

ACAA 认证 Web 应用架构工程师

ACAA Certified Web Application Architecture Engineer

考试大纲

【概述】

ACAA 认证 Web 应用架构工程师 大纲根据行业基础需求来定义,测试包括【正则表达式】

【移动 Web 开发】【前端模块化】【React 框架】【组件化开发】【Git 和 SVN 的使用】

六个部分,试题由 60 道单项选择题以及 2 道实践操作题组成,满分为 100 分,综合成绩

60 分为通过。其中选择题满分 40 分,24 分为通过; 操作题满分 60 分。

本测试考核内容包括:

- 正则表达式
- 移动 Web 开发
- 前端模块化
- React 框架
- 组件化开发
- Git 和 SVN 的使用

【考试目标】

本考试是基于 Web 应用架构工程师 领域。由本考试所认可的技能和知识是 ACAA 认证

Web 应用架构工程师 必须掌握的。

【测试内容】

在准备参加【ACAA Web 应用架构工程师】认证考试之前,请认真阅读下面的主要测试内容,以判断自己是否符合本课程测试目标并有能力通过认证考试,在考试过程中禁止参考任何资料。本大纲分为六个部分,括号内的数字表示该部分在本课程测试中所占的题数。

【考试内容大纲】

【选择题部分】

本测试考核内容包括:

正则表达式 (10 题)

难度等级: ★★

正则表达式是 Web 应用架构工程师必须掌握的核心技能之一,通过学习 JavaScript 正则表达式、事件,并且完成表单验证案例,你对 js 有更深入的理解,主要内容包含事件流、IE 事件处理、跨浏览器事件处理、事件对象与类型,正则表达式的创建、匹配字符串、转义、字符类、重复以及常用字符。

移动 Web 开发 (8 题)

难度等级: ★★★

移动端如今已经占据了 Web 开发的半壁江山,如今手机及网络普及率已远超过过去我们的想象,能在手机淘宝上买东西,想必很少会有人在网页淘宝上买,这就凸显出了移动端开发的重要性。在 4G 早已成熟,5G 蓄势待发的当下,H5 应用正在威胁原生应用的地位,且企业开发原生 APP 同时需要安卓和 IOS 两种开发人员的情况下,H5 开发学习成本和人力成本的优势逐步凸显,当下掌握移动端开发已经是一名前端开发人员的必备技能。

前端模块化 (12 题)

难度等级：★★

前端模块化是随着前端在 web 应用开发中重要程度日益凸显的产物，在发展过程中，CommonJS 使 JavaScript 模块化编程正式登上舞台，为解决 script 标签异步加载和 JS 加载过程中导致浏览器停止渲染的问题，AMD (对应 requireJS) 和 CMD (对应 SeaJS) 也应运而生，在 MVVM 框架横行的今天，掌握前端模块化可以帮助前端工程师更好地去理解使用框架。

React 框架 (10 题)

难度等级：★★★

React 是一个用于构建用户界面的 JavaScript 库，主要用于构建 UI，是 MVC 中的 V (视图)，起源于 Facebook 的内部项目，用来架设 Instagram 的网站，并于 2013 年 5 月开源。React 拥有较高的性能，代码逻辑简单，社区活跃，至今已经发展成熟，是大型企业级 web 应用的优秀选择，掌握 react 已经是一名前端工程师的必备技能。

组件化开发 (12 题)

难度等级：★★★★

程序员着手解决的问题之一就是提升开发效率，而作为一个前端架构师，掌握组件化开发是必不可少的要求，将企业级网站组件化，能极大提升开发效率，且能为以后的开发业务提供成熟的组件化解决方案，能否将企业级网站做到高内聚、低耦合组件化是判断一个前端架构师的重要能力之一。

Git、SVN 的使用（8 题）

难度等级：★★★

Git 和 SVN 是企业常用的版本控制系统，Git 是世界上最先进的分布式版本控制系统，而 SVN 在国内也被广泛使用，熟练掌握这两个版本控制系统，是一名合格的 Web 应用架构工程师所必备的技能。

ACAA 认证 Web 应用架构工程师操作题

【操作题部分】

前端架构开发（2 题）

考试形式基于 Web 应用架构操作，考试时间为 2 小时。总分 60 分。

（操作题部分推荐提前进行相关类型练习）

难度等级：★★★★

1 . React 中受控组件 (controlled component) 与不受控制的组件 (uncontrolled component) 有什么区别？请简单描述并写出 React 的 7 个生命周期，简述各个生命周期的调用时机。

测试要求

要求熟练掌握 React 组件的概念，熟练使用生命周期的钩子函数。

2. 请依次写出以下 git 命令：

- (1) 初始化创建 git 仓库；
- (2) 将修改放入暂存区；
- (3) 提交暂存区的修改；
- (4) 比较工作区与版本库里最新版本的差别；
- (5) 查看远程仓库地址；
- (6) 设置远程仓库地址；
- (7) 回退到上一个版本；
- (8) 创建并切换到 dev 分支。

测试要求：

熟练掌握常用git命令。