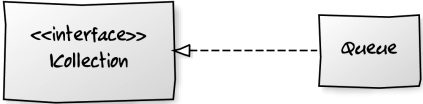
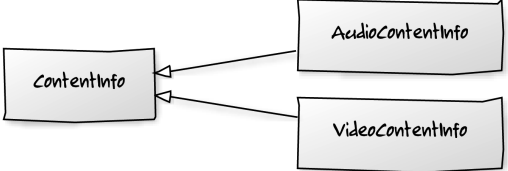


## A UML Primer

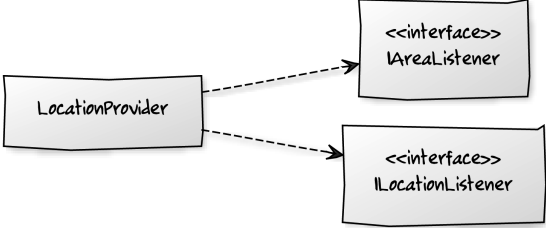
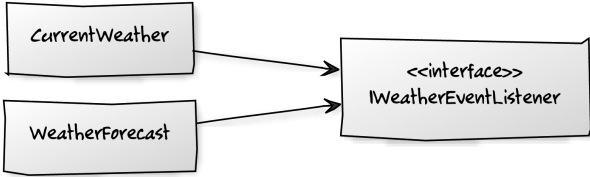
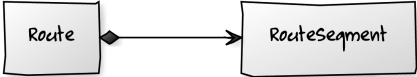
This appendix gives an explanation of the subset of UML notation that we've used in the book.

## B.1 Class Diagrams

<p><b>Interface inheritance</b></p> <p>Queue derives from and implements the interface defined by ICollection. See Chapter 4, 'bada Fundamentals', for more information about interfaces.</p>	 <pre> classDiagram     class ICollection {         &lt;&lt;interface&gt;&gt;     }     class Queue     Queue .. &gt; ICollection             </pre>
<p><b>Inheritance</b></p> <p>Both AudioContentInfo and VideoContentInfo derive from ContentInfo.</p>	 <pre> classDiagram     class ContentInfo     class AudioContentInfo     class VideoContentInfo     ContentInfo &lt; -- AudioContentInfo     ContentInfo &lt; -- VideoContentInfo             </pre>

(continued)



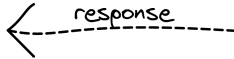

*(continued)*

<p><b>Dependencies</b></p> <p>LocationProvider uses IAreaListener and ILocationListener to provide location updates.</p>	 <pre> classDiagram     class LocationProvider     class IAreaListener["&lt;&lt;interface&gt;&gt; IAreaListener"]     class ILocationListener["&lt;&lt;interface&gt;&gt; ILocationListener"]     LocationProvider ..&gt; IAreaListener     LocationProvider ..&gt; ILocationListener           </pre>
<p><b>Simple association</b></p> <p>CurrentWeather and WeatherForecast are connected to IWeatherEventListener in some way; the classes work together. The application requests a list of WeatherForecast or the CurrentWeather and this will be returned using the IWeatherEventListener.</p>	 <pre> classDiagram     class CurrentWeather     class WeatherForecast     class IWeatherEventListener["&lt;&lt;interface&gt;&gt; IWeatherEventListener"]     CurrentWeather --&gt; IWeatherEventListener     WeatherForecast --&gt; IWeatherEventListener           </pre>
<p><b>Composition</b></p> <p>Route contains one or more RouteSegment. If the diamond is not filled in, this represents aggregation, a 'has a' relationship.</p>	 <pre> classDiagram     class Route     class RouteSegment     Route *--&gt; RouteSegment           </pre>

**Note:** The UML class diagrams in Chapter 6 were created using the excellent free online tool [yUML](http://yuml.me), available at: [yuml.me](http://yuml.me).

## B.2 Sequence Diagrams

Sequence diagrams show how objects and processes interact with each other over a timeline – the vertical line shown on the diagrams in Chapter 5, representing the lifetime of an object. In Chapter 5, we use sequence diagrams to show how the application interacts with the bada Server in response to user requests, such as to make their profile searchable, and how messages are sent between the objects involved in handling the request.

<p><b>An entity.</b> In this case the object MyBuddyService of class BuddyService.</p>	
<p>A <b>message</b> sent between objects.</p>	
<p>A <b>response</b> to a request.</p>	
<p>A <b>process</b> performed in response to a message.</p>	

**Note:** The sequence diagrams used in Chapter 5 were created using the free online web tool at [www.websequencediagrams.com](http://www.websequencediagrams.com).

