

一 监听器概念

- 1 在 **servlet** 技术中，预先定义了 8 大监听器来监听 **web** 开发过程中对应的事件
- 2 例如：创建、销毁 **Session**，向 **session** 写入数据，删除数据

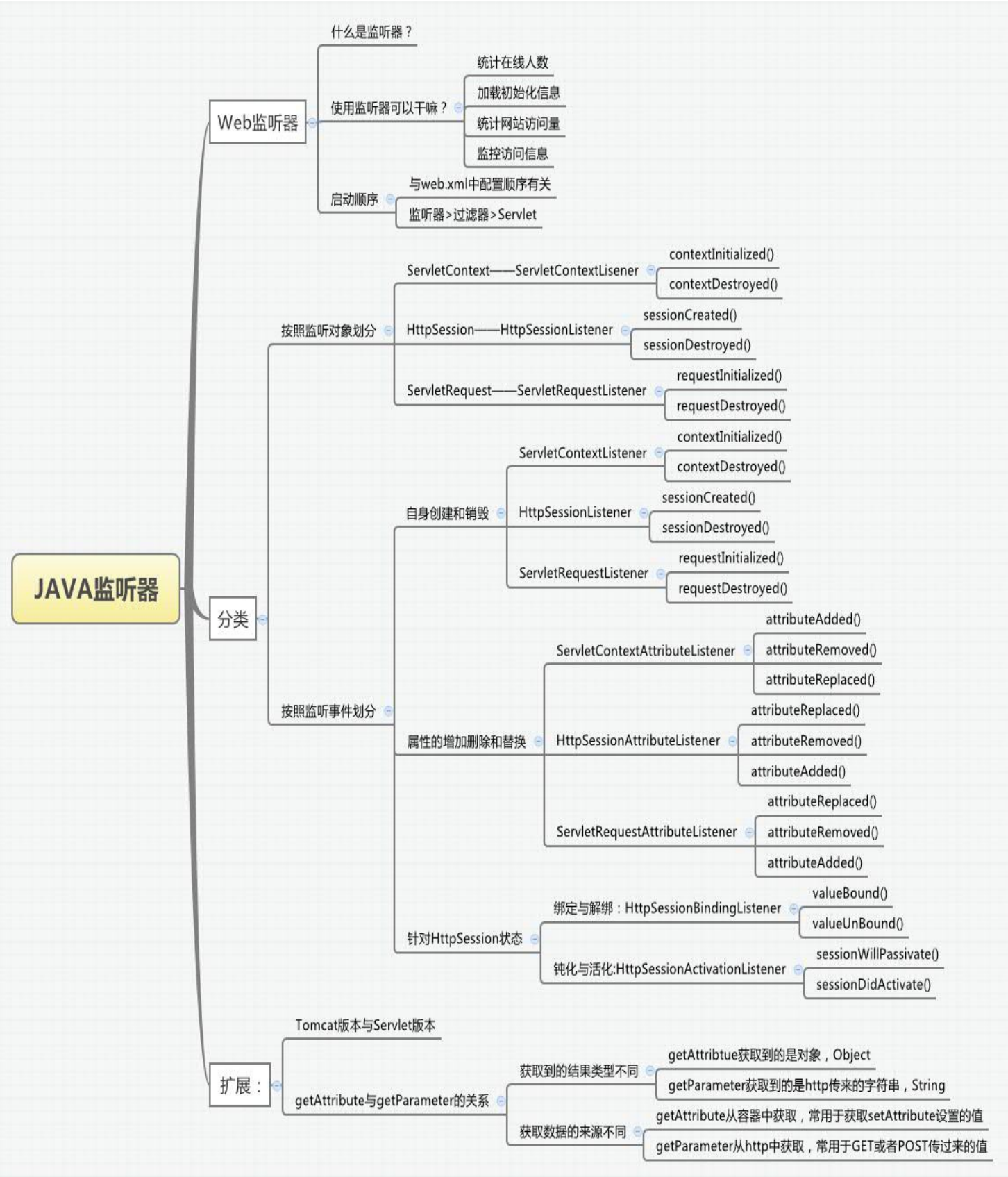
二 开发步骤

- 1 写一个类，实现监听器接口
- 2 在 **WEB.XML** 中配置监听器

三 监听器生命周期

和 **WEB** 应用一致，启动时创建，销毁时销毁

四 监听器分类



1 监听三大作用域的创建和销毁的监听器

1.A ServletContextListener 接口

1.A.1 监听 ServletContext 的创建与销毁

1.B ServletRequestListener 接口

1.B.1 监听 Request 的创建与销毁

1.C HttpSessionListener 接口

1.C.1 监听 Session 的创建和销毁

```
public class MyHttpSessionListener implements HttpSessionListener {  
    @Override  
    public void sessionCreated(HttpSessionEvent httpSessionEvent) {  
  
    }  
  
    @Override  
    public void sessionDestroyed(HttpSessionEvent httpSessionEvent) {  
        |  
    }  
}
```

1.C.2 配置

```
    <uri-pattern>/"/</uri-pattern>  
    </filter-mapping>-->  
    <listener>  
        <listener-class>cn.shu.blog.listener.MyHttpSessionListener</listener-class>  
    </listener>  
</web-app>
```

2 监听三大作用域属性被添加或销毁的监听器

2.A ServletContextAttributeListener 接口

2.A.1 监听 ServletContext 的属性创建与销毁

2.B ServletRequestAttributeListener 接口

2.B.1 监听 Request 的属性创建与销毁

2.C HttpSessionAttributeListener 接口

2.C.1 监听 Session 的属性创建和销毁

3 监听和 session 相关的二个监听器

3.A HttpSessionBindingListener

用来监听对象绑定搭配 Session 的监听器

3.B HttpSessionActivationListener

用来监听 Session 被钝化和活化的监听器