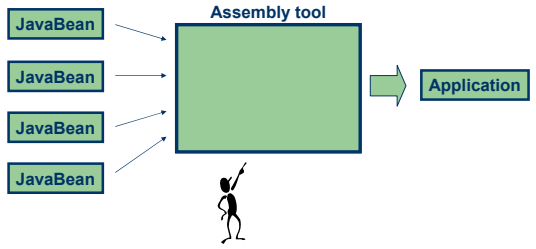
JavaBeans

Jens Gustavsson

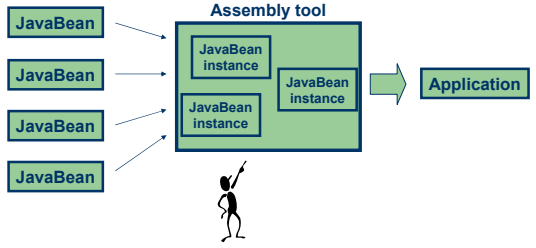
Overview



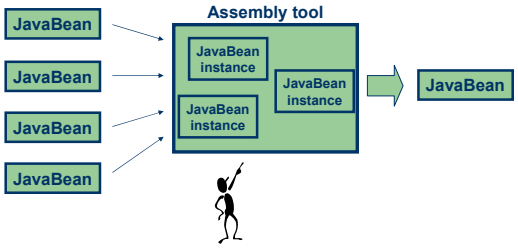
Introduction

- Component model for Java
- A JavaBean is a component
- Visual composition
- Several tool and component developers
- JavaBeans != Enterprise JavaBeans

Overview



Overview



Example: Alarm Clock

- Properties
  - Current time
  - Alarm time
  - Alarm status (set/not set)
- Events
  - Alarm (source)

Customization and Connections

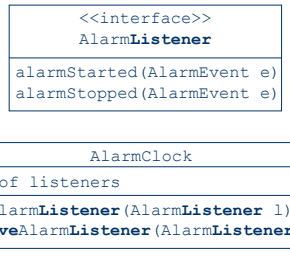
What does a bean look like?

- Public constructor without parameters
- Implement interface Serializable
- Design patterns defines properties and events

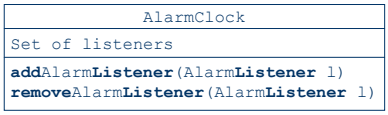
Properties

```
class AlarmClock implements Serializable {
    public AlarmClock() {…}

    public boolean getAlarmStatus() {…}

    public void setAlarmStatus(boolean value) {…}

    …
}
```

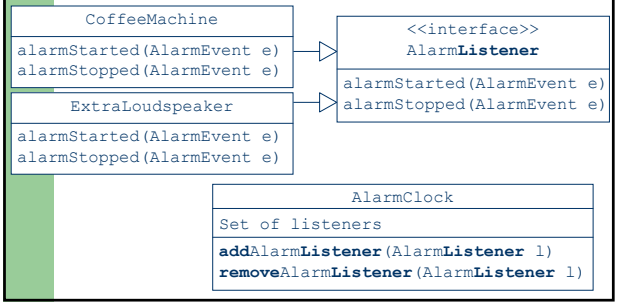
Events



Events



Events



Advanced features

- Optional class: `AlarmClock`**`Info`**
- Optional class: `AlarmClock`**`Customizer`**
- Indexed properties
- Bound properties
- Constrained properties

Miscellaneous

- Swing components are JavaBeans
- Bridge JavaBean - ActiveX

How does it work?

- Java techniques:
  - Serialization makes persistence possible
  - Dynamic class loading
  - Reflection makes naming convention based scheme possible

Evaluation

- Strengths
  - Simple - easy to use
  - Standard - mix vendors
  - Applicable for GUI development
- Weaknesses
  - Only suitable for GUI development
  - Not usable for non-programmers
  - Weak component market