鸿蒙高级认证题库

- 官方地址
  基础认证
  高级认证
  题目至2024.7.25号 单选正确率90%，多选正确率80%！
  官方题目也在不断更新，难度不断上升。
  试题内容一共60题，40题单选、20题多选。题库估计加起来有200道。

HarmonyOS应用开发者高级认证 题库

- 单选题

7.如果要实现Row组件内的子元素均匀排列，且第一个元素与行首对齐，最后一个元素与行尾对齐，需要使用justifyContent的哪个枚举值（A）
A. SpaceBetween
B. End
C.Start
D. SpaceEvenly

5.下面的配置存在有几处错误(1处)
module.json5配置文件:

```
'module":{
  “name”: “entry”
  “abilities”:[
  {
    “name”:“EntryAbility”“srcEntry”:./ets/entryability/EntryAbility.ets"
    .··“skills”:[
    f
    entities":["entity.system.home
    ,
    “actions”:
    "ohos.want.action.home

    entities":
    "entity.system.home
    “actions”:[
    "ohos.want.action.home

    “metadata”:[

    “name”:
    “ohos.entry.shortcuts”
    “resource”:
    “p r o f i l e : s h o r t c u t s c o n f i g " 在 / r e s o u r c e s / b a s e /
    c
    onfig"在/resources/base/profile/日录下配置shortcuts
    c

    onfig.json配置文件:shortcuts":[
    ,
    shortcutId":"label":"shortcutLabel"icon":"media:shortcutIcon”
    “wants”:[
    “bundleName”:“com.ohos.hello”
    “moduleName”:
    “abilityName”:
    EntryAbility
```

D.1

6下面的配置存在有几处错误（2处）

app.json5配置文件：

```
{
  “app”: {
  “bundleName”: “com.example.myapplication”,
  “vendor”: “example”,
  “versionCode”: 1000000,
  “versionName”: “1.0.2”,
  “icon”: “m e d i a : a p p i c o n " , " l a b e l " : " media:app_icon", "label":
  i

  con","label":"string:app_name”,
  “bundleType”: “app”
}
}
```

module.json5配置文件：

```
{
  “module”: {
  “name”: “feature”,
  // …
  “atomicService”: {

    "preloads":[

    {
      "moduleName":"feature"
    }
    ]
  }

}
}
```

D.2

1. 为了使isShow参数值与半模态界面的状态同步，可以使用下列那种方式双向绑定isShow参数？（C $$)

A、&&

```
@Entry
@Component
struct SheetTransitionExample {
  @State isShow: boolean = false
  @State isShow2: boolean = false
  @State sheetHeight: number = 300;

  @Builder
  myBuilder() {
    Column() {
      Button("change height")
        .margin(10)
        .fontSize(20)
        .onClick(() => {
          this.sheetHeight = 500
        })
    }
    .width('100%')
    .height('100%')
  }

  build() {
    Column() {
      Button("transition modal 1")
        .onClick(() => {
          this.isShow = true
        })
        .fontSize(20)
        .margin(10)
        .bindSheet(&&this.isShow,this.myBuilder(),{
          height:this.sheetHeight
        })
    }
    .justifyContent(FlexAlign.Center)
    .width('100%')
    .height('100%')
  }
}
```

B、@@

```
@Entry
@Component
struct SheetTransitionExample {
  @State isShow: boolean = false
  @State isShow2: boolean = false
  @State sheetHeight: number = 300;

  @Builder
  myBuilder() {
    Column() {
      Button("change height")
        .margin(10)
        .fontSize(20)
        .onClick(() => {
          this.sheetHeight = 500
        })
    }
    .width('100%')
    .height('100%')
  }

  build() {
    Column() {
      Button("transition modal 1")
        .onClick(() => {
          this.isShow = true
        })
        .fontSize(20)
        .margin(10)
        .bindSheet(@@this.isShow,this.myBuilder(),{
          height:this.sheetHeight
        })
    }
    .justifyContent(FlexAlign.Center)
    .width('100%')
    .height('100%')
  }
}
```

C、$$

```
// xxx.ets
@Entry
@Component
struct SheetTransitionExample {
 @State isShow: boolean = false
 @State isShow2: boolean = false
 @State sheetHeight: number = 300;

 @Builder
 myBuilder() {
   Column() {
     Button("change height")
       .margin(10)
       .fontSize(20)
       .onClick(() => {
         this.sheetHeight = 500
       })
   }
   .width('100%')
   .height('100%')
 }

 build() {
   Column() {
     Button("transition modal 1")
       .onClick(() => {
         this.isShow = true
       })
       .fontSize(20)
       .margin(10)
       .bindSheet($$this.isShow,this.myBuilder(),{
         height:this.sheetHeight
     })
   }
   .justifyContent(FlexAlign.Center)
   .width('100%')
   .height('100%')
 }
}
```

D、

```
// xxx.ets
@Entry
@Component
struct SheetTransitionExample {
  @State isShow: boolean = false
  @State isShow2: boolean = false
  @State sheetHeight: number = 300;

  @Builder
  myBuilder() {
    Column() {
      Button("change height")
        .margin(10)
        .fontSize(20)
        .onClick(() => {
          this.sheetHeight = 500
        })
    }
    .width('100%')
    .height('100%')
  }

  build() {
    Column() {
      Button("transition modal 1")
        .onClick(() => {
          this.isShow = true
        })
        .fontSize(20)
        .margin(10)
        .bindSheet(this.isShow,this.myBuilder(),{
          height:this.sheetHeight
      })
    }
    .justifyContent(FlexAlign.Center)
    .width('100%')
    .height('100%')
  }
}
```

8、根据上面代码，以下解释正确的是 （D）

注：被注释的代码是额外加上的，用于测试，并非题目原本的

```
enum Mode {
  fullScreen,
  halfScreen
}

@Entry
@Component
struct Index {

  @State title: string = "";
  @State mode: Mode = Mode.fullScreen;

  isShowTitle(): boolean {
    if (this.mode == Mode.fullScreen) {
      this.title = "Title";

      return true;
    }else {
      this.title = "Section";
      return false;
    }
  }

  build() {
    Column(){
      if (this.isShowTitle()) {
        Text(`${this.title}`)
      } else {
        Text(`${this.title}`)
      }
      ChangeMode({ mode: this.mode})
    }
  }
}

@Component
struct ChangeMode {
  //@Link mode: Mode;
  @Prop mode: Mode;
  build() {
    Row({space: 20}) {
      Button('full screen').onClick(() => {
        this.mode = Mode.fullScreen
```

```
        })
        Button('half screen').onClick(() => {
          this.mode = Mode.halfScreen
        })
        //Text(this.mode.toString())
      }
    }
  }
```

A、本例子可以运行起来，所以代码没有问题。

B、在ChangeMode里改变mode的值，会触发其父组件Page的Title内容的切换。

C、为了避免@Prop的拷贝，可以优化使用@Link，在该例子中行为和@Prop一样。

D、在自定义组件Page的build方法里改变状态变量是非法操作，可能导致未定义的异常UI行为。

11、依次点击A、B、C、D四个按钮，其中不会触发UI刷新的是：(C)

```
class Info{
  name: string;

  constructor(name: string) {
    this.name = name;
  }
}

@Entry
@Component
struct Index {
  @State nameList: Info[] = [new Info("Tom"), new Info("Bob"), new Info("John")]

  build() {
    Column() {
      ForEach(this.nameList, (item: Info) => {
        Text(`${item.name}`)
      })
      Button("A")
        .onClick(() => {
          this.nameList.push(new Info("Lucy"))
        })
      Button("B")
        .onClick(() => {
          this.nameList[0] = new Info("Eric")

        })
      Button("C")
        .onClick(() => {
          this.nameList[0].name = "Jim"
        })
      Button("D")
        .onClick(() => {
          this.nameList = [new Info("Barry"), new Info("Cindy"), new Info("David")]

        })
    }
  }
}
```

A、A

B、B

C、C

D、D

1、下面关于方舟字节码格式PREF_IMM16_v8_v8描述正确的是（）

16位前缀操作码，16位立即数，2个8位寄存器

2、ARKTS支持以下哪个函数？(C)

A.Object.getOwnPropertyDescriptor();

B.Object.getOwnPropertyDescriptors();

C.Object.values();

D.Object.hasOwnProperty();

3、下面关于方舟字节码格式IMM16_ID16_IMM8描述正确的是 （7.15更新）

8位操作码，16位立即数，16位id，8位立即数

4、以下哪些赋值语句在arkts中是合法

value2 let value2:string|null=null

5、Text组件不支持以下哪种ABCD代码中哪种使用方式？（C） 7.15更新 [存疑B]

A：

```
@Entry

@Component

struct TextExample{

        build(){

                Column({space:8}){

                        Text('textshadow').fontsize(9).fontcolor(0xcccccc)
                }
        }
}
```

B:

```
@Entry

@Component

struct SpanExample{build(){Flex({ direction: FlexDirection.Column, alignItems: ItemAlig

Text(){Span('In Line')Span('Component')Span('!')}}..width('100%').height(250).padding({

}}
```

C:

```
@Entry

@Component

struct styledstringDemo{scroll:Scroller =new Scroller();

layout:TextLayoutManager =new TextLayoutManager();

controller1:TextController =new TextController();

async onPageShow(){this.controller1.setLayout(this.layout)}



build(){Column(){Text(undefined,{controller:this.controller1 })}.width('100%')}}
```

D:

```
@Entry

@Component

struct styledStringDemo {

scroll:Scroller =new Scroller();

mutableStyledString: Mutablestyledstring = new Mutablestyledstring("test hello world",

start:0,length:5,

styledKey:styledstringKey.FONT,

styledValue:new Textstyle({fontColor:Color.Pink })}]);

controller1:TextController=new TextController();



async onPageshow(){

this.controller1.setstyledstring(this.mutablestyledstring)}



build(){

Column(){Text(undefined,{ controller: this.controller1 })}.width('100%')}}
```

6、以下哪个装饰器用来表示并发共享对象？（D）

A.@Style

B.@Shared

C.@State

D.@Sendable

7、以下关于ArkUI NavDestination组件的生命周期执行顺序中正确的是

A onWillappear->onAppear->onWillShow->onShow->onWillHide->onHidden->onWillDisappear->onDisappear

8、下面哪种转场效果在入场动画时，表现为从透明度为8、相对于组件正常显示位置x方向平移10evp的状态，

到默认的透明度为1、相对于组件不平移的状态，且透明度动画和平移动画的动画时长均为2000ms?(B) 7.15更新

A.TransitionEffect.asymmetric(TransitionEffect.OPACiTY.animation({duration:2000}),TransitionEffect.translate({x:100}).animation({duration: 2000}))

B.TransitionEffect.OPACITY.animation({duration: 2000)).combine(TransitionEffect.translate({x:100}))

C.TransitionEffect.OPACITY.combine(TransitionEffect.translate({x:100}).animation({duration: 2000}))

D.TransitionEffect.translate({x:100}).combine(TransitionEffect.OPACITY.animation({duration: 2000}))

9、使用promptAction.showToast如何设置显示在其他应用之上?

toastshowmode.TOP_MOAST

10、依次点击ABCD四个按钮，其中不会触发ui刷新的是（button AB） 7.15更新多选题

```
@Entry

@Component

struct Index {

 @State count:number=0;

 @State @Watch('onValueChange') value:number=50;

 onValueChange() {

  this.count = this.value

 }

 build() {

  Column(){

   Text(`${this.count}`)

   Button('A')

    .onClick(() => {

     this.count = 0

    })

   Button('B')

    .onClick(() => {

     for (let i = 0; i < 1000; i++) {

      this.count = i

     }

     for (let i = 1000; i > 0; i--) {

      this.count = i
```

```
      }
      this.count--
    })
  Button('C')
    .onClick(() => {
      this.value = 100
    })
  Button('D')
    .onClick(() => {
      setInterval(() => {
        this.count++
      },1000)
    })
  }
 }
}
```

11、从桌面冷启动如下应用，点击Change按钮5次，整个过程中，代码中的2条log依次出现的次数（1,0） 7.20更新

```
class Data {
  num: number
  type: string
  constructor(num: number,type:string)
  {this.num = num;
    this.type = type;}
}

@Reusable
@Component
struct Item {
  @State data: Data |undefined = undefined;
  aboutToAppear(): void {
    console.log(“Demo log1”)
  }

  aboutToReuse(params: ESObject): void {
    console.log(“Demo log2”);
    this.data=params.data
  }

  build() {
    Text(“num=”+this.data?.num+“,type=”+this.data?.type)
  }

}

@Entry
@Component
struct Test1Page {
  data1:Data=new Data(1,“type1”)
  data2:Data=new Data(2,“type2”)
  @State data:Data=this.data1
  build() {
    Column(){
      if (this.data.type==“type1”){
        Item({data:this.data}).reuseId(this.data.type)
      }else {
        Item({data:this.data}).reuseId(this.data.type)
      }
      Button('Change').onClick(()=>{
        if (this.data===this.data1) {
          this.data=this.data2
```

```
          }else{
            this.data=this.data1
          }
        })
      }
    }
  }
```

12、现有一个宽高分别为200px的xcomponent

A、offsetX为-20

offsetY为50

surfaceWidth为200

surfaceHeight为500

13、以下关于垂直滚动Grid组件使用cachedcount属性的说明正确的是（B） 7.15更新

B.设置cachedCount为1，则Grid在显示范围上下各缓存1个GridItem

14、在使用DevEco studio的profiler进行Harmonyos应用性能优化的流程中，以下哪个步骤最恰当地描述了开发者利用profiler工具进行性能问题识别、定位、优化及验证的完整过程

C.利用"Realtime Monitor"初步识别性能瓶颈，创建深度分析任务定位根因，根据分析结果优化代码，再用"Realtime Monitor"验证优化效果

15、一个复杂的项目，该项目不仅包含主入口模块（Entry Module)，还有多个特性的功能模块(Feature Modules/HSP)，并且这些模块间存在着相互调用关系。

为了确保在调试过程中能够完整地测试所有交互逻辑，需要将涉及到的所有模块的HAP包都部署到目标设备上。

请从以下选项中选择正确的操作步骤来配置DevEco Studio，以便一次性部署和调试项目中的多个模块

B进入“Run >Edit Configurations"菜单，在"Deploy Multi Hap"选项卡下，勾选"Deploy Multi HapPackages”，随后在列表中选择需要部署的模块。

16、当您开始开发一个应用/服务时，首先需要根据工程创建向导，创建一个新的工程，工具会自动生成对应的代码和资源模板。关于新建工程，下列选项说法正确的是?

A. Compatible SDK是兼容的最低API Version。

17、项目中包含多个模块和数千行代码。随着开发的深入，项目中的ArkTS源代码文件逐渐积累了大量import语句，其中不乏未使用的import以及不规范的排序情况，

关于DevEco studio的编辑器的“optimize Imports”，以下说法正确的是(A) 7.18更新

A. 可以在菜单栏中依次点击"code”>"Reformat code"来达到优化import的目的，因为"Optimize imports"功能已整合进"Reformat code" 中。

18、小李正在使用DevEco studio进行Harmonyos应用的开发工作，他需要对一个频繁被调用的函数calculateData()进行重构，为了帮助小李高效地找到calculateData()函数的所有引用位置，并确保重构时考虑周全，以下哪个步骤是正确的使用DevEco studio的"Find

Usages”功能的操作方法

A小李只需将光标定位在calculateData)函数名上，右键点击并选择"Find Usages”，或者直接使用快捷键Alt +F7(macOS为Option + F7) ,DevEco Studio会自动列出该函数在项目中的所有引用位置。

19、在使用DevEco studio的Profiler进行Harmonyos应用或服务内存管理优化时，以下哪个描述最准确地概述了“Allocation Insight”功能在识别和解决内存问题中的作用

A. Allocation Insight通过分析应用服务运行时的内存分配及使用情况，辅助定位内存泄漏、内存抖动和溢出问题，支持优化内存使用

20、在使用DevEco Studio的Profiler进行Harmonyos应用或服务性能分析时，面对应用出现卡顿、加载慢等性能瓶颈问题，以下哪个描述最贴切地说明了“Time场景分析任务”的功能及其对开发者优化流程的帮助

A. Time场景分析任务展示热点区域内的CPU和进程级调用栈耗时情况，支持代码跳转，助力开发者快速定位并优化耗时较长的代码段

21.开发者在编写ArkUI代码时，想要提前预览下所编写的组件的效果，下述哪个组件可以使用DevEco studio Previewer正常预览?

B.@Preview @Component struct TitleSample { @StorageProp('title'”) title: string = 'PlaceHolder'; build( ){Text(this.title)}}

22.want参数的entities匹配规则错误的是 A
A、调用方传入的want参数的entities为空，待匹配应用组件的skills配置中的entities不为空，则entities匹配失败。

B、调用方传入的want参数的entities为空，待匹配应用组件的skills配置中的entities为空，则entities匹配成功。

C、调用方传入的want参数的entities不为空，待匹配应用组件的skills配置中的entities为空，则entities匹配失败。

D、调用方传入的want参数的entities不为空，待匹配应用组件的skills配置中的entities不为空且包含调用方传入的want参数的entities，则entities匹配成功。

23.HAR(Harmony Archive)是Harmonyos提供的共享包，以下关于HAR的描述错误的是（D）

D HAR不支持使用page页面

24.在资源覆盖时，以下优先级排序正确的是(A)

A. AppScope>HAP包自身模块>dayjs模块>lottie模块

25.D hsp包编译后的产物是.hsp文件 7.20更新

26.根据代码，以下ABCD解释正确的是(D) 7.16更新

```
enum Mode {

fullScreen,halfScreen

}



@Entry

@Component

struct Page{

@State title: string =”@state mode:Mode = Mode.fullScreen;

isShownTitle(): boolean{if(this.mode == Mode.fullScreen){this.title = "Title";

return true;

} else {

this.title="Section";return false;}}



build(){

Column(){

if(this.isShownTitle()){Text(`${this.title}`)}else{

Text(`${this.title}`)}

ChangeMode({ mode: this.mode})

}}}



@component

struct ChangeMode {@Prop mode: Mode;build(){
```

```
Row({space:20}){
Button('full screen').onclick(()=>{this.mode = Mode.fullScreen;})Button('half screen').
})}}}
```

A本例子可以运行起来，所以代码没有问题，

B为了避免@Prop的拷贝，可以优化使用@Link，在该例子中行为和@Prop-样。

C在ChangeMode里改变mode的值，会触发其父组件Page的Title内容的切换

D在自定义组件Page的build方法里改变状态变量是非法操作，可能导致未定义的异堂UI行为.

27.一个应用的一个UIAblity，其exported字段配置为false，以下哪个场景可以拉起这个

C.UIAbility.caller应用在后台，申请了长时任务，有START ABILITY_FROM BACKGROUND权限;

28.以下关于ArkUI NavDestination组件的生命周期执行顺序中正确的是?(A)

A.onWillappear->onAppear->onWillShow->onShow->onWillHide->onHidden->onWillDisappear->onDisappear

B.onWillappear->onWillShow->onShow->onAppear->onWillHide->onHidden->onWillDisappear->onDisappear

C.onWillappear->onAppear->onWillShow->onShow->onWillHide->onWillDisappear->onHidden->onDisappear

D.onWillappear->onAppear->onWillShow->onShow->onWillDisappear->onWillHide->onHidden->onDisappear

29.singleton模式的UIAbility，在冷启动时生命周期的执行顺序是:(B)

B onCreate->onWindowStageCreate->onForeground

30.hiAppEvent提供的Watcher接口，()属性不配置，会导致编译报错，产生"ArkTS compiler Error"

D.name

31.作为一个应用开发者，想搭建运维平台，想在应用内定时读取当前的内存信息，可以通过(B)接口来实现。

B hiDebug 7.18修正

32.当使用状态变量进行ArkUI组件间数据通信的时候，如果两个组件间没有直接的嵌套关系（非父子和祖孙关系组件），

但是他们又属于同一页面，最佳的装饰器应该选用哪个？（C）7.12修正

A.@Provide+@Consume

B.@State+@Link

C.LocalStorage

D.AppStorage

33.当标记了@Reuseable的自定义组件实现了组件复用后，这个组件的复用范围是什么？

C标记了@Reuseable的自定义组件的父组件范围内

34.ArkuI组件复用的作用机制是减少了什么时间从而降低了丢帧率？7.17更新

A组件节点和对象的创建时间

35.下面持续集成描述哪项是错误的:

B.持续集成就是持续编译，二者异曲同工

36.在moduleA(HAP类型)中有一个图片名为image.png，在moduleB(HAR类型)也存在一个图片名为image.png，而moduleA依赖于moduleB，那么在moduleA的编译产物hap包中，image.png存在情况是:

A. 仅存在moduleA的image.png

37.以下关于应用架构技术选型说法不正确的是()

C.元服务和应用可以共用一个代码工程，采用多目标产物构建方式，构建出应用和元服务两个产物，用于上架。

38.某App有A、B、C、D四个团队分别负责Module A、Module B 、Module C和Module D四个业务模块。随着业务的发展。Module A需要跳转到Module B、Module C的页面:Module B需要跳转到Module C、Module D的界面，Module C需要跳转到Module A的界面，Module D需要跳转到Module B和Module C的界面。由于复杂的依赖关系，导致一旦有变化就需要知会各个团队，所以该团队的架构师想要解糯各个业务模块，以下哪些做法是不推荐的(C)

A. 采用RouterModule作为中介者并用动态import解载各个业多模块。

B. 可以采用Navigation作为页面导航根容器，将其放在entry中，其他Module 的页面作为Navigation的子页面。

C. 采用静态import方式引入对应跳转的页面。

D. 在RouterModule中采用路由表方式解聘各个业务授块

39.某App依赖了3个ohpm库，这3个库占用的体积都比较大。在App的技术架构中，有多个hap和多个hsp均依赖这3个库，为了减少app的首包大小，以下哪些做法是无效的?

A 将某些特性做成按需加载模块，若这3个ohpm仅在按需加载模块里面使用，则将其打包在按需加载模块中。

40.关于长时任务开发使用的接口是

B 使用startBackgroundRunning申请任务，使用stopBackgroundRunning取消任务

41.hiAppEvent提供的Watcher接口，需要订阅到OS的崩溃事件，正确的实现方式（A）

```
hiAppEvent.addWatcher({
 name: "watcher",
 appEventFilters: [
 {
  domain: hiAppEvent.domain.OS,
  names: [hiAppEvent.event.APP_CRASH]
 }
 ],
 onReceive: (domain: string, appEventGroups: Array<hiAppEvent.AppEventGroup>) => {
 }
 })
```

42.现有一个宽高分别为200px的XComponent组件，其绑定了一个XComponentController(xcController)，依次进行如下操作：

(1) xcController.setXComponentSurfaceRect({surfaceWidth: 150, surfaceHeight: 500})

(2) 设置XComponent组件的padding为{ top: 5px, left: 10px, bottom: 15px, right: 20px }

(3) 将XComponent组件大小改为300px×300px

(4) 给XComponent组件设置一个宽度为2px的边框

(5) xcController.setXComponentSurfaceRect({ offsetX: -20, offsetY: 50, surfaceWidth: 200, surfaceHeight: -100 })

之后，调用xcController.getXComponentSurfaceRect()的返回值为

选择 { offsetX: 75, offsetY: -100, surfaceWidth: 150, surfaceHeight: 500 } 7.16更新

43.以下关于Taskpool和Worker的描述正确的是（A）

A.TaskPool支持任务延时执行

B.开发者需要自行管理taskpool的数量及生命周期

C.Worker自行管理生命周期，开发者无需关心任务负载高低

D.TaskPool和Worker的任务执行时长上限都是无限制

44、以下a1\a2\a3\a4哪些赋值语句在ArkTS中是合法的？(a2)

```
class A{v:number =0;}

class B extends A{u:string ='';}

class c{

v:number =0;}

let a1:A = new C();

let a2:A= new B();

let a3: B = new A();

let a4: C= new B();
```

45、ArkTs支持以下哪些函数？ （C）(JS是都支持的)

A. Object.isPrototypeOf();

B. Object.getOwnPropertySymbols();

C. Object.keys();

D. Object.isExtensible();

46、张工正在使用DevEco studio进行一个复杂项目的开发工作，项目中包含了成千上万行代码且涉及众多模块。在重构代码的过程中，他意识到需要对一个核心类名进行更改，考虑到这个类在整个项目中被广泛引用，手动修改不仅耗时且容易出错。

基于DevEco studio提供的代码编辑功能，以下哪个描述最准确地概述了张工如何高效且安全地完成对类名的更改，同时确保整个项目中所有相关引用同步更新？（B 最长的）

A、张工需打开项目搜索功能，输入旧类名找到所有匹配项，逐一进行替换，完成更名操作。

B、张工选中需要更名的类名，使用快捷键Shift+F6或右键菜单Refactor ->Rename，在弹出框中输入新名称并选择替换范围后，点击“Refactor”完成更名操作，确保所有相关引用自动更新。

C、张工只需简单选中需要更名的类名，按下Delete键删除后直接输入新名称，DevEco Studio会自动识别并更新所有引用。

D、张工在代码编辑器中右键点击该类名，选择“Find Usage”，手动浏览所有引用位置并逐一修改为新名称。

47在一个包含多个模块(如entry、feature、service、library等)的大型Harmonyos应用项目中，如果某个模块feature对另外一个公共库模块1ibrary有依赖，

如何通过DevEcostudio正确配置项目依赖关系?(C)

A无需配置，直接在代码中编写import xxx from'library'

B在library的oh-package,json5文件的dependencies字段中配置feature的依赖

C在feature的oh-package.json5文件的dependencies字段中配置library的依赖

D在feature的build-profile.json5文件的dependencies字段中配置library的依赖

48在使用DevEco studio的Profiler进行Harmonyos应用或服务内存管理优化时，以下哪个描述最准确地概述了“Allocation Insight”功能在识别和解决内存问题中的作用?

(B)

A Allocation Insight详细展示应用运行时的每条语句柄分配记录，便于开发者逐一检查内存使用，但不提供内存泄漏的自动识别功能

B Allocation Insight通过分析应用服务运行时的内存分配及使用情况，辅助定位内存泄漏、内存抖动和溢出问题，支持优化内存使用

C Allocation Insight主要关注于内存碎片整理，减少内存分配的不连续性问题，对内存泄漏和溢出问题的检测不是其主要功能

D Allocation Insight仅提供内存分配总量的概览，帮助开发者宏观了解内存使用趋势，但对于具体泄漏或抖动问题无能为力

49项目需要为不同的设备形态(如手机、智能手表)提供定制化构建。请说明如何在DevEcostudio中设置不同的构建配置，以生成针对不同设备的hap包?(A)

A 在模块级别build-rofile.json5定义多个target，在每个target的config/deviceType中定义不同的设备类型

B 在工程级别build-profile.json5定义多个product，在每个product的config/distributionFilter中定义不同的设备类型

C 在模块级别build-profile,json5定义多个target，在每个target的config/distributionFilter中定义不同的设备类型

D 在工程级别build-profile.json5定义多个product，在每个product的config/deviceType中定义不同的设备类型

50下面的配置一共存在有几处错误？(2处错误) 7.16更新

module.json5配置文件:

```
{
"module":{"name": "entry",
"abilities":[{
"name": "EntryAbility",
"srcEntry":"./ets/entryability/EntryAbility.ets",
"skills":[
{
"entities":[ "entity.system.home"],
"actions":["ohos.want.action.home"]
]}],
"metadata":[{
"name": "ohos.entry.shortcuts",
"resource":"$profile:shortcuts_config"}
]}]}}
```

在/resources/base/profile/目录下配置shortcuts_config.json配置文件:

```
{
"shortcuts":[
{"shortcutId": "id_test1",
"label":"shortcutLabel",
"icon":"$media:shortcutIcon",
"wants":[{
"bundleName":"com.ohos.hello",
```

```
"moduleName": "entry",

"abilityName": "EntryAbility"

}]}]}
```

2处错误：

skills字段被错误地放置在了abilities数组的第一个元素的内部，它应该位于abilities数组外部，与abilities同级。

metadata层级也有问题。

51应用开发中使用的各类资源文件，需要放入特定子目录中存储管理，以下关于资源说法错误的是（A）

A rawfile目录，支持创建多层子目录，子目录名称可以自定义，文件夹内可以自由放置各类资源文件。目录中的资源文件会被编译成二进制文件，并赋予资源文件ID。

B resfile目录，应用安装后，resfile资源会被解压到应用沙箱路径，通过Context属性resourceDir获取到resfile资源目录后，可通过文件路径访问。

C stage模型多工程情况下，共有的资源文件放到AppScope下的resources目录。

D base目录是默认存在的目录，二级子目录element用于存放字符串、颜色、布尔值等基础元素，media、profile存放媒体、动画、布局等资源文件。

A中资源文件不会被编译成二进制文件并赋予资源文件ID

52通过aa工具拉起com.example.test的EntryAbility，并传参给EntryAbility，具体参数是number类型的[key1，1][key2,2]和string类型的[key3,teststring][key4,''],

下边ABCD哪个aa 命令是正确的？（D） 不带"号的 7.15更新

A aa start -b com.example.test -a EntryAbility --pi key1 1 --pi key2 2 --ps key3 testString --psn key4"

B aa start -b com.example.test -a EntryAbility --pi key1 1 key2 2 --ps key3 testString --psn key4

C aa start -b com.example.test -a EntryAbility --pi key1 1 --pi key2 2 --ps key3 testString --ps key4

D aa start -b com.example.test -a EntryAbility --pi key1 1 --pi key2 2 --ps key3 testString --psn key4

53应用发生崩溃，接口可以获取到崩溃时调用栈 （C）

A、hiDebug

B、hiTraceMeter

C、hiAppEvent

D、hiLog

54关于ArkUI的ForEach和LazyForEach，下列说法错误的是?(B)

AForEach和LazyForEach会根据定义的键值生成规则为数据源的每个数组项生成唯一键值，并创建相应的组件。

B长列表滚动场景，优先使用ForEach。

C当在滚动容器中使用了LazyForEach，框架会根据滚动容器可视区域按需创建组件，当组件滑出可视区域外时，框架会进行组件销毁回收以降低内存占用。

DLazyForEach需要配合cachedCount和组件复用一起使用，以达到性能的最优表现。

55为了提高性能，所以List组件支持懒加载，可以通过配置cachedcount属性设置缓存列表项的数量。当我们不设置List的属性cachedcount时，该属性的默认值是?

（B）

A.0

B.1

C.2

D.3

56当标记了@Reuseable的自定义组件实现了组件复用后，这个组件的复用范围是什么？（A）

A.标记了@Reuseable的自定义组件的父组件范围内

B.整个页面都可以复用

C.标记了@Reuseable的自定义组件的外层容器节点范围内

D.整个应用内都可以复用

57下面持续交付&持续部署描述哪个是正确的:(C) 7.19更新

A.持续交付可以随时随地部署到生产环境

B.持续交付(CD,Continuous Delivery):指的是，频繁的将软件的新版本，交付给质量团队或者用户，以供评审。如果评审通过，代码就进入生产阶段。

它强调的是，不管怎么更新，软件是随时随地可以交付的。

C.在持续交付实践中，要考虑处理故障回滚和紧急修复，以确保系统在出现问题时能够快速恢复和修复。

D.持续部署是将代码库中的任何更改都应该自动且快速地投入生产环境。持续部署等同于持续交付。

58以下关于应用架构技术选型说法不正确的是(A) HSP

A随着业务的发展，应用功能会越来越多，某些功能可以做成动态加载，动态加载的模块采用HAR工程来构建，方便复用和共享。

B对于初始版本的应用，功能比较简单，可以考虑采用单HAP加上多个HAR工程构建代码工程，

C元服务和应用可以共用一个代码工程，采用多目标产物构建方式，构建出应用和元服务两个产物，用于上架。

D一些应用的扩展能力，比如备份、服务卡片，可以采用ExtensionAbility做成单独的feature HAP包，独立分发。

59以下对系统兼容性的理解正确的是(C) A观点有点模糊 选了C

A安全法律法规等不可控因素会导致系统非兼容性变更，开发者需要积极适配

B系统能力都会保持绝对的兼容性，不能因为任何非兼容性的修改而导致开发者成本上升

C已发布的系统能力有可能会发生非兼容性变更，比如新增特性或修改问题导致的行为不兼容，这种情况下应用需要关注changelog并进行适配。

D应用不需要关注系统的兼容性变化，那都是系统开发人员需要关注的事情

60某App有A、B、C、D四个团队分别负责ModuleA、ModuleB、ModuleC和ModuleD四个业务模块，随着业务的发展，ModuleA需要跳转到ModuleB、ModuleC的页面，ModuleB需要跳转到ModuleC、ModuleD的界面，ModuleC需要跳转到ModuleA的界面，ModuleD需要跳转到ModuleB和ModuleC的界面。

由于复杂的依赖关系，导致一旦有变化就需要知会各个团队所以该团队的架构师想要解耦各个业务模块，以下哪些做法是不推荐的(D)

A可以采用Navigation作为页面导航根容器，将其放在entry中，其他Module的页面作为Navigation的子页面。

B采用RouterModule作为中介者并用动态import解耦各个业务模块。

C在RouterModule中采用路由表方式解耦各个业务模块。

D采用静态import方式引入对应跳转的页面。

61关于短时任务开发使用的接口是（D）

A使用startBackgroundRunning申请任务，使用stopBackgroundRunning取消任务

B使用publishReminder发布一个提醒类通知，使用cancelReminder取消一个指定的提醒类通知

C使用startWork申请任务，使用stopWork取消任务，使用getWorkStatus获取任务状态

D使用requestSuspendDelay申请任务，使用getRemainingDelayTime获取任务剩余时间

62如果想让outer button响应事件，hitTestBehavior该怎么配置 (C)

```
import promptAction from '@ohos.promptAction';
// xxx.ets
@Entry
@Component
struct HitTestBehaviorExample {
build() {
// outer stack
Stack() {
Button('outer button')
.onClick((event) => {
promptAction.showToast({ message:'click事件触发------++++2‘ });
})
// inner stack
stack() {
```

Button('inner button')

.onTouch((event) => {

promptAction.showToast({ message: 'click事件触发------1' });

})

A.HitTestMode.Default

B.HitTestMode.Block

C.HitTestMode.Transparent

D.HitTestMode.None

63 从桌面冷启动如下应用，点击change按钮5次，整个过程中，代码中的2条log依次出现的次数，最初和最后画面上显示的【num=?】的数字是

2,4 开头的 7.18更新
2,4,1,2 [存疑]

64-1 已知下列代码PageOne页面为navigation中的某一子页面，依次点击PageOne页面中toPageTwo按钮，PageTwo页面中toPageOne按钮，

此时点击get按钮获取全部名为name的NavDestination页面的位置索引为（[0,2]）

64-2 此时获取当前页面的路由栈数量为多少 （3个） 7.18更新

65 在Harmony0s应用开发中，当开发者遇到需要分析Release版本应用的崩溃或异常堆栈信息时，为了准确地将堆栈追踪信息对应到源代码的具体位置，以下哪个描述是正确的做法或理解？（B最长的）

A、开发者需手动将Release构建生成的so文件与源代码进行映射，配合第三方工具进行堆栈信息还原，虽然过程繁琐，但最终能定位到问题代码位置

B、DevEco Studio提供的Release应用堆栈解析功能，要求开发者在遇到问题时，需上传构建产物中的特定文件B(如so、source map、nameCache等)到指定工具或界面，借助这些文件辅助解析堆栈信息，实现从Release堆栈到源码的映射，便于快速定位问题

C、因为Release应用经过优化和去除Debug信息，直接从堆栈跟踪到源代码行号是不可能的，开发者只能依靠日志信息手工推测问题所在

D、DevEco Studio通过集成的Release应用堆栈解析功能，自动利用构建时产生的so文件、source map文件nameCache文件等，无需额外操作即可直接在Release应用中提供详细的源码级堆栈追踪信息

66、ARKTS支持以下哪个函数？（A）

Object.keys();或 Object.values()

Object.getOwnPropertySymbols();

Object.isPrototypeOf();

Object.isExtensible();

67、开发者张工想要高效地管理Harmonyos设备中的文件，包括查看文件列表、进行文件搜索、新建及删除操作，以及在设备与PC间传输文件，而无需使用命令行工具。

以下哪个选项最能准确概括张工能通过哪个工具直接在DevEco studio界面完成上述所有操作？（A）

A.DevEco Studio的Device File Browser

B.DevEco Studio的Terminal面板

C.DevEco Studio的Project Explorer

D.DevEco Studio的Log面板

68、在使用DevEco studio进行混合语言开发时，开发者小李通过Napi引用了Native 接口的文件(例如d.ts文件)。他想要直接从这些接口跳转到其对应的c/C++函数实现处进行代码审查。

请从以下选项中选择最合适的操作步骤来帮助小李实现这一目标。(D最长的)

A小李可以将鼠标光标置于想要查看实现的接口名称上，按下鼠标右键，在出现的上下文菜单中寻找并点击“Find Usages”(查找用法)，在结果中筛选出C/C++的实现。

B为了查看C/C++函数实现，小李必须先切换到DevEco Studio的C/C++开发环境视图，之后在项目的资源管理B.器中找到对应的C/C++源文件手动打开，才能查看函数代码

C小李应该打开包含TypeScript接口声明的文件，然后手动在项目中搜索相应的C/C++源文件，逐个检查以找到匹配的函数实现。

D在声明或引用了Native接口的文件中，比如d.ts文件，小李可以直接选中接口名称，右键点击并在弹出的菜单中选择“Go To>Implementation(s)”(转到>实现)。还可以使用快捷键Ctrl+Alt+B;如果是macOS用户，则使用Command+Option+B，直接跳转到对应的C/C++数实现位置。

69、使用DevEco studio进行复杂的跨设备功能开发与调试工作，期间频繁依赖本地模拟器来模拟多样化的设备环境。

在这样的背景下，以下关于DevEco studio本地模拟器所支持的规格与功能，哪一项描述是准确的?（C）7.15更新

A.本地模拟器和真机的能力没有任何差异，真机上可以支持的能力在模拟器上都可以

B本地模拟器当前不支持查看HiLog以及FaultLog

C本地模拟器上运行的应用无需进行签名，简化了调试过程。

D本地模拟器当前不支持单元测试框架和U测试框架的运行

70、当前您在开发一个ArkTs、stage模型的Harmonyos工程，关于当前ArkTS工程目录结构，下列选项说法错误的是? (A) 7.18更新 A是配置文件不是脚本

A build-profile.json5:应用级编译构建任务脚本。

B entry>src>main>module.json5:Stage模型模块配置文件，主要包含HAP的配置信息、应用在具体设备上的配置信息以及应用的全局配置信息。

C oh-package.json5:描述依赖配置，如:依赖覆盖(overrides)、依赖关系重写(overrideDependencyMap)和参数化配置(parameterFile)等

D AppScope>app.json5:应用的全局配置信息。

71、一个复杂的项目，该项目不仅包含主入口模块(EntryModule)，还有多个特性的功能模块(Feature Modules/HSP)，并且这些模块间存在着相互调用关系。

为了确保在调试过程中能够完整地测试所有交互逻辑，需要将涉及到的所有模块的HAP包都部署到目标设备上。

请从以下选项中选择正确的操作步骤来配置DevEco studio，以便一次性部署和调试项目中的多个模块。（A最长的）

A进入“Run> Edit Configurations"菜单，在“Deploy Multi Hap”选项卡下，勾选“Deploy Multi HapPackages”，随后在列表中选择需要部署的模块。

B直接点击运行按钮，DevEco Studio会弹出对话框询问需要部署哪些模块，从中选择需要的模块后开始调试。

C在项目结构界面手动选择每个模块，单独编译并逐一将生成的HAP包通过HDC命令推送到设备上。

D无需特殊配置，DevEco Studio会自动检测到项目中的所有模块依赖，并在每次调试运行时自动部署所有相关HAP包。

72、应用程序开发调试过程中，经常需要安装新应用进行调测，下面安装应用操作错误的是（A） 7.15更新

A. hdc install -p ohosapp.hap

B. bm install -p ohosapp.hap

C. bm install -p ohosapp.hap -r

D. bm install -p /data/app/

73、可以通过下面那个接口拉起导航类的垂域面板（B）

A. startAbilityForResult

B. startAbilityByType

C. startAbilityByCall

D. startAbility

74、某个应用开发了一个UIAbilityA，其启动模式是specified，并且对应的AbilityStage的实现如下：

```
import Abilitystage from'@ohos.app.ability.AbilityStage';
import type Want from '@ohos.app.ability.Want';
export default class MyAbilitystage extends Abilitystage {
this.instanceIndex =0;
onAcceptWant(want: Want):string{
if(want.abilityName === 'UIAbilityA'){
if(want.parameters && want.parameters.instanceKey === 'test'){
return 'test_instance_' + this.instanceIndex++;
}else { return'test_instance'; }}
return 'MyAbilitystage';}}
```

依次调用如下方法4次启动UIAbi1ityA，value分别是"test""test”"testA""testA"，则当前运行期UIAbility实例有几个?

```
function testSpecified(context,value){

let want: Want ={

deviceId:'',

bundleName:'com.samples.stagemodelabilitydevelop',

abilityName: 'UIAbilityA',

moduleName:'entry',

parameters:{

instanceKey: value

}

};

context.startAbility(want).then(()=>{

hilog.info(DOMAIN NUMBER, TAG, 'Succeeded in starting UIAbilityA.');

}).catch((err:BusinessError)=>{

hilog.error(DOMAIN_NUMBER, TAG, "Failed to start UIAbilityA. code is ${err.code})

})

}
```

3个实例。（两次testA应该只启动同一个实例，所以3和4，选了3）

75、作为应用开发者，你使用hiAppEvent订阅了崩溃事件。应用崩溃后，从onReceive接口返回的AppEventInfo中(D)属性可以获取崩溃调用栈信息。

A.name

B.eventType

C.domain

D.params

76、我们需要避免在逐帧调用的接口中执行耗时操作，下面哪个选项不属于上述的接口？（D）

A、onTouch

B、onScroll

C、onAreaChange简化了调试过程

D、aboutToReuse

77、以下关于HAP(Harmony Ability Package)说法正确的是(A)

A.应用工程如果包含多个Module，在应用i汤上架时，会将多个.hap文件打包成一个.app文件。

B.应用工程编出的app文件中，只能包含一个hap文件。

C.HAP是应用安装和运行的基本单位，在DevEco Studio工程目录中，一个HAP对应一个Module。应用打包时，所有的Module都只能生成.hap文件。

D.DevEco Studio会在编译构建时，不需要对HAP进行一致性校验。

78、某业务团队的架构师发现某个特性用的频率比较少，但是这个特性占用空间资源还是比较大的。为了减少首包下载体积，准备将该特性解耦出来，

并对外提供API方便主模块调用。以下说法正确的是(D) 7.15更新

A.将该特性做成hap包，通过Ability组件暴露出来给主app使用。

B.将该特性做成H5模块，通过web组件加载远程资源使用，

C.将该特性做成动态加载的har包，暴露接口给主模块使用。

D.将该特性做成按需加载的hsp包，暴露接口给主模块使用。

79、以下Websocket连接开发的步骤顺序，描述错误的是？（AB）顺序错误

A.创建一个WebSocket连接，返回一个WebSocket对象

B.导入需要的webSocket模块

C.根据URL地址，发起WebSocket连接

D.(可选)订阅WebSocket的打开、消息接收、关闭、Error事件

E.使用完WebSocket连接之后，主动断开连接

F.调用Session.start方法开启metadata数据输出

80、项目中涉及多个类的继承与重写。为了快速实现子类对父类方法的重写，小华想利用DevEco studio提供的便捷功能来提高开发效率。他了解到，通过一个特定的操作流程，

可以直接依据父类的模板生成子类中需要重写的方法代码，而无需手动编写完整方法体，在DevEcostudio中，

如何正确使用Override Methods功能来快速生成子类需要重写的方法代码？（A）

A.将光标定位到子类的定义处，按下Ctrl+0(或右键单击选择Generate..>Override Methods)，在弹出的对话框中选择要重写的方法，点击OK完成生成。

B.在项目结构视图中找到目标子类，双击打开后直接在代码编辑区输入重写方法的签名，DevEco Studio将自动完成剩余代码。

C.将光标放置于任何代码行，按下Ctr1+B，然后在弹出菜单中选择Override Methods，之后勾选需要重写的方法并确认。

D.通过菜单栏File>Settings，配置Override Methods快捷方式，之后在代码中仅需选中父类方法名，即可自动在子类中生成重写代码。

81、项目需要同时进行应用和元服务的开发，并针对当前项目工程中的代码可以分别构建出应用和元服务的包，

如何在DevEco studio中设置不同的构建配置，达成这个目的（A）

A.在工程级别build-profile.json5定义两个product，将两个product的bundleType分别设置成app和atomicService

B.在模块级别build-profile.json5定义两个target，将两个target的bundleType分别设置成app和atomicService

C.修改工程级别的AppScope/app.json5中的bundleType值为atomicService

D.修改工程级别的AppScope/app.json5中的bundleType值为app

82、DevEco Studio提供HarmonyOS应用/服务的UI预览界面与源代码文件间的双向预览功能，支持ets文件与预览器界面的双向预览。关于双向预览，下列选项说法错误的是?

关于双向预览，下列选项说法错误的是?(B)

A.选中组件树中的组件，则对应的代码块和UI界面也会高亮显示

B.双向预览不支持通过组件的属性面板实时修改属性或样式

C.选中布局文件中的代码块，则在U界面会高亮显示，组件树上的组件节点也会呈现被选中的状态。

D.选中预览器U界面中的组件，则组件树上对应的组件将被选中，同时代码编辑器中的布局文件中对应的代码块高亮显示。

83、关于代码门禁理解正确的是:（B 同事工作）

A.门禁级检查的范国和版本级检查的范围保持一致、尽可能多的在MR门禁阶段就拦截防护住问题，保障问题可以及时清理掉。

B.代码门禁则是在代码含并之前就验证代码来保护生干分支的究整性。通过这种方式，可以保护主分支代码避免因合码导致的构建中断，以确保 master分支代码始终是可部署的，并且不会因明显的错误而影响到你正在并行开发的同事工作。

C.代码门禁是一项代码质量保随措施。目的是要求开发人员提交的代码必须满足一些要求才能合入代码仓库。门禁必须强制要求包括编译通过、单元测试覆盖率达标、代码静志检查无告警、全量功能测试用例、DFX专项测试都通过。

D.标准 CI构建是在代码会并后检壹已提交代码的功能完整性，这种方法会导致代码合并到master后编译失败导致没有可用版本部署。通过提高滚动构建的频度就可以代替代码门禁，保代码主千及时发现并解决问题。

84、一个应用项目工程中，模块依赖关系如所示：

1、A(entry)模块依赖于B(har)、C(har)、D(shared)

2、D(shared)模块依赖于C(har)

那么在最终编译结果.app文件中，存在的编译产物是哪种？ （第一种）

第一种：a.hap + d.hsp

第二种：A.hap+ D.hsp +C.har

第三种：A.hap +B.har +C.har + D.hsp

第四种：A.hap + B.har + D.hsp

85、张工在使用DevEco studio开发Harmony0s应用时，遇到了代码编译警告和错误。为了提高开发效率，

哪一项正确描述了张工如何利用DevEco studio的Quick Fix功能来有效管理和修复代码中的问题?(B)

A.张工应该首先使用Ctrl+Shift+F快捷键全局搜索问题，然后手动在搜索结果中找出代码警告和错误的原因及位置。

B.张工通过双击Shift键打开搜索框，输入"problems”打开问题工具面板，双击具体告警条目可直接跳转到问题代码行。接着，将光标置于告警位置，利用弹出的悬浮窗选择合适的修复建议或点击“More actions”以查看更多修复选项。

C.张工只需在代码编辑界面按下F1键，DevEco Studio会自动识别当前光标所在行的错误并直接修复。

D.张工在代码编辑界面看到红色波浪线标记的错误时，直接右键点击错误代码，选择“Delete Line”以移除错误代码行，从而“修复”问题。

86、开发者小林正在使用DevEco studio开发一款Harmony0s应用，并在真机上进行调试。他在运行应用的过程中突然发现一处UI布局需要微调，希望在不中断当前应用运行的情况下看到调整效果，

基于DevEcostudio提供的Hot Reload(热重载)能力，以下哪一种做法能让小林最有效地实现他的需求?（C）

A.使用模拟器替代真机进行调试，因为Hot Reload仅在模拟器上支持代码改动的即时生效

B.继续运行应用，手动重启应用后检查布局是否符合预期

C.在不关闭应用的情况下，直接修改代码并保存，借助Hot Reload功能在真机上实时查看布局调整的效果

D.立即停止应用，修改代码后重新编译并部署到真机上

87、一个应用有2个UIAbility组件，其module.json中abilities标签的配置如下方代码。

在手机设备上，执行如下操作后:

1.启动UIAbility1，然后back键返回销毁UIAbility1;

2.启动UIAbility2,然后back键返回销毁UIAbility2;

3.启动UIAbility2,然后back键返回销毁UIAbility2;

进入多任务列表，能看看到该应用的几个任务视图:（3个) 7.16更新（可能1不确定）、

88、在UIAbility的onCreate生命周期中通过EventHub的on注册"event1"和"event2"事件。 (A)7.15更新[存疑]

```
import{hilog}from'@kit.PerformanceAnalysisKit';

import{UIAbility,Want,AbilityConstant} from'@kit.AbilityKit';

const DOMAIN_NUMBER:number = 0xFF00;

const TAG: string ='[EventAbility]';

export default class EntryAbility extends UIAbility{

 onCreate(want:Want,launchParam:AbilityConstant.LaunchParam):void{

  //获取UIAbility实例的上下文

  let context = this.context;

  //获取eventHub

  let eventhub=this.context.eventHub;

  //执行订阅操作

  eventhub.on('event1',this.eventFunc);

  eventhub.on('event2',this.eventFunc);

  hilog.info(DOMAIN_NUMBER,TAG,'%{public}s','AbilityonCreate');

 }

 eventFunc(argOne:Context,argTwo:Context):void{

  hilog.info(DOMAIN_NUMBER,TAG,'receive.'+`${argOne},${argTwo}`);

  return;

 }

}
```

2.在UI组件的click处理事件中调用如下的eventHubFunc，连续点击2次该控件后，运行日志输出是ABCD中的哪项?

```
import common from '@kit.AbilityKit';

import{promptAction} from '@kit.ArkUI'

@Entry

@Component

struct Page_EventHub {

 private context=getContext(this) as common.UIAbilityContext;

 eventHubFunc():void{

  this.context.eventHub.emit('event1');

  this.context.eventHub.emit('event2',2,'test2');

  this.context.eventHub.off('event1');

 }

 build() {

  Column(){

   // ...

   List({initialIndex:0}){

    ListItem(){

     Row(){

      // ...

     }

     .onClick(() => {

      this.eventHubFunc()

      promptAction.showToast({
```

```
        message:$r('app.string.EventHubFuncA')
       })
      })
     }
    }
   }
  }
 }
```

A、

```
[Example].[Entry].[EntryAbility]receive.[]
[Example].[Entry].[EntryAbility] receive.[2,"test2"]
[Example].[Entry].[EntryAbility] receive.[2,"test2"]
```

B、

```
[Example].[Entry].[EntryAbility]receive.[]
[Example].[Entry].[EntryAbility]receive.[2,"test2"]
[Example].[Entry].[EntryAbility]receive.[]
```

C、

```
[Example].[Entry].[EntryAbility]receive.[]
[Example].[Entry].[EntryAbility] receive.[2,"test2"]
```

D、

```
[Example].[Entry].[EntryAbility]receive.[]

[Example].[Entry].[EntryAbility]]receive.[2,"test2"]

[Example].[Entry].[EntryAbility]receive.[]

[Example].[Entry].[EntryAbility]]receive.[2,"test2"]
```

89、关于自动化测试描述正确的是: （C） 7.15更新

A.DT(开发者测试)就是UT，可看护的范围包括边界值问题、空指针或赋值错误内部业务逻辑问题等等。

B.自动化测试因提高效率，减少重复工作的特性而被广泛采用:自自动化测可以替代手动测试在处理复杂、难以预测的用户交互或特殊边界条件。

C.XTS子系统是认证测试套件的集合，当前包括acts(application compatibilitytest suite)应用兼容性测试套件，后续会拓展dcts(device compatibility testsuite)设备兼容性测试套件等。

D.Fuzz测试就是通过构造不规则的输入，从而触发程序的某种bug;Fuzz测试属于白盒测试。Fuzz测试也叫做模糊测试，通过输入非法字段，并观察软件是否异常来实现。一方面可以通过向软件输入非法字段，另一方面也可以通过向网络服务发送异常报文。

90、以下关于动态import说法正确的是(C)

A.动态import不支持导入SDK的API，如@ohos.*

B.动态import支持懒加载，所以不能用于提升页面的加载速度。

C.动态import支持加载HSP模块、HAR模块、OHPM包、Native库

D.动态import和静态import相比，灵活性更好，性能更好。

91、以下示例代码中可以进行动画的属性有哪些?(1、2、3都可以)

```
@Componentstruct MyComponent{
@state compwidth:number=100;
@state compHeight:number=100;
@State compRadius:number=32;
build(){
Column(){}
.width(this.compWidth)//1
.height(this.compHeight)//2
.animation({curve:Curve.Ease,duration:280})
.borderRadius(this.compRadius)//3
.onclick(()=>{this.compwidth += 10;
this.compHeight +=10;
this.compRadius += 4;
})
}
}
```

92、Harmonyos应用开发团队正着手优化一款面向全球市场的在线教育应用，该应用在特定课程直播环节出现了性能波动和响应延迟的问题，严重影响用户体验。

打算利用DevEco Profiler来进行性能优化。DevEco Profiler其设计核心和主要优势是什么？（B）

A.DevEco Profiler主要是一个自动化修复工具，能自动检测并解决所有HarmonyOS应用的性能问题

B.DevEco Profier依据Top-Down设计理念，通过高度整合的数据展示范式，提供从宏观到微观的性能数据分析，加速开发者定位和解决问题的过程

C.DevEco Profiler采用Bottom-Up设计原则，从底层代码细节开始逐步构建性能模型

D. DevEco Profiler专注于用户界面设计的美化，使开发者操作更为直观

93、在组件中，经常需要使用字符串、图片等资源。HSP中的组件需要使用资源时，一般将其所有资源放在HSP包内，

而非放在HSP的使用方处，以符合高内聚低耦合的原则。下面访问HSP资源错误的是(B)

A.通过$r访问HSP中的资源。Image($r('app.media.example')) .id('example') .borderRadius('48px')

B.使用相对路径的方式，访问HSP中的资源。Image("../../resources/base/media/example.png") .id('example') .borderRadius('48px')

C.挎包访问HSP内资源时，推荐实现一个资源管理类，以封装对外导出的资源。

将需要对外提供的资源封装为一个资源管理类：

//library/src/main/ets/ResManager.ets

export class ResManager{

static getPic():Resource{

return $r('app.media.pic');

}

static getDesc():Resource{

return $r('app.string.shared_desc');

}

}

对外暴露的接口，需要在入口文件index.ets中声明：

// library/index.ets

export{ResManager}from'./src/main/ets/ResManager';

94、在方舟字节码的函数调用规范中，前三个参数表示的含义分别是：（D）

A.new.Target、this、函数对象本身

B.new.Target、函数对象本身、this

C.this、函数对象本身、new.Target

D.函数对象本身、new.Target、this

95、开发者小张正在使用DevEco Studio开发一款Harmony0S应用，他遇到了一个仅在应用实际运行环境中出现的问题，需要调试已部署在设备上的应用以定位问题根源，

为了能够在应用已经运行的情况下介入调试，小张应该采用哪种调试方法？（C）

A. 使用"Profile"功能，因为这同样能提供对运行时应用的监控与调试能力。

B. Run without Debugging，先让应用自由运行，随后手动附加调试器

C. Attach Debugger to Process，这允许他连提到正在运行的应用进程进行调试。

D. 使用Debug功能，将应用重新推包运行调试。

96、在开发HarmonyOS应用工程时，随看业务的发展，现在需要创建一个模块，关于在DevEco Studio中创建Module，

下列选项哪种方式是错误的？（A）

A. 在hvigor目录下，单击鼠标右键，选择New>Module，创建新的Module;此时module将创建在该文件目录下。

B. 在工程根目录下创建一个新的Directory，可在该目录下单击鼠标右键，选择New>Module-，创建斯的Module，收时module将销建在该文件且录下。

C. 选中工程目录中任意文件，然后在菜单栏选择File > New > Module，开始创建新的Module，此时该module将创建在工程根目录下。

D. 鼠标移到工程目录顶部，单击鼠标右键，选择New>Module-，开始创建新的Module，此时设module将创建在工程根目录

97、在一个包含多个模块（如entry、feature、har、hsp等）的大型Harmonyos应用项目中，如果要对某个静态共享模块构建出静态构建包产物，

如何通过DevEcoStudio进行构建?

A.选中har模块，点击build菜单栏make module 'har'

98、关于延时任务开发使用的接口是

使用startWork申请任务，使用stopWork取消任务，使用getWorkStatus获取任务状态

99、代理提醒

使用publishReminder发布一个提醒类通知，使用cancelReminder取消一个指定的提醒类通知

100、关于静态检查描述错误的是:

A.静态检查 可以检测代码中的语法错误和潜在的逻辑错误，也支持检测代码在运行时现的错误，因此静态检查可以替代动态测试。

- 多选题（20道 3分一道 比较重要）

1.哪些是持续集成最佳实践？ 717更新

除了Pipeline、专职Ops外所有。

2.Navigation组件说法正确的是（Navigation和Router）：

三种显示模式、来自不同module、支持column/row

3.下面关于Module的说法正确的是（4个选项开头为Static Library、Shared Library、entry、feature)

关于feature、Static Library、Shared Library3个的描述是正确的

4.关于TaskPool和Worker的描述正确的是（多线程）：

TaskPool支持任务优先级

Worker时长上线无限制

5.用户购买商品后，你需要及时发放相关权益。但实际应用场景中，若出现异常将导致应用无法知道用户实际是否支付成功，从而无法及时发放权益，即出现掉单情况。

为了确保权益发放，你需要在以下哪些场景检查用户是否存在已购未发货的商品:(ABC)

A、createPurchase请求返回1001860051-由于已经拥有该商品，购买失败时

B、createPurchase请求返回1001860001-内部错误时

C、应用启动时

D、finishPurchase请求返回1001860052-由于未拥有该商品，发货失败时

6.以下哪几个代码段可以实现ArkWeb同层渲染能力(CD) 7.15 更新("object",test") + embed id 这两个选项

代码段A:

```
Web(...).enableNativeEmbedMode(true)
<object id="view" type="native/contents" width="100%" height="100%"style="background-co
```

代码段B:

```
Web(...).enableNativeEmbedMode(true).registerNativeEmbedRule("native",test")
<object id="view" type="native/contents" width="100%" height="100%"style="background-co
```

代码段C:

```
Web(...).enableNativeEmbedMode(true).registerNativeEmbedRule("object",test")
<object id="view" type="test/contents" width="100%" height="100%"style="background-colo
```

代码段D:

```
Web(...).enableNativeEmbedMode(true)
<embed id="view" type="native/contents" width="100%" height="100%"style="background-col
```

7.下面关于混淆的描述正确的是哪几项（AD） 7.19更新

A、修改应用混淆配置，新配置需要重新全量编译应用才能生效

B、可以在HAR模块工程的build-profile.json5中的obfuscation.consumerFiles字段中配置导出的混淆配置，该配置仅在编译依赖该HAR的模块时生效。

C、在工程build-profile.json5中的obfuscation.ruleOptions.files字段中配置该工程的混淆配置，该配置仅在编译该工程时生效。

D、支持顶层作用域名称、属性名称、文件名称等多种混淆功能

8.以下代码片段哪几处违反了ArkTS语法规范（ACD）

```
function foo(value1: number,value2: number){
return value1 + value2;
}
```

A foo();

B foo(1,2);

C foo(1,2,3);

D foo(1,2,3,4);

9.在ArkTS中，以下哪些声明类的方式是正确的?(c1 c2)

```
class c1{value:number =0;}

class C2{value: number;

constructor(value:number){this.value = value;}}

class C3{constructor(value:number){this.value = value;}}

class c4{

value: number;}
```

10.下面代码块符合Node-API开发规范的是（AD）（Demo2 和 Demo3） 7.16更新

代码块A:

```
static napi_value Demo1(napi_env env, napi_callback_info info)

{size_t argc;napi_value argv[10]={nullptr};napi_get_cb_info(env,info,&argc, argv, nullp
```

代码块B:

```
static napi_value Demo4(napi_env env, napi_callback_info info)

{size_t argc=5;napi_value argv[3]={nullptr};napi_get_cb_info(env,info,&argc, argv, null
```

代码块C:

```
static napi_value Demo2(napi_env env, napi_callback_info info)

{size_t argc =0;

napi_get_cb_info(env,info,&argc, nullptr, nullptr, nullptr);

if(argc ==0){return nullptr;}

napi_value* argv = new napi value[argc];

napi_get_cb_info(env,info,&argc, argv, nullptr, nullptr);
// 业务代码

return nullptr;}
```

代码块D:

```
static napi_value Demo3(napi_env env, napi_callback_info info)

{size_t argc =2;napi_value argv[2]={nullptr};

napi_get_cb_info(env,info, &argc, nullptr, nullptr, nullptr);
// 业务代码

return nullptr;}
```

11.以下数据类型中，哪些是Sendable数据。(ABC) 7.16更新

```
import{ lang }from '@kit.ArkTs';

class c{}

interface I extends lang.ISendable {}
```

A. interface I

B. number

C. string

D. class C

12.项目组开发的Harmonyos应用工程，为了保护hap代码资产，如何在DevEcostudio中启用混淆的功能，并设置相应的混淆规则（CD）

A、在混淆规则配置文件consumer-rules.txt中进行混淆规则的配置

B、release模式下将buildOption/arkOptions/ruleOptions/enable设置为false

C、在混淆规则配置文件obfuscation-rules.txt中进行混淆规则的配置

D、release模式下将buildOption/arkOptions/ruleOptions/enable设置为true

A中consumer-rules.txt文件主要用于库（library）项目，指定库在被其他项目依赖时应用的混淆规则。

但对于应用项目本身，主要使用obfuscation-rules.txt来配置混淆规则。所以A我没选。

13、使用如下的代码去启动一个ability时，下面哪种代码块中skills定义的组件能够被匹配到:(ABCD) 717更新

```
let want ={

"uri":"https:// www.test.com: 8080/query/books",

"type" : "text/plain"}

context.startAbility(want).then((data))=>{console.log(TAG +"startAbility success");

}).catch((err))=>{

console.log(TAG + "startAbility failed.");

}
```

代码块A：

```
"skills":[ {
"uris":[{
"scheme": "https","host":"www .test.com","pathstartwith" :"query/books","type”:"text/*”
]}]
```

代码块B：

```
"skills":[ {
"uris":[ {
"scheme": "https","type” :"text/*"}]}]
```

代码块C：

```
"skills":[ {
"uris":[ {
"scheme":"https","host":"www .test.com","pathstartwith" :"query/books","type" :"text/pl
}]}]
```

代码块D：

```
"skills":[ {
"uris":[ {
"scheme":"https","host":"www .test.com","type" :"text/plain"
}]}]
```

14、出于安全因素考虑，ArkTs不支持以下哪些语法？(CD) 7.19更新 [A存疑]

new Function('a' , 'b', 'return a + b')

Object.entries()

eval()

with()

15、ArkTs对并发编程API和能力进行了增强，以下描述正确的是？（AD） 7.16确认

A.在并发API的形式上，目前主要有两种:Worker和TaskPool

B.单次I/O任务的开发场票中，必须使用TaskPool进行开发。

C.默认情况下，Sendable数据在ArKTS并发实例间(包括主线程、TaskPool&Worker工作线程)传递的行为是拷贝传选。

D.CPU密集型任务场豪中，当在务不需要长时间(3分钟)占据后台线程，而是一个个独立的任务时，推荐使用TaskPool进行开发。

16、下面代码符合ArkTS编程规范的是? (ABCD)

const arr =[1,2,3];

if(flag){//..}else {//..}

function fight(): void {console.log("Swooosh!");}

if(isJedi){fight();}

17、以下ABCD代码片段哪几处违反了ArkTS语法规范? (BCD)

```
class Point {

public x: number;public y: number;

constructor(x:number,y:number){

this.x = x

this.y = y

}}
```

代码A：let p=new Point(1.0,1.0);

代码B：delete p.x;

代码C：p.z=2.0;

代码D：p.x='He11o!';

18、关于混淆的描述正确的是：（ABCD）全选 7.19更新

B针对工程源码的混淆可以降低工程被破解攻击的风险，缩短函数名、类名和属性名，减小应用的大小。

D代码混淆已经被集成了到SDK中，可以在DevEco Studio中方便地使用。

19、在ArkTS中，以下A、B、C、D个代码片段正确的是

BD正确

代码段B：

```
function fn(x:string): string {
return x;
}
type funcType =(ns:string|number)=> string;
let func: funcType =fn;
```

代码段D：

```
function fn(x:string |number): string {
return 'value:' + x;
}
type funcType=(ns:string|number)=>string;
let func:funcType =fn;
```

20、应用开发的有些场景中，如果希望根据条件导入模块或者按需导入模块，可以使用动态导入代替静态导入，下面导入场景中适合使用动态import的是（ABCD）

A.当被导入的模块，在加载时并不存在，需要异步获取。

B.当静态导入的模块很明显的降低了代码的加载速度且被使用的可能性很低，或者并不需要马上使用它。

C.当静态导入的模块很明显的占用了大量的系统内存且被使用的可能性很低

D.当被导入的模块说明符，需要动态构建。

21、方舟字节码文件格式，描述正确的是：

uint16_t、二进制产物

22、以下4个代码片段哪几个函数违反了ArkTS语法规范:(foo1、foo4) 7.15更新

```
function foo1(value1?: number, value2: number){
if(value1 == undefined){
return value2;}
return value1 + value2;}

function foo4( ...array: number[],value: number){
return value;
}

function foo2(value1: number, value2?: number){
if(value2 == undefined){
return value1;}
return value1 + value2;}

function foo3(value: number, ...array: number[]){return value}
```

23、ArkTS中，哪些属性声明是正确的：

短的两个1、4

24、ArkTS中的import用法，正确的是：

ABCD

import { export1 } from "ets file name"

import { export1 as alias1 } from "ets file name"

import * as name from "ets file name"

import defaultExport from "ets file name"

25、可显示图片的代码，ABCD选项 7.18更新

A.

```
@Entry
@Component
struct ImageExample {
  build() {
    Column({ space: 10 }) {
      Image($r('app.media.earth'))
      .width(100)
        .height(100)
    }
  }
}
```

B.

```
@Entry
@Component
struct ImageExample {
  build() {
    Column({ space: 10 }) {
      Image("https://www.example.com/xxx.png")
      .width(100)
        .height(100)
    }
  }
}
```

C.

```
import image from '@ohos.multimedia.image'
@Entry
@Component
struct ImageExample {
  @State imagePixelMap: image.PixelMap | undefined = undefined

  async aboutToAppear() {
    this.imagePixelMap = await this.getPixmapFromMedia($r('app.media.app_icon'))
  }

  build() {
    Column() {
      Image(this.imagePixelMap)
        .width(200)
        .height(200)
    }
```

```
  }
  private async getPixmapFromMedia(resource: Resource) {
    let unit8Array = await getContext(this)?.resourceManager?.getMediaContent({
      bundleName: resource.bundleName,
      moduleName: resource.moduleName,
      id: resource.id
    })
    let imageSource = image.createImageSource(unit8Array.buffer.slice(0, unit8Array.buf
    let createPixelMap: image.PixelMap = await imageSource.createPixelMap({
      desiredPixelFormat: image.PixelMapFormat.RGBA_8888
    })
    await imageSource.release()
    return createPixelMap
  }
}

D.
import { DrawableDescriptor } from '@ohos.arkui.drawableDescriptor'

@Entry
@Component
struct ImageExample {
  private resManager = getContext().resourceManager

  build() {
    Row() {
      Column() {
        Image((this.resManager.getDrawableDescriptor($r('app.media.drawable').id) as Dr
      }.height('50%')
    }.width('50%')
  }
}
```

26、3处手势 （AC） 7.19更新

A .gesture(

TapGesture({ count: 2 })

.onAction((event: GestureEvent) => {

if (event) {

this.value = JSON.stringify(event.fingerList[0]) }

}))

C .parallelGesture(……)

27、显示字体20号

minFontSize不对，其余对的（有个代码很复杂的，实际运行的时候import失败，默认这个是对的）

28、ABCD按钮UI不刷新

两个for循环的

29、代码重构

ABCD

30、大型应用模块化开发最佳实践（ABCD） 7.20更新 一次上架也算进去

31、某个应用的启动框架配置文件详细信息如下，以下说法正确的是（ABD） 7.16确定

关于Task003的描述是错的，其他三项对的

32、以下module.json5配置文件正确的是(ABD)

type 属性设为 har的配置是错的。

33、下面关于Module的说法正确的是（4个选项开头为Ability、Library、entry、feature）

ABCD 全对

34、hiAppEvent提供的Watcher接口，订阅到的系统事件，哪些包含HiLog日志?（BD）

A.CPU高负载事件

B.卡死事件

C.启动耗时事件

D.崩溃事件

35、以下A到G选项，哪些是持续部署最佳实践?(BEFG)

A.手工配置管理:

1、直接修改生产环境上的配置来改变系统配置

2、集群中各节点的行为有所不同;

3、靠人手工恢复环境。手动记载配置包括操作系统、应用服务器、关系型数据库管理系统、Web服务器或其他基础设施设置。

B：监控和回滚机制:实时监控部署后的应用状态，如有问题及时回滚。

C：开发完成之后再向类生产环境部署:当软件被第一次部署到类生产环境(比如试运行环境)时，就是大部分开发工作完成时，至少是开发团队认为“该软件开发完成了”。

D：手工部署:持续部署可以采用手工部署的方式发布软件:

1、有一份非常详尽的文档，该文档描述了执行步骤及每个步骤中易出错的地方;

2、以手工测试来确认该应用程序是否运行正确;

3、在发布时，常常会修正一些在发布过程中发现的问题。

E：灰度发布:先在小部分用户或区域进行部署，观察没问题后再全面推广

F：环境一致性:保持开发、测试、生产等环境的高度一致性。

G：自动化部署流程:从代码提交到部署的整个流程应尽可能自动化。

36、下面关于Node-API数据类型描述正确的是？(AC)

A napi_env:用于表示Node-API执行时的上下文

B napi_threadsafe_function_cal_mode:该枚举类型定义了两个常量，用于指定在何时释放线程安全函数的回调函数

C napi_status:是一个枚举数据类型，表示Node-API接口返回的状态信息

D napi_threadsafe_function_release_mode:该枚举类型定义了两个常量，用于指定线程安全函数的调用模式

37、List组件onscrollIndex事件触发时机是？（ABC）7.20更新

A.List组件列表滚动时每帧触发

B.List组件首次加载完成时触发

C.List组件显示区域内第一个子组件或最后一个子组件或中间位置子组件索引值变化时触发

D.List组件滚动停止时触发

38、HarmonyOS应用开发者小张，正在利用DevEco Studio进行一款复杂应用的UI界面调试。小张了解到ArkUI Inspector是DevEco Studio内置的一项强大工具，能够显著提升UI调试效率。

基于ArkUI Inspector的特性描述，下列哪些描述是正确的？(ABCD)

A.交互式组件选择:用户既可以在组件树视图中选择组件，使U界面上对应组件高亮显示并展示其属性详情:也可以直接在U1布局显示界面上点击选择组件

B.UI快照管理:支持导出应用的UI界面为快照图片，并允许这些快照被导入回ArkUl Inspector中，便于离线分析或分享讨论UI设计方案

C.UI效果查看:开发者能够查看连接真机上运行的应用程序的U显示效果，页面组件树结构，以及选中组件的属性信息

D.性能监控:提供详细的UI渲染性能指标，帮助开发者识别布局瓶颈和渲染效率问题，从而优化应用性能

39、HSP支持导出ArkU组件、接口，供应用内的其他HAP/HSP引用，下面正确的是（ 选ABC错了）

导出ts类和方法

导出ArkUI组件

导出native方法

40、为了加快构建速度，提高开发效率，可以如何调整hvigor配置，从而优化构建速度？（BCD） 7.20更新

A.启用hvigor的typeCheck，在增量场景下进行对hvigorfile.ts进行类型检查

B.启用hvigor的incremental，在增量场景下检查任务是否可以跳过

C.启动hvigor的daemon模式，在增量场景下复用缓存

D.启用hvigor的parallel，在增量场景下进行并行编译处理

41、某业务团队发现用户对他们App的某个特性使用率并不高，为了节省用户首次下载安装包的体积，考虑将该特性做成按需加载，那么推荐该特性使用的工程模块类型是?

A.hap

B.hsp

C.app

D.har

(BC)

42、以下napi代码有问题的是?

43 使用ArkUI组件复用之后，还是没有显著提升列表滑动场景的性能，属于组件复用未生效可能的原因有? (BCD) 7.20更新

A没有在aboutToReuse更新关联的状态变量数据。

B复用的自定义组件中使用if等条件渲染语句导致结构不相同，未使用reuseId标记。

C页面嵌套了过多自定义组件。

D在aboutToReuse回调函数中更新了冗余的状态变量

44 以下代码片段哪几处违反了ArkTS语法规范。(ACD)

function foo(value:number){return value;}

A foo('');
B foo(0);
C foo(undefined);
D foo(null);

45 下面关于方舟字节码文件格式描述正确的是（CD）

A.方舟字节码文件中不包含字节码文件内容的adler32校验和

B. 方舟字节码文件中数据类型uint32_t表示32-bit无符号整数，采用大端字节序

C.方舟字节码文件中数据类型uint16_t表示16-bit无符号整数，采用小端字节序

D.方舟字节码文件是ArkTS/TS/JS编译后的二进制产物

46 ArkTs中不能使用以下哪些类型。（AB）7.20更新

A. unknown

B. any

C. union type

D. tuple type

47 当前动态import支持导入的模块类型有哪些? (ABCD)

A.动态import支持加载HSP模块

B.动态import支持加载远程HAR模块

C.动态import支持加载OHPM模块

D.动态import支持加载本地HAR模块

48 ArkTs是鸿蒙生态的应用开发语言。下列说法正确的是 （ABD）

A.ArKTS提供了声明式UI范式、状态管理支持等相应的能力，让开发者可以以更简洁、更自然的方式开发应用

B.针对JavaScript(简称JS)/TS并发能力支持有限的问题，ArKTS对并发编程API和能力进行了增强。

C. TS/JS代码支持import ArkTS代码。

D. ArKTS在保持TypeScript(简称TS)基本语法风格的基础上，进一步通过规范强化静态检查和分析，使得在程序运行之前的开发期能检测更多错误，提升代码健壮性，并实现更好的运行性能。

49 下面关于混淆规则描述正确的是 （AB）

-enable-export-obfuscation:开启直接导入或导出的类或对象的名称和属性名混淆

-disable-obfuscation:关闭所有混淆

-enable-toplevel-obfuscation:开启属性混淆

-enable-property-obfuscation:开启顶层作用域名称混淆

50 那些属性声明是正确的

value1 value4的声明

51 下面ABCD关于混淆规则描述正确的是（AC）

A. -print-namecache filepath: 将名称缓存保存到指定的文件路径。

B. -keep-property-name [,identifiers,..]:指定要保留的顶层作用域的名称

C. -keep-file-name [,identifiers...]:指定要保留的文件/文件夹的名称

D. -keep-global-name [,identifiers....]:指定想保留的属性名

52 如下ABC 3处手势，有机会执行的是哪几处? (AC)

A 含有count2的代码

C parallelGesture

```
@Entry
@Component
struct ListTest {
  scroller: Scroller = new Scroller()
  scroller2: Scroller = new Scroller()
  scroller3: Scroller = new Scroller()
  private arr: number[] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
  private childRecognizer: GestureRecognizer = new GestureRecognizer()
  private currentRecognizer: GestureRecognizer = new GestureRecognizer()
  private lastOffset: number = 0

  build() {
    Stack({ alignContent: Alignment.TopStart }) {
      Scroll(this.scroller) {
        Column() {
          Text("Scroll Area")
          .width('90%')
          .height(150)
            .backgroundColor(0xFFFFFF)
            .borderRadius(15)
            .fontSize(16)
            .textAlign(TextAlign.Center)
            .margin({ top: 10 })
          List({ space: 20, initialIndex: 0 }) {
            ForEach(this.arr, (item: number) => {
              ListItem() {
                Text('' + item)
                .width('100%').height(100).fontSize(16)
                  .backgroundColor(Color.Blue)
                  .textAlign(TextAlign.Center).borderRadius(10)
              }
            }, (item: string) => item)
          }
          .listDirection(Axis.Vertical) // 排列方向
          .scrollBar(BarState.Off)
          .friction(0.6)
          .divider({ strokeWidth: 2, color: 0xFFFFFF, startMargin: 20, end
          .edgeEffect(EdgeEffect.None) // 边缘效果设置为Spring
          .height(1000)
          .width('90%')
          .id("inner")
        }.width('100%')
      }
```

```
      .id(“outer”)
      .height(600)
        .scrollable(ScrollDirection.Vertical) // 滚动方向纵向
        .scrollBar(BarState.On) // 滚动条常驻显示
        .scrollBarColor(Color.Gray) // 滚动条颜色
        .scrollBarWidth(10) // 滚动条宽度
        .edgeEffect(EdgeEffect.None)
        .onScroll((xOffset: number, yOffset: number) => {
          console.info(xOffset + ’ ’ + yOffset)
        })
        .onScrollEdge((side: Edge) => {
          console.info(‘To the edge’)
        })
        .onScrollStop(() => {
          console.info(‘Scroll Stop’)
        })
      A .gesture(
        TapGesture({ count: 2 })
          .onAction((event: GestureEvent) => {
            if (event) {
              this.value = JSON.stringify(event.fingerList[0])
            }
          })
      )
      B .gesture(
        PanGesture({PanDirection.Vertical})
          .onActionUpdate((event: GestureEvent)=>{
            console.log("zcb onActionUpdate event offsetY " + event.offset
          })
      )
      C .parallelGesture(
        PanGesture({PanDirection.Vertical})
          .onActionUpdate((event: GestureEvent)=>{
            console.log("zcb onActionUpdate event offsetY " + event.offset
          })
      )
    }.width(‘100%’).height(‘100%’).backgroundColor(0xDCDCDC)
  }
}
```

53 如果想让grid上的捏合手势手势生效，而不跟grid上的滚动手势形成冲突，.xxxx?手势接口应该怎么配置? ABC (A存疑)

A. gesture

B.priorityGesture

C.parallelGesture

D. GesureGroup

54 以下哪些是可以在Navigation中使用pushPathByName接口传递的params的参数类型 （ABD） 720更新

A string

B arrayBuffer

C map<string,string>

D record<string,string>

55 4个按钮，不会触发UI刷新 正确

A count=0

C value=100

56 在开发Harmonyos应用的多元化测试环境中，DevEco studio引入了本地模拟器(Loca1Emulator)作为重要工具，旨在帮助开发者在个人开发机器上高效模拟Harmonyos环境，进行应用或服务的快速运行与细致调试。请根据本地模拟器的实际应用场景与系统要求，选出所有正确的描述选项(BCD)

A.开发者需要注意的是，DevEco Studio的本地模拟器可以在虚拟机内部进一步运行，以节省硬件资源。

B.为了保证流畅的运行和调试体验，本地模拟器推荐macOS系统版本至少为12.5以上。

C.DevEco Studio的本地模拟器允许开发者在个人电脑上模拟HarmonyOS环境，便于应用或服务的运行与调试。

D.mac计算机配置方面，为了确保本地模拟器的稳定运行，推荐至少配备8GB RAM。

57 通过如下openLink的接口启动，如下哪些配置的UAbility不可能被拉起？（3个） 7.19更新

两项http的配置无法拉起，还有一项https代码如下：

```
{
“name”: “TargetAbility”,
“skills”: [
{
“actions”:[
“ohos.want.action.sendData”,
],
“entities”:[
“entity.system.browsable”,
],
“uris”:[
{
“scheme”: “https”,
“host”: “www.test.com”,
“port”: “8080”,
“path”: “path”,
“autoVerify”: ture
}
]
}
]
```

```
}
```

58 以下关于Localstorage的说法正确有哪些? (ACD)

A.被@Component装饰的组件最多可以访问一个LocalStorag?实例和AppStorage，未被@Entry装饰的组件不可被独立分配LocalStorage实例，只能接受父组件的LocalStorage实例。

B.LocalStorage中的所有属性都是不可变的。

C.组件树的根节点，即被@Entry装饰的@Component，可以被分配一个LocalStorage实例，此组件的所有子组件实例将自动获得对该LocalStorage实例的访问权限。

D.应用程序可以创建多个LocalStorage实例，LocalStorage实例可以在页面内共享，也可以通过GetShared接口，获取在UIAbility里创建的GetShared，实现跨页面、UAbility内共享。

59 module ABCD 正确

60 如下哪些方式可实现图片动态播放？（ACD） 7.16确定

代码A：

```
@Entry

@Component

struct ImageAnimatorExample {

  @State iterations: number = 1

  build() {

    Column({ space: 10 }) {

      ImageAnimator()

        .images([

          {

            src: $r('app.media.img1')

          },

          {

            src: $r('app.media.img2')

          },

          {

            src: $r('app.media.img3')

          },

          {

            src: $r('app.media.img4')

          }

        ])

        .duration(2000)
```

```
        .fillMode(FillMode.None).iterations(this.iterations).width(340).height(240)
        .margin({ top: 100 })
    }.width('100%').height('100%')
  }
}
```

代码B：

```
import {AnimationOptions, AnimatedDrawableDescriptor} from '@ohos.arkui.drawableDescrip
import image from '@ohos.multimedia.image'


@Entry
@Component
struct ImageExample {
  pixelmaps: Array<PixelMap> = [];
  options: AnimationOptions = {duration:2000, iterations:1};
  @State animated: AnimatedDrawableDescriptor | undefined = undefined;


  async aboutToAppear() {
    this.pixelmaps = await this.getPixelMaps();
    this.animated = new AnimatedDrawableDescriptor(this.pixelmaps, this.options);
  }


  build() {
    Column() {
      Row() {
        Image(this.animated)
          .width('500px').height('280px')
      }.height('50%')
      Row() {
```

```
      Button('once').width(100).padding(5).onClick(() => {
        this.options = {duration:2000, iterations:1};
        this.animated = new AnimatedDrawableDescriptor(this.pixelmaps, this.options);
      }).margin(5)
    }
  }.width('50%')
}

private async getPixmapFromMedia(resource: Resource) {
  let unit8Array = await getContext(this)?.resourceManager?.getMediaContent({
    bundleName: resource.bundleName,
    moduleName: resource.moduleName,
    id: resource.id
  })
  let imageSource = image.createImageSource(unit8Array.buffer.slice(0, unit8Array.buf
  let createPixelMap: image.PixelMap = await imageSource.createPixelMap({
    desiredPixelFormat: image.PixelMapFormat.RGBA_8888
  })
  await imageSource.release()
  return createPixelMap
}
```

```
  private async getPixelMaps() {
    Mypixelmaps.push(await this.getPixmapFromMedia($r('app.media.icon'))) //对应资源图片名
    return Mypixelmaps;
  }
}
```

代码C：

```
import {AnimationOptions, AnimatedDrawableDescriptor} from '@ohos.arkui.drawableDescrip
import image from '@ohos.multimedia.image'


@Entry
@Component
struct ImageExample {
  pixelmaps: Array<PixelMap> = [];
  options: AnimationOptions = {duration:2000, iterations:1};
  @State animated: AnimatedDrawableDescriptor | undefined = undefined;


  async aboutToAppear() {
    this.pixelmaps = await this.getPixelMaps();
    this.animated = new AnimatedDrawableDescriptor(this.pixelmaps, this.options);
  }


  build() {
    Column() {
      Row() {
        Image(this.animated)
          .width('500px').height('280px')
      }.height('50%')
      Row() {
```

```
      Button('once').width(100).padding(5).onClick(() => {
        this.options = {duration:2000, iterations:1};
        this.animated = new AnimatedDrawableDescriptor(this.pixelmaps, this.options);
      }).margin(5)
    }
  }.width('50%')
}

private async getPixmapListFromMedia(resource: Resource) {
  let unit8Array = await getContext(this)?.resourceManager?.getMediaContent({
    bundleName: resource.bundleName,
    moduleName: resource.moduleName,
    id: resource.id
  })
  let imageSource = image.createImageSource(unit8Array.buffer.slice(0, unit8Array.buf
  let createPixelMap: Array<image.PixelMap> = await imageSource.createPixelMapList({
    desiredPixelFormat: image.PixelMapFormat.RGBA_8888
  })
  await imageSource.release()
  return createPixelMap
}
```

```
  private async getPixelMaps() {
    let Mypixelmaps:Array<PixelMap> = await this.getPixmapListFromMedia($r('app.media.i
    return Mypixelmaps;
  }
}
```

代码D:

```
@Entry
@Component
struct ImageExample {
  build() {
    Column({ space: 10 }) {
      Image($r('app.media.earth')) //对应资源图片名后缀为gif
        .width(100)
        .height(100)
    }
  }
}
```

61 如何实现类似下图布局（BC） A选项采用Grid方式不符

B.

```
@Entry
@Component
struct Demo {
  // 忽略其他辅助代码
  dataSource: ItemDataSource = new ItemDataSource(100)
  itemHeightArray: number[] = []
  colors: number[] = [0xFFC0CB, 0xDA70D6, 0x6B8E23, 0x6A5ACD, 0x00FFFF, 0x00FF7F]
  scroller: Scroller = new Scroller()
  @State sections: WaterFlowSections = new WaterFlowSections()
  sectionMargin: Margin = {
    top: 10,
    left: 5,
    bottom: 10,
    right: 5
  }
  oneColumnSection: SectionOptions = {
    itemsCount: 3,
    crossCount: 1,
    rowsGap: 10,
    margin: this.sectionMargin,
    onGetItemMainSizeByIndex: (index: number) => {
      return this.itemHeightArray[index % 100]
    }
  }
  lastSection: SectionOptions = {
    itemsCount: 97,
    crossCount: 2,
    margin: this.sectionMargin,
    onGetItemMainSizeByIndex: (index: number) => {
      return this.itemHeightArray[index % 100]
    }
  }

  aboutToAppear() {
    this.setItemSizeArray()
    // 初始化瀑布流分组信息
    let sectionOptions: SectionOptions[] = []
    sectionOptions.push(this.oneColumnSection)
    sectionOptions.push(this.lastSection)
    this.sections.splice(0, 0, sectionOptions)
  }

  build() {
```

```
    WaterFlow({ scroller: this.scroller, sections: this.sections }) {
      LazyForEach(this.dataSource, (item: number) => {
        FlowItem() {
          ReusableFlowItem({ item: item })
        }
        .width('100 %')
        .backgroundColor(this.colors[item % 5])
      }, (item: string) => item)
    }
    .columnsGap(10)
    .rowsGap(5)
    .backgroundColor(0xFAEEE0)
    .width('100 %')
    .height('100 %')
  }
}
```

C.

```
@Entry
@Component
struct Demo {
  // 忽略其他辅助代码
  dataSource: ItemDataSource = new ItemDataSource(100)
  itemHeightArray: number[] = []
  colors: number[] = [0xFFC0CB, 0xDA70D6, 0x6B8E23, 0x6A5ACD, 0x00FFFF, 0x00FF7F]
  scroller: Scroller = new Scroller()

  aboutToAppear() {
    this.getItemSizeArray()
  }

  build() {
    Column() {
      List({ scroller: this.scroller, space: 10 }) {
        ListItem() {
          Grid() {
            GridItem() {
              Text(‘
              GoodsTypeList’
              )
            }.backgroundColor(this.colors[0])

            GridItem() {
              Text('AppletService')
            }.backgroundColor(this.colors[1])

            GridItem() {
              Text('ReloadData')
            }.backgroundColor(this.colors[2])
          }
          .rowsGap(10)
          .columnsTemplate('1fr')
          .rowsTemplate('1fr 1fr 1fr')
          .width('100%')
          .height(100)
        }

        ListItem() {
          WaterFlow() {
            LazyForEach(this.datasource, (item: number, index: number) => {
              FlowItem() {
```

```
                // 使用可复用自定义组件
                ReusableItem({ item: item + index })
              }
              .width('100%')
              .height(this.itemHeightArray[item % 100])
              .backgroundColor(this.colors[item % 5])
            }, (item: number) => '' + item + this.itemHeightArray[item % 100])
          }
          .id('waterflow')
          .columnsTemplate("1fr 1fr")
          .columnsGap(10)
          .rowsGap(5)
          .width('100%')
          .height('100%')
          .nestedScroll({
            scrollForward: NestedScrollMode.PARENT_FIRST,
            scrollBackward: NestedScrollMode.SELF_FIRST
          })
        }
      }
      .scrollBar(BarState.Off)
      .edgeEffect(EdgeEffect.None)
    }
    .width('100%')
    .padding({ left: 10, right: 10 })
  }
}
```

62、以下关于ArkTS线程实例间传输实现方式描述正确的是（BC） 7.15更新 去掉了D

以下代码块中关于ArkTS线程实例间传输实现方式的调用语句，描述正确的是ABCD中哪几项?

```
import { taskpool,worker } from '@kit.ArkTs';

@sendable

class A{}

let a: A= new A();

@Concurrent

function foo(a: A){}

let task:taskpool.Task = new taskpool.Task(foo, a)

let w= new worker.Threadworker("entry/ets/workers/Worker.ets" )

taskpool.execute(task).then(()=>{});

w.postMessageWithSharedSendable(a);

task.setCloneList([a]);

taskpool.execute(task).then(()=>{});

w.postMessage(a);
```

A： w.postMessage(a);，Worker 共享传输实现方式

B： taskpool.execute(task).then(()=>{});，TaskPool 共享传输实现方式

C： w.postMessageWithSharedSendable(a);，Worker 共享传输实现方式

D： task,setCloneList([a]); taskpool.execute(task).then(()=>{};，TaskPool 共享传输实现方式

63、下面关于动态import描述正确的（ACD）

A.HAR模块间只有变量动态import时可以进行模块解耦

B.常量动态import也必须配置runtimeOnly选项

C.动态import根据入参是否为常量字符串分为常量动态import和变量动态import两种

D.动态导入import()是个异步接口，调用后将返回一个promise

64、开发者开发了一个应用，该应用只有一个hap包，其module.json5中abilities的配置如下所示，包含1个UIAbility(无Web组件)、1个FormExtensionAbility组件、

1个WorkSchedulerExtensionAbility组件，那么该应用在运行过程中，最多会存在几个应用进程：（3个） 7.19更新

```
"abilities":
{
  "name" : "EntryAbility",
  "srcEntry" : "./etc/entryability/EntryAbility.ts",
  "description" : "$string:EntryAbility_desc",
  "exported" : ture
  }
,
"extensionAbilities":
{
  "name": "ClockFormExtensionAbility",
  "srcEntrance": "./ets/form/ClockFormExtensionAbility.ts",
  "type": "form"
  },
{
  "name": "TipWorkSchedulerExtensionAbility",
  "srcEntrance": "./ets/service/TipWorkSchedulerExtensionAbility.ts",
  "type": "workScheduler"
  }
```

65、下面代码段符合ArkTS编程规范的是（D）A存疑没定义condition

A：

```
if (condition) {
```

```
console.log('success');

}
```

B：

```
for (let idx= 0 ; idx <5 ; ++idx)

console.log(idx);
```

C：

```
let maxCount = 10,isCompleted=false;

let pointX,pointY;

pointX = 10;pointY = 0
```

D：

```
let maxCount = 10;

let isCompleted= false;

let pointX = 0;

let pointY = 0;
```

66、在基于Stage模型开发的应用项目代码下，都存在一个app.json5配置文件，用于配置应用的全局信息，

以下ABCD4个appjson5配置文件错误的是 （CD） 7.17更新

A、

```
{
  "app": {
  "bundleName":"com.example.myapplication",
  "vendor":"example",
  "versionCode":1000000,
  "versionName":"1.0.2",
  "icon":"$media:app_icon",
  "label":"$string:app_name"
}
}
```

B、

```
{
  "app": {
  "bundleName":"com.example.myapplication",
  "vendor":"example",
  "versionCode":1000000,
  "versionName":"1.0.2",
  "icon":"$media:app_icon",
  "label":"$string:app_name",
  "bundleType":"app"
```

```
}
}
```

C、

```
{
  "app": {
  "bundleName":"com.example.myapplication",
  "vendor":"example",
  "versionCode":1000000,
  "versionName":"1.0.2",
  "icon":"$media:app_icon",
  "label":"app_name",
  "bundleType":"app"
}
}
```

D、

```
{
  "app": {
  "bundleName":"com.example.myapplication",
  "vendor":"example",
```

```
    "versionCode":1000000,
    "icon":"$media:app_icon",
    "label":"$string:app_name",
    "bundleType":"app"
  }
  }
```

67、在ArkTS中，以下哪些属性的声明是正确的。（value1、3、4）

```
  class C{
   value1:number = 0;
   value2?:number = null;
   value3:number| undefined = undefined;
   value4?:number;
  }
```

68、下面关于方舟字节码指令含义说明正确的是？（BC）7.19更新

A假设寄存器v0存放了对象A, 累加器(acc)存放了对象B,那么执行指令"lda v0"后，v0存放对象B，acc存放对象B

B.假设寄存 器v0存放了对象A,累加器(acc)存放了对象B,那么执行指令"lda v0"后，v0存放对象A，acc存放对象A

C. 假设寄存 器v0存放了对象A,寄存器v1存放了对象B,那么执行指令'mov v0, v1"后，v0存放对象B, v1存放对象B

D.假设寄存器v0存放了对象A, 寄存器v1存放了对象B,那么执行指令"movv0,v1"后，v0存放对象A v1存放对象A

69、Code Linter针对ArKT5/TS代码进行最佳实践/编程规范方面的检查，最佳实践/编程规范方面的检查规则可以配置，针对codelinter的配置项一下哪些说法是正确的？(ACD)

A.ruleSet:配置检查使用的规则集，规则集支持一次导入多条规则。

B.rules:可以基于ruleSet配置的规则集，新增额外规则项，但是无法修改ruleSet中规则默认配置

C.ignore:配置无需检查的文件目录，其指定的目录或文件需使用相对路径格式，相对于code-inter.json5所在工程根目录，例如:build/**/*

D.fles:配置待检查的文件名单，如未指定目录，规则适用于所有文件,例如:["**/*.ets","**/*.js","**/.ts"]

70、在使用DevEco Studio进行HarmonyOS应用开发和调试过程中，开发者小张遇到应用运行时意外终止的情况，他需要快速定位并解决导致应用崩溃的问题。以下哪些做法可以帮助小张有效分析和处理这些问题?

A. 利用系统自动生成的FaultLog，包括App Freeze、CPP Crash、JS Crash、System Freeze和ASan报告，这些报告会详细记录故障发生时的环境、堆栈信息和可能的故障原因，是排查问题的重要参考

B. 当怀疑问题是由于C++代码中的内存错误（如数组越界、内存泄露、重复释放内存）引起时，进入“Run/Debug Configurations”设置界面，勾选启用Address Sanitizer（ASan），然后重新部署应用进行测试以获取更详细的内存问题报告

C. 查看DevEco Studio log工具栏输出的错误日志，根据日志提示分析应用崩溃的具体原因及代码位置

D. 若遇到App运行卡顿或系统整体无响应（App Freeze、System Freeze）的情况，可以通过性能分析工具中的Frame Insights和Allocation Insights功能，分析应用的资源消耗情况，寻找可能的瓶颈

71、以下代码片段哪几个class/interface违反了ArkTS语法规范（AD）

```
class Person {}

class Student extends Person {}

class Instructor implements Person {}

interface Shape {}

interface Circle implements Shape {}

class Square implements Shape {}
```

A.Circle

D.Instructor

72、ArkTS是鸿蒙生态的应用开发语言。以下哪些选项是ArkTS的设计理念。

A.ArkTS保留了TS大部分的语法特性，帮助开发者更容易上手ArkTS。

C.通过规范强化静态检查和分析，减少运行时的类型检查，从而降低了运行时负载，提升执行性能。

D.通过规范强化静态检查和分析，使得许多错误在编译时可以被检测出来，降低代码运行错误的风险。

73、开发者小李正在使用DevEco Studio开发一款面向HarmonyOS的应用，该应用需要在多种设备上表现出一致的稳定性和优秀的用户体验。为了确保高质量的发布，小李意识到需要实施一套全面的测试策略，覆盖代码的自动化测试和手动测试，还需要衡量代码的测试覆盖率，以确定测试的充分性。在DevEco Studio的测试框架下，

以下描述中，哪些是正确的？（ACD）

A.无论选择Instrument Test还是Local Test，DevEco Studio均内置了详尽的测试报告功能，实时显示测试进度，且直接在IDE中可查看代码覆盖率报告，无需外部工具。

C.DevEco Studio的测试框架提供测试用例执行能力，包含基础接口以编写和输出测试结果，鼓励用户创建易于维护的自动化测试脚本，并且统计代码覆盖率。

D.Instrument Test：测试用例存储于项目的ohosTest目录，要求在HarmonyOS设备或模拟器上执行，兼容ArkTS与JS语言编写。

74、在开发过程中，我们可以将每个功能模块作为一个独立的Module进行开发。关于Module，下列选项说法正确的是？（ABC）

A.Shared类型的Module：动态共享库。HSP中的代码和资源可以独立编译，运行时在一个进程中代码也只会存在一份。

B.HAR类型的Module：静态共享库。HAR中的代码和资源跟随使用方编译，如果有多个使用方，它们的编译产物中会存在多份相同拷贝。

C.feature类型的Module：应用的动态特性模块，编译后生成feature类型的HAP。一个应用中可以包含一个或多个feature类型的HAP，也可以不包含。