

TestStand 高级架构师认证 CTA 的考试范围

认证 ID	认证名称	工作描述:
CTA	TestStand 高级架构师认证	TestStand 高级架构师可根据一个给定的工业测试要求, 提供有效且全面的测试解决方案以满足各种复杂需求。

任务 ID	任务	考察目标 ID	考察目标
CTA-ASD-01	利用 TestStand 的模块化架构设计一个 TestStand 系统	CTA-ASD-01-01	说明 TestStand 架构中处理模型和回调函数的作用。
		CTA-ASD-01-02	说明 TestStand 程序员帮助文档中以下部分的相关知识: <ul style="list-style-type: none"> ▪ API 函数结构 ▪ TestStand 的属性及方法的相关知识
		CTA-ASD-01-03	根据给定的一系列材料及要求, 设计并实现相关的测试序列文件和代码模块。
		CTA-ASD-01-04	根据给定的 TestStand 项目要求, 确定以下 TestStand 组件的功能和属性, 设计最佳的解决方案: <ul style="list-style-type: none"> ▪ TestStand 引擎 ▪ 用户界面 <ul style="list-style-type: none"> ▪ 序列编辑器 ▪ 操作员界面 ▪ 处理模型 <ul style="list-style-type: none"> ▪ 处理模型回调函数 ▪ 序列文件 ▪ 模块适配器 ▪ 其他回调函数 <ul style="list-style-type: none"> ▪ 引擎回调函数 ▪ 前端回调函数 ▪ TestStand 应用程序接口函数 (API)
		CTA-ASD-01-05	根据给定的一系列情况及要求, 利用 TestStand 帮助系统获得需要的信息。
		CTA-ASD-01-06	使用给定类的对象获取另一个类的对象。
		CTA-ASD-01-07	根据给定的一系列条件和要求, 使用序列关联对象获取所需信息。
		CTA-ASD-01-08	说明查询字符串及其相关特性。
		CTA-ASD-01-09	根据给定要求, 创建一个正确的查询字符串, 以访问特定的数据。
		CTA-ASD-01-10	说明使用 ActiveX 自动化适配器从序列中调用

			TestStand API 的过程。
		CTA-ASD-01-11	使用 ActiveX 自动化适配器，从一个特定的序列中调用 TestStand API 完成给定的任务。
CTA-ASD-02	利用 TestStand 处理模型 (TestStand Process Models) 的设计特点，创建最佳的测试解决方案	CTA-ASD-02-01	说明处理模型在简化测试操作过程中的作用。
		CTA-ASD-02-02	说明在处理模型中放弃使用组件而选用模型回调函数的好处。
		CTA-ASD-02-03	说明使处理模型响应配置参数的方法及其好处。
		CTA-ASD-02-04	说明使用 TestStand 自带处理模型作为程序起点的好处。
		CTA-ASD-02-05	根据给出的一系列条件和要求，创建一个 TestStand 解决方案，使标准测试操作最简化。
		CTA-ASD-02-06	根据给定的条件和要求，选择合适的处理模型进行定制。
		CTA-ASD-02-07	根据给定的一系列 TestStand 项目条件和要求，运用现有和定制组件的优化组合，创建一个最优化的解决方案。
CTA-ASD-03	将回调函数整合在 TestStand 程序中，以便在特定的环境下执行操作	CTA-ASD-03-01	分析给定的一系列条件，以决定是否需要模型回调函数，如果需要，是使用已定义的函数还是新建一个函数。
		CTA-ASD-03-02	说明在 TestStand 系统设计中回调函数所带来的灵活性、模块化和控制功能。
		CTA-ASD-03-03	根据给出的一系列材料及要求，在 TestStand 项目中设计并实现相应的回调函数。
		CTA-ASD-03-04	根据特定测试序列生成回调函数模型，以便定制测试程序。
		CTA-ASD-03-05	在测试序列中创建一个引擎回调函数来改变序列运行方式，从而满足测试需求。
		CTA-ASD-03-06	根据给定条件，创建一个引擎回调函数。
		CTA-ASD-03-07	说明使用前端回调函数的合适情况。
		CTA-ASD-03-08	根据给定的一系列条件和要求，将前端回调函数整合在 TestStand 解决方案中。
CTA-ASD-04	为 TestStand 应用创建用户自定义类型	CTA-ASD-04-01	说明在设计 TestStand 自定义类型的时候需要考虑的因素。
		CTA-ASD-04-02	解释自定义类型在 TestStand 中的封装过程。
		CTA-ASD-04-03	根据给出的 TestStand 定制类型及其相关应用程序，分析该定制类型在程序中的合理性。
		CTA-ASD-04-04	根据给定的一系列要求和材料，设计合适的步骤类型并加以应用。
		CTA-ASD-04-05	根据给定的针对某应用程序的一系列数据要求，创建一个适合的定制数据类型。
CTA-ASD-05	设计和实现	CTA-ASD-05-01	说明在设计操作接口的时候需要考虑的因素。

	TestStand 操作界面	CTA-ASD-05-02	<p>根据给定的要求，确定需要修改的操作员界面，并证明您的决定是正确的</p> <ul style="list-style-type: none"> ▪ 简单的操作员界面 ▪ 完整的操作员界面
		CTA-ASD-05-03	根据给定的一系列要求和材料，为某项 TestStand 的应用设计并实现一个合适的操作员界面。
		CTA-ASD-05-04	识别并说明 TestStand 定制操作员界面的模块组件。
		CTA-ASD-05-05	说明 TestStand 自带操作员界面的内部架构。
		CTA-ASD-05-06	根据给出的条件和要求，应用一个与处理模型通信的 TestStand 操作员界面。
		CTA-ASD-05-07	<p>说明以下通信的方法：</p> <ul style="list-style-type: none"> ▪ 处理模型到操作员界面的通信 ▪ 操作员界面到处理模型的通信
		CTA-ASD-05-08	根据给定的要求，修改默认的 TestStand 操作员界面，以便在多线程系统中显示信息。
CTA-ASD-06	在 TestStand 的定制过程中实现错误处理机制	CTA-ASD-06-01	说明默认的 TestStand 运行时错误处理机制。
		CTA-ASD-06-02	<p>解释下列运行时错误信息保存的地方：</p> <ul style="list-style-type: none"> ▪ 步骤错误 ▪ 序列错误
		CTA-ASD-06-03	区分致命和非致命错误，并说明当遇到这两种错误时会出现的结果。
		CTA-ASD-06-04	比较和区分 Step.Result.Error 和 RunState.SequenceError
		CTA-ASD-06-05	根据给定的 TestStand 应用，举例全面验证错误处理机制。
		CTA-ASD-06-06	根据给定的一系列要求和材料，在一个 TestStand 项目中设计并实现一个合适的错误处理机制。
CTA-ASD-07	在 TestStand 中设计并实现一个多线程应用程序	CTA-ASD-07-01	说明使用多线程测试系统的好处
		CTA-ASD-07-02	根据给定的一系列条件和系统要求，决定是否有必要使用多线程 TestStand 解决方案
		CTA-ASD-07-03	说明 TestStand 中相关的线程和执行对象
		CTA-ASD-07-04	区分同步和异步测试系统
		CTA-ASD-07-05	<p>说明在设计多线程 TestStand 系统时，为避免以下多线程问题而采用的方法：</p> <ul style="list-style-type: none"> ▪ 死锁 ▪ 优先级倒置 ▪ 线程
		CTA-ASD-07-06	根据给定的条件和要求，创建一个同步测试系统。

		CTA-ASD-07-07	根据给定的条件和要求，创建一个异步测试系统。
		CTA-ASD-07-08	根据给定的条件和要求，设计一个 TestStand 定制应用程序，该程式能正确实现多线程功能。
		CTA-ASD-07-09	<p>在一个给定的 TestStand 应用中，评估下列多线程问题出现的可能性：</p> <ul style="list-style-type: none"> ▪ 死锁 ▪ 优先级倒置 ▪ 线程饥饿
CTA-ASD-08	将良好的设计准则整合到 TestStand 应用中	CTA-ASD-08-01	根据给定的条件和要求，创建需求列表，以应用于解决方案的设计
		CTA-ASD-08-02	<p>详细阐述在设计处理模型时需要考虑的事项：</p> <ul style="list-style-type: none"> ▪ 用户的测试序列文件在运行时可能会使用模型回调函数修改处理模型。 ▪ 引擎回调函数会影响在处理模型中运行的所有用户序列文件。 ▪ 对用户测试序列文件时所使用处理模型的说明
		CTA-ASD-08-03	<p>详细阐述在 TestStand 应用程序中，步骤设计选项所需要考虑的事项：</p> <ul style="list-style-type: none"> ▪ 引擎回调函数可能影响各种用户序列中的所有步骤 ▪ 属性加载可能影响某一特定序列中的所有特性 ▪ 步骤特性可能影响特定类型的所有步骤
		CTA-ASD-08-04	根据给定的一系列要求，设计一个步骤类型，利用内置的 TestStand 功能，将外部代码的使用降到最低。
		CTA-ASD-08-05	<p>讨论以下场合通用的编程技巧：</p> <ul style="list-style-type: none"> ▪ SequenceContext ▪ ActiveX 对象句柄 ▪ 资源释放 ▪ 序列范例 ▪ 硬件独立 ▪ 镜像目标应用 ▪ 测试的定制特性和（或）需求 ▪ 错误检测
		CTA-ASD-08-06	<p>描述 TestStand 系统中以下对象所产生的效果</p> <ul style="list-style-type: none"> ▪ 文件结构 ▪ 相对与绝对路径 ▪ 步骤执行顺序

			<ul style="list-style-type: none"> ▪ 定制默认 TestStand 数据类型 ▪ 应用程序分配 ▪ 执行状态监控
		CTA-ASD-08-07	根据给定的设计目标，使用模块方法设计一个 TestStand 项目
		CTA-ASD-08-08	在 TestStand 项目中整合通用的编程技巧,包括: <ul style="list-style-type: none"> ▪ 代码重用 ▪ 合理利用 TestStand Executive ▪ 适合的错误处理 ▪ 在 TestStand 应用程序接口 (API) 中合理使用和重用对象 ▪ 为系统测试创建序列范例 ▪ 合理使用处理模型
CTA-ASD-09	为 TestStand 应用程序生成一个硬件接口层 (HIL)	CTA-ASD-09-01	区分基于功能和基于仪器的 HIL 设计
		CTA-ASD-09-02	解释在 TestStand 系统中整合 HIL 的好处
		CTA-ASD-09-03	以 IVI 步骤类型作为一个 HIL 范例来进行说明
		CTA-ASD-09-04	解释仪器对话管理器 (Instrument Session Manager) 的功能
		CTA-ASD-09-05	解释下列 HIL 设计中所考虑因素产生的影响: <ul style="list-style-type: none"> ▪ 不同层的驱动程序所需要的信息 ▪ 从底层到上层的设计策略 ▪ 相应的函数参数
		CTA-ASD-09-06	根据给定的一系列要求，设计一个以功能为基础的硬件接口层
		CTA-ASD-09-07	根据给定的一系列要求，设计一个以仪器为基础的硬件接口层
		CTA-ASD-09-08	根据给定的条件，将仪器对话管理器 (Instrument Session Manager) 整合到 TestStand 应用中