## URL类的使用

```java
import java.net.MalformedURLException;
import java.net.URL;
public class Test5 {
    public static void main(String[] args) throws MalformedURLException
        URL u = new URL("http://www.google.cn:80/webhp#aa?canhu=33");
        System.out.println("获取与此url关联的协议的默认端口：" + u.getDefault
        System.out.println("getFile:" + u.getFile()); // 端口号后面的内容
        System.out.println("主机名：" + u.getHost()); // www.google.cn
        System.out.println("路径：" + u.getPath()); // 端口号后，参数前的内容
        // 如果www.google.cn:80则返回80.否则返回-1
        System.out.println("端口：" + u.getPort());
        System.out.println("协议：" + u.getProtocol());
        System.out.println("参数部分：" + u.getQuery());
        System.out.println("锚点：" + u.getRef());

        URL u1 = new URL("http://www.abc.com/aa/");
        URL u2 = new URL(u, "2.html"); // 相对路径构建url对象
        System.out.println(u2.toString()); // http://www.abc.com/aa/2.ht
    }
}
```

## 【示例12-6】最简单的网络爬虫

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;

public class Test6 {
    public static void main(String[] args) {
        basicSpider();
    }
    //网络爬虫
    static void basicSpider() {
        URL url = null;
        InputStream is = null;
        BufferedReader br = null;
        StringBuilder sb = new StringBuilder();
        String temp = "";
        try {
            url = new URL("http://www.baidu.com");
            is = url.openStream();
            br = new BufferedReader(new InputStreamReader(is));
            /*
             * 这样就可以将网络内容下载到本地机器。
             * 然后进行数据分析，建立索引。这也是搜索引擎的第一步。
             */
            while ((temp = br.readLine()) != null) {
                sb.append(temp);
            }
            System.out.println(sb);
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
            try {
                is.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```