**首先介绍浏览所需的数据结构：**

**BrowserContext类**：保存浏览过程中所需要的上下文。其运行在UI线程中,所有方法均只能在UI线程中调用。

主要方法简介：

l  net::URLRequestContextGetter\* GetRequestContext()

获取URL请求所需要的URLRequestContext结构

l  ResourceContext\* GetResourceContext()

获取资源加载时需要的数据

**SiteInstance类**：站点对象,具体实现类为SiteInstanceImpl。一个SiteInstance对象对应于相同域名段的站点实例,如www.baidu.com和zhidao.baidu.com,其站点为baidu.com。不同的域名段有不同的SiteInstance对象，

这些SiteInstance对象都在BrowsingInstance注册备案. SiteInstance对象采取引用计数的方式控制生命周期。

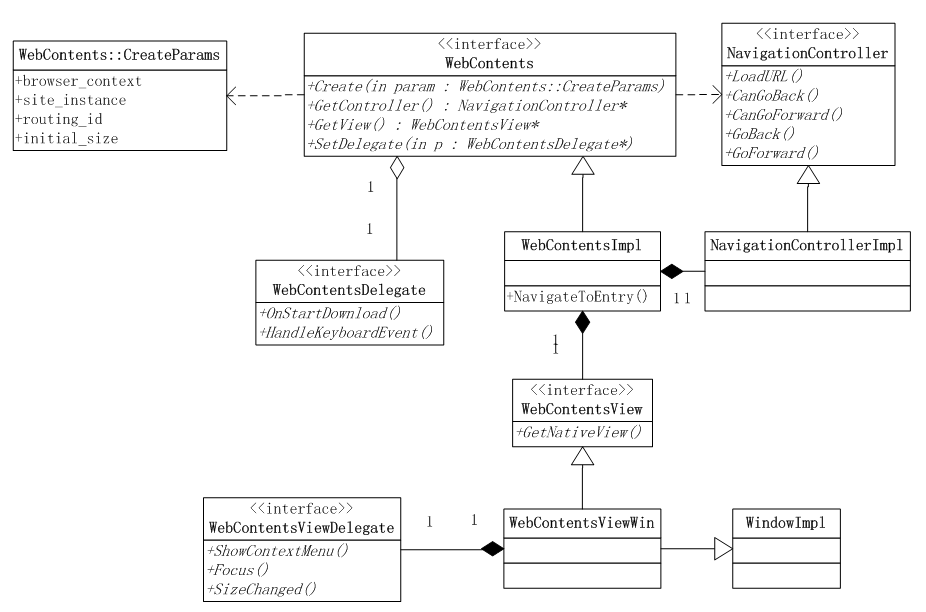
**BrowsingInstance类：**理论上与BrowserContext一一对应,维护浏览期间所产生的SiteInstance对象。

BrowsingInstance内部维护一个站点URL(如:baidu.com)到SiteInstance对象的映射，并提供注册/取消注册、查找的接口。相当于SiteInstance的cache对象。

BrowsingInstance对象为诸多SiteInstance对象所共享,同样采用引用计数的方式控制生命周期

**WebContents类对应于一页面窗口**

其结构图大致如下:



创建WebContents需要填充WebContents::CreateParam参数。在CreateParam参数中需要指定: BrowserContext对象、

SiteInstance对象、routing\_id(用于路由消息的发送目的地、initial\_size指定初始大小、context暂时没用上。

WebContentsDelegate接口：用于截获WebContents的更改通知以及提供相关的定制功能

NavigationController接口：负责页面的导航逻辑：前进、后退、刷新等。其内部维护一个导航过的NavigationEntry列表

WebContentsView接口:抽象WebContents的view界面部分，其windows下最终的实现类为WebContentsViewWin

WebContentsViewDelegate类:用于定制网页界面的扩展功能,比如显示右键菜单、焦点管理、尺寸调整等。