

1. ArkTS相对于TypeScript主要删除了哪些特性？

- A. 纯粹只与动态类型相关的特性 B. 静态类型相关特性 C. 在运行时具有重大开销的TypeScript特性
D. 所有上述

答案：AC

解析：ArkTS删除了那些纯粹只与动态类型相关或在运行时具有重大开销的TypeScript特性。

通过这样做，可以在ArkTS里实现完整的静态类型，提升性能

2. ArkTS和TypeScript的共同子集是指什么？

- A. 所有“好的”特性在标准TypeScript中与之完全相同。 B. 所有“不好的”特性在标准TypeScript中都被删除了。 C. ArkTS和TypeScript的所有功能。 D. ArkTS是TypeScript的一个子集。

答案：A

解析：见HarmonyOS开发中级篇P4

3. ArkTS的类型检查是怎样的？

- A. 可配置的，开发人员可以使其更严格或更放松。 B. 有风险的，如果我们错误配置了类型检查器，那么静态类型带来的好处都将丧失。 C. 完全静态的，不能放松或关闭类型检查。 D. 上述都不正确。

答案：C

解析：见HarmonyOS开发中级篇P5

4. TypeScript编译后会丢失什么信息？

- A. 类型信息 B. 变量名称 C. 函数名称 D. 上述都不正确。

答案：A

解析：见HarmonyOS开发中级篇P5

5. 在ArkTS中，以下哪个运算符是不被支持的？ A. + B. - C. ~ D. delete

答案：D

解析：ArkTS中，对象布局在编译时就确定了，且不能在运行时被更改。因此，删除属性的操作没有意义。 [详见从TypeScript到ArkTS适配规则](#)

6. ArkTS相对于TypeScript的主要优势是什么？

A. 更严格的类型检查 B. 编译后的类型信息保留 C. 更少的运行时类型检查 D. 更快的编译速度

答案：C

解析：见HarmonyOS开发中级篇P5

7.在ArkTS中，以下哪个特性是显式标注的？ A. 泛型函数类型实参 B. 对象字面量的类型 C. 数组字面量元素类型 D. 以上都是显式标注的

答案：D

解析：[详见从TypeScript到ArkTS适配规则](#)

8. TypeScript的类型信息在编译后是否保留？

A. 是的，保留在代码中。 B. 不是，类型信息在编译后丢失。 C. 取决于开发人员的配置。 D. 以上都不正确。

答案：B

解析：见HarmonyOS开发中级篇P5

9.在ArkTS中，以下哪个特性是不支持的？ A. 类型推断 B. ArkUI之外的装饰器 C. 泛型 D. 接口

答案：B

解析：ArkTS除了ArkUI中的装饰器，不允许使用其他装饰器[详见从TypeScript到ArkTS适配规则](#)

10.在ArkTS中，以下哪个特性是强制要求的？ A. 类型推断 B. 强制使用静态类型 C. 禁止在运行时变更对象布局 D. 以上都是强制要求的

答案：D

[详见从TypeScript到ArkTS适配规则](#)

11. ArkTS是哪种操作系统的优选主力应用开发语言？

A. HarmonyOS B. iOS C. Android D. Windows

答案：A

解析：ArkTS是HarmonyOS操作系统的优选主力应用开发语言

12. 在ArkTS中，与UI相关联的数据可以在哪些范围内传递？

A. 组件内使用 B. 不同组件层级间传递（如父子组件、爷孙组件） C. 应用全局范围内传递 E. 上述所有

答案：E

解析：ArkTS提供了多维度的状态管理机制。在UI开发框架中，与UI相关联的数据可以在组件内使用，也可以在不同组件层级间传递，比如父子组件之间、爷孙组件之间，还可以在应用全局范围内传递或跨设备传递。另外，从数据的传递形式来看，可分为只读的单向传递和可变更的双向传递。开发者可以灵活地利用这些能力来实现数据和UI的联动。[详见初识ArkTS语言](#)

13. ArkTS提供了哪些渲染控制的能力？

A. 条件渲染 B. 循环渲染 C. 数据懒加载 D. 上述所有

答案：D

解析：ArkTS提供了渲染控制的能力。条件渲染可根据应用的不同状态，渲染对应状态下的UI内容。循环渲染可从数据源中迭代获取数据，并在每次迭代过程中创建相应的组件。数据懒加载从数据源中按需迭代数据，并在每次迭代过程中创建相应的组件。[详见初识ArkTS语言](#)

14.在UI开发方面，ArkTS相对于TypeScript进行了扩展，增加了声明式UI描述、自定义组件和动态扩展UI元素的能力。以下哪个选项描述了ArkTS相对于TypeScript在UI开发方面的主要扩展？

A. 基本语法 B. 状态管理 C. 渲染控制 D. 上述所有

答案：D

[详见初识ArkTS语言](#)

15.在ArkTS中，哪些特性被完全支持，无需任何修改？ A. 类型推断 B. 泛型 C. 接口 D. 所有上述特性

答案：D

解析：根据测试，对于已遵循最佳TypeScript实践的项目，代码库中90%到97%的内容可以保持原封不动，包括类型推断、泛型、接口等特性。[详见从TypeScript到ArkTS适配规则](#)

16.在ArkTS中，以下哪个关键字不能用于声明变量？ A. let B. const C. var D. 以上都可以

答案：C

解析：ArkTS必须使用关键字 `let` 代替 `var` 来声明变量。[详见从TypeScript到ArkTS适配规则](#)

17.在ArkTS中，以下哪个特性是不支持的？ A. 使用any类型 B. 类型推断 C. 接口 D. 泛型

答案：A

解析：不支持 `any` 类型，所有使用 `any` 的代码都需要引入显式类型。[详见从TypeScript到ArkTS适配规则](#)

18.ArKTS在设计时考虑的重要目标是什么？ A. 提高运行速度 B. 提高代码可读性 C. 降低编译时间 D. 强制使用静态类型

答案：D

解析：ArkTS强制使用静态类型,静态类型是ArkTS最重要的特性之一。[详见从TypeScript到ArkTS适配规则](#)

19.在ArkTS中，使用静态类型的主要原因是什么？ A. 提高代码可读性 B. 提高运行速度 C. 减少编译时间 D. 强制使用静态类型作为语言特性

答案：B

解析：[详见从TypeScript到ArkTS适配规则](#)

20.在ArkTS中，以下有无哪个特性是支持的？ A. 支持JSX表达式 B. 支持解构赋值 C. 支持在catch语句标注类型 D. 以上都是不支持的

答案：D

[详见从TypeScript到ArkTS适配规则](#)