

# 接口测试自动化基础

## 接口自动化概述

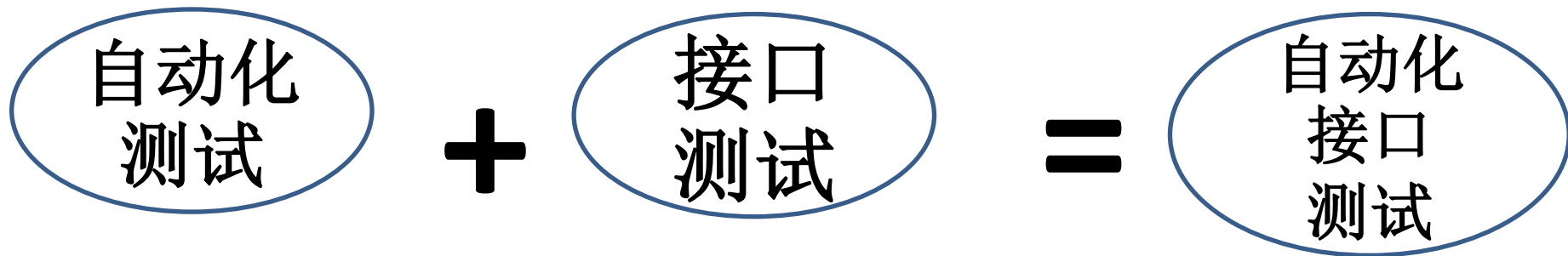
# 本章大纲

- 接口自动化概述
- HttpClient
- 引入测试框架

# 接口自动化概述



# 接口自动化概述



# 接口自动化概述

- 接口文档分析
- 接口用例设计
- 使用PostMan执行接口测试
- 使用HttpClient执行接口测试

# HttpClient 工具介绍

- HttpClient 是 **Apache** Jakarta Common 下的子项目，可以用来提供高效的、最新的、功能丰富的支持 HTTP 协议的**客户端编程工具包**，并且它支持 **HTTP** 协议最新的版本。
- <http://hc.apache.org/httpcomponents-client-ga/>
- <http://hc.apache.org/httpcomponents-client-ga/httpclient/apidocs/>

# HttpClient 环境搭建

- Java环境和Eclipse

- 下载:

- <http://www.oracle.com/technetwork/java/javase/overview/index.html>

- 安装+配置

- 验证: `javac -version`

- HttpClient环境

# HttpClient 开发过程

- 创建Java工程
- 新建libs库目录
- HttpClient 工具下载及引入
  - <http://hc.apache.org/downloads.cgi>
  - 工程中引入jar包



# 第一个例子

- 通过httpclient来访问下百度首页

```
CloseableHttpClient httpClient = null;
CloseableHttpResponse response = null;
try {
    //初始化httpClient对象
    httpClient = HttpClients.createDefault();
    //初始化HttpGet请求
    HttpGet httpGet = new HttpGet("https://www.baidu.com");
    //执行http请求,得到response
    response = httpClient.execute(httpGet);
    //打印http返回状态码
    System.out.println(response.getStatusLine());
    //用HttpEntity得到response返回的实体
    HttpEntity entity = response.getEntity();
    if(entity != null) {
        //打印body内容
        System.out.println(EntityUtils.toString(entity, "UTF-8"));
    }
    EntityUtils.consume(entity);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    response.close();
    httpClient.close();
}
```