



信息工程系

教

案

课程名称： 鸿蒙应用开发

教 师： 陈凯斌

总学时： 36

理论学时： 0

实训学时： 36

上课班级： 电子信息工程技术 24 级

授课学期： 2025-2026 第 1 学期

第 1 章 课程介绍和鸿蒙开发环境搭建(2 课时)

教学目标

1、知识目标：

理解掌握 HarmonyOS 的基本架构和技术特点与一次开发多端部署理念；熟悉 DevEco Studio 的四大功能，包括工程管理、代码编辑、编译构建、模拟调试。

鸿蒙是一款面向万物互联的全新操作系统，其特点如下。

1. 分布式架构

鸿蒙采用分布式架构设计，可以在各种设备之间实现高效的连接和通信，如智能手机、平板电脑、智能穿戴设备、智能家居等各种物联网设备。

2. 多终端适配

鸿蒙支持多种终端，能够在不同类型的终端上运行，为开发者提供了更加灵活的选择。

3. 统一开发平台

鸿蒙提供了统一的开发平台，开发者可以通过一套代码实现多个平台的应用开发，减少了开发成本和工作量。

4. 流畅的用户体验

鸿蒙提供了流畅的用户体验，包括快速响应、高效运行、稳定可靠等用户体验。

5. 安全和隐私保护

鸿蒙注重安全和隐私保护，采用了多种安全机制来保护用户数据和隐私，确保用户信息不被泄露或滥用。

6. 开放生态

鸿蒙秉持开放生态的理念，为开发者提供丰富的开发工具和资源，支持多种开发语言和框架，鼓励开发者共同参与其生态建设。

2、能力目标：

独立完成 Windows 10/11 系统下 DevEco Studio 的下载、安装、首次启动向导、并能创建 Hello World 工程，最终在本地 Previewer 或远程真机运行成功。

下载和安装DevEco Studio

鸿蒙为开发者提供了DevEco Studio，使用它可以开发鸿蒙应用。DevEco Studio支持运行在Windows和macOS中，本书主要基于Windows操作系统进行讲解。

在使用Windows操作系统时，DevEco Studio的基本配置要求如下。

- 操作系统：Windows 10 64位及以上版本。
- 内存：至少8G内存，推荐16GB及以上。
- 网络：能够访问互联网。



3、素质目标：

培养“先查官方、再问同学、最后找老师”的自主学习习惯；
形成“环境配置即工程起点”的质量意识；
树立“动手前先记录、出错后立即复盘”的工程日志意识。

4、课程思政目标：

引导学生主动使用华为开源镜像站并理解“自主可控、信息安全”背后的国家战略；强调遵守开源协议、尊重知识产权；通过小组互助安装，体验“集中力量办大事”的集体主义精神。

重难点内容

重点：DevEco Studio 的完整安装流程、必选组件的选择逻辑、Hello World 项目

运行成功的标志。

难点：路径中包含空格或中文导致的构建失败；环境变量未生效时的手动刷新方法。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：教师展示智能家居、车机、手表等同屏协同视频，提问“一套代码如何运行在多设备？”引出 HarmonyOS 理念，激发兴趣。
2. 讲授：结合课件讲授 HarmonyOS 架构，呈现 DevEco Studio 的功能布局。



3. 演示：演示从官网下载、双击安装、首次启动向导、关键选项勾选到 Hello World 运行的一次性成功流程。
4. 实践：学生独立完成同样流程。
5. 协作排障：结合实践遇到的问题，根据官方文档和搜索到的解决方案，尝试互助解决问题。
6. 总结：教师回顾关键节点，布置课后任务：每人提交一份“安装踩坑笔记”，下次课互评。

下载和安装DevEco Studio

通过华为开发者联盟可以下载DevEco Studio，从华为开发者联盟进入DevEco Studio的下载页面，如下图所示。



17

下载和安装DevEco Studio

- ① 双击DevEco Studio安装包启动安装程序，进入“欢迎使用DevEco Studio安装程序”界面。
- ② 单击“下一步”按钮，会转到“选择安装位置”界面。



18

下载和安装DevEco Studio

- ③ 单击“浏览”按钮可以重新设置安装位置，单击“下一步”按钮进入“安装选项”界面。
- ④ 单击“下一步”按钮进入“选择开始菜单目录”界面。



19

下载和安装DevEco Studio

- ⑤ 单击“安装”按钮进入“安装中”界面。

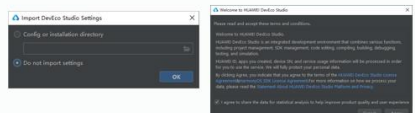
安装成功后，进入“DevEco Studio安装程序结束”界面。



20

下载和安装DevEco Studio

- ⑥ 选中“运行 DevEco Studio”复选框，单击“完成”按钮，会弹出“Import DevEco Studio Settings”对话框。
- ⑦ 单击“OK”按钮，会弹出“Welcome to HUAWEI DevEco Studio”对话框。



21

下载和安装DevEco Studio

- ⑧ 单击“Agree”按钮，会弹出“Welcome to DevEco Studio”窗口。



- ⑨ 单击“Create Project”按钮可以创建一个项目，单击“Open...”按钮可以打开文件夹，单击“Get from VCS”按钮可以通过版本控制工具获取远程项目。

22

第 2 章 DevEco 项目创建与目录结构（2 学时）

教学目标

1、知识目标：

理解 DevEco 项目的创建流程；

掌握 project name、bundle Name、save location 等配置字段含义；

使用 DevEco Studio 创建项目

下面对填写信息界面的各项进行介绍。

- **Project name**：用于填写项目名称，由大小写字母、数字和下划线组成。
- **Bundle name**：用于设置包名，包名通常使用反写域名的形式。
- **Save location**：用于设置项目的保存路径，由大小写字母、数字和下划线组成，不能包含中文字符。
- **Compatible SDK**：用于设置最低兼容的 API 版本。
- **Module name**：用于设置默认模块的名称，默认是 entry。
- **Device type**：用于设置支持的设备类型。

熟悉项目工程目录结构和关键文件

在创建鸿蒙项目时，会自动生成一些文件和目录，鸿蒙项目的一级目录如下所示。

目录	作用
.hvm	用于存放构建配置文件
.idea	用于存放开发工具配置文件
AppScope	用于存放应用全局需要的资源文件
entry	用于存放应用模块文件，包括入口文件、代码和资源等
hvm	用于存放自动化构建工具，包括任务注册编排、工程模型管理等
oh_modules	用于存放项目所依赖的第三方库文件

2、能力目标：

能用 Empty Ability 模板创建一个名为 MyFirstApp 的项目，能在 Preview 中验证页面显示“Hello World”，并能根据教师指令定位并打开指定配置文件。

3、素质目标：

养成“先读目录结构、再写业务代码”的工程规范意识；
培养给文件和文件夹起“见名知意”英文名称的习惯。

4、课程思政目标:

在讲解开源库引用时，强调 LICENSE 文件必须一并保留，树立合法合规意识；通过分组“找文件”竞赛，体验团队协作与集体荣誉感。

重难点内容

重点: DevEco 项目的创建流程，特别是以 Empty Ability 为模板。

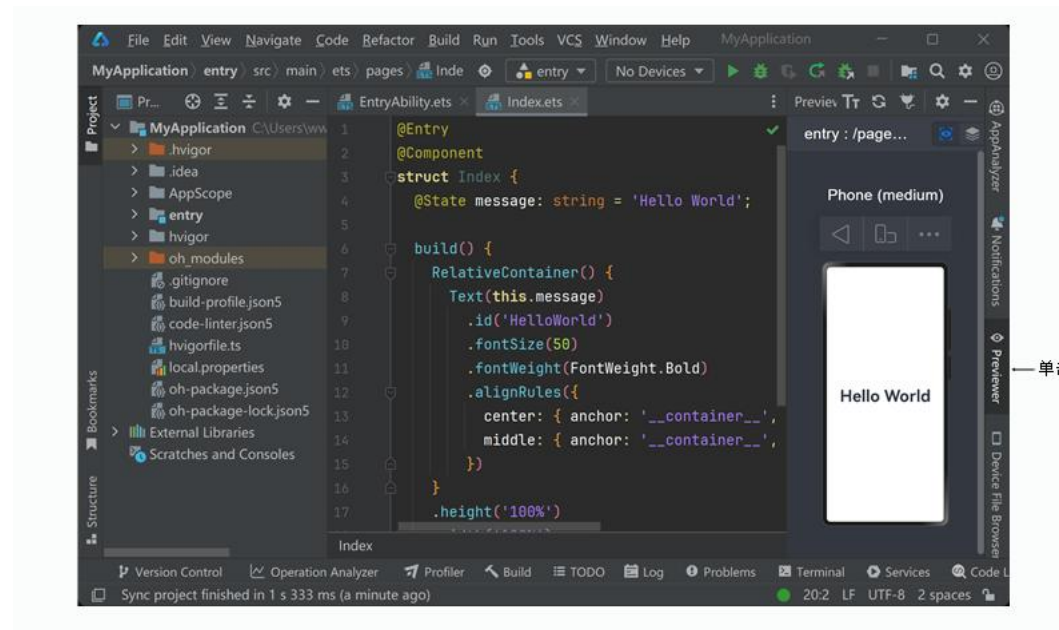
难点: 项目创建过程中的各种选项配置，及关键目录文件的功能和位置。

教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师展示一个完整项目的文件树，提问“如果让你找图标，第一步去哪?”引出目录结构重要性。
2. 讲授: 利用课件和导入的完整项目，讲解每个重要目录和文件的位置和作用，以及相关的文件、目录建立方法。
3. 演示: 教师现场新建项目，演示配置新项目的具体流程和相关的注意事项，并逐层展开目录，进一步讲解每个文件的职责，并最终利用预览器展示新建示例项目的运行效果。
4. 实践: 学生跟随创建 MyFirstApp 项目。



5. 总结: 根据现场实践的情况，统一讲解常见的问题；教师总结回顾关键注意点，布置作业

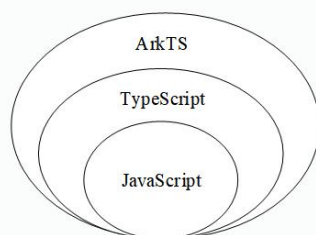
第3章 ArkTS 基础一（变量、类型与注释）（2学时）

教学目标

1、知识目标：

掌握 ArkTS 与 TypeScript、JavaScript 的关系与区别；

ArkTS并不是一门全新的语言。由于JavaScript应用广泛，TypeScript又使JavaScript更加严谨，华为选择在TypeScript的基础上开发ArkTS语言，从而使已经具备JavaScript或TypeScript基础的开发者可以快速上手鸿蒙应用的开发。



掌握 let、const 使用方法；

掌握 number、string、boolean、undefined、null 五种基本类型及其字面量写法；

掌握单行 // 与多行 /* */ 两种注释方式。

1. 单行注释

单行注释以 “//” 开始，到该行结束为止，示例代码如下。

```
console.log('你好'); // 调试输出
```

2. 多行注释

多行注释以 “/*” 开始，以 “*/” 结束，示例代码如下。

```
/*  
调试输出  
*/  
console.log('你好');
```

2、能力目标：

能在 DevEco Studio 中新建 BasicArkTS.ets 文件，独立编写包含三种类型变量的代码，控制台上打印格式统一为“变量名: 值”；

能使用 console.error 输出自定义报错信息。

`console.log()`语句的语法格式如下。

```
console.log(参数1, 参数2, ...);
```

- 小括号中的参数表示要输出的内容，可以传入1个或多个参数，多个参数使用英文逗号分隔，第1个参数必须是字符串，其余参数如果是其他类型数据则会被自动转为字符串。
- “...” 用于在文档中说明在参数2的后面还可以写更多参数，实际编写代码时不用写“...”。
- 语句末尾的分号“;”表示语句分隔符，在它后面可以写下一条语句。通过换行的方式可以省略语句分隔符。

3、素质目标：

培养“变量未声明不使用”的严谨习惯；

养成“写一行、测一行”的即时反馈意识；

树立“注释是写给别人看更是写给三个月后的自己看”的长远视角。

4、课程思政目标：

在强调注释精准时，引入“工匠精神”，体会严谨细致对国家安全与民生工程的重大意义。

重难点内容

重点：let 与 const 的区别、类型推断机制、console 日志查看位置。

难点：undefined 与 null 的语义差异及在实际调试中的误判场景。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：教师展示一段无类型注解的代码错误，引出静态类型检查的价值。
2. 讲授：利用课件展示各个 ArkTS 语言关键字的说明和使用方法。
3. 现场编码：结合每个关键字的使用方法说明，逐个演示使用，并让学生同步编写相关代码，在 Preview Log 查看输出。
4. 同伴互评：学生两人交换电脑，互相添加缺失注释并打分。
5. 总结：三个注意事项：即不要魔法数、不要无注释的函数、不要混用 undefined 与 null；布置练习作业。

第4章 ArkTS 基础二（运算符、流程控制）（4学时）

教学目标

1、知识目标：

熟悉算术、比较、逻辑、赋值四类运算符的优先级；

在ArkTS中，运算符遵循先后的运算顺序，这种顺序称作运算符的优先级。

结合方向	运算符
无	()
左 (new除外)	..、[]、new (有参数, 无结合性)
右	new (无参数)
无	++ (后置)、-- (后置)
右	!、~、- (负数)、+ (正数)、++ (前置)、-- (前置)、typeof、delete
右	**
左	*/、%
左	+、-
左	<<、>>、>>>
左	<、<=、>、>=、in、instanceof

掌握 if/else、switch 多分支逻辑；

掌握 for、while 编写计数控制与条件控制两类循环的方法；

掌握模板字符串拼接输出方式。

2、能力目标：

熟练使用各类运算符完成指定的相关数学运算和逻辑运算

熟练使用 for 循环和 while 循环完成各种循环逻辑。

3、素质目标：

培养“先画流程图、再写代码”的逻辑思维；

树立“循环条件三要素”自检清单习惯。

4、课程思政目标：

在调试循环边界时，强调毫厘之差可能导致重大事故，树立质量底线意识。

重难点内容

重点：运算符优先级、多分支逻辑和循环逻辑。

难点：分支条件设置，及循环边界值的极端情况处理；

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：教师回顾过去已学习的编程语言，如 C 语言或者 python 语言，引出其与 arkTS 编程语言的相似性和差异点，及相应的学习方式介绍。
2. 讲授：课件展示讲解各类运算符的使用方式；图示讲解各种选择结构和循环结构的原理逻辑。

算术运算符用于对两个**数字**或**变量**进行**算术运算**，与数学中的加法、减法、乘法、除法运算类似。

运算符	运算	示例	结果
+	加	3 + 3	6
-	减	6 - 3	3
*	乘	3 * 5	15
/	除	8 / 2	4
%	取模 (取余数)	5 % 7	5
**	幂运算	4 ** 2	16
++	自增 (前置)	a = 2; b = ++a;	a = 3; b = 3;
	自增 (后置)	a = 2; b = a++;	a = 3; b = 2;
--	自减 (前置)	a = 2; b = --a;	a = 1; b = 1;
	自减 (后置)	a = 2; b = a--;	a = 1; b = 2;

赋值运算符用于将运算符**右侧的值**赋给**左侧的变量**。

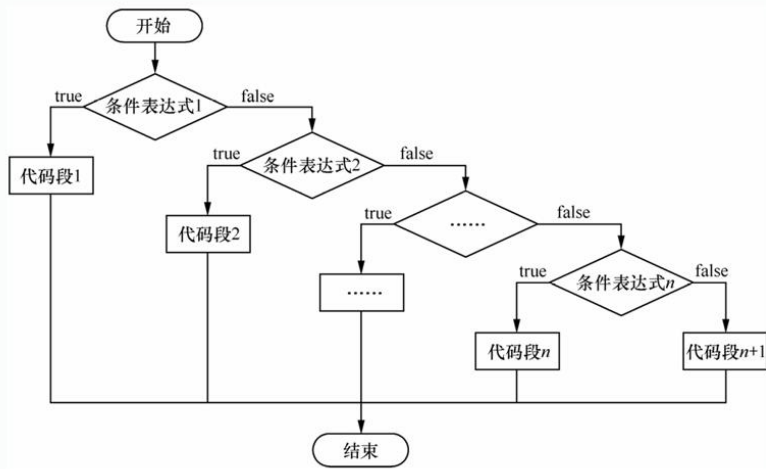
运算符	运算	示例	结果
=	赋值	a = 1, b = 2;	a = 1, b = 2;
+=	加并赋值	a = 1, b = 2; a += b;	a = 3, b = 2;
	字符串拼接并赋值	a = 'abc'; a += 'def';	a = 'abcdef';
--	减并赋值	a = 4, b = 3; a -= b;	a = 1, b = 3;
*=	乘并赋值	a = 4, b = 3; a *= b;	a = 12, b = 3;
/=	除并赋值	a = 4, b = 2; a /= b;	a = 2, b = 2;
%=	取模并赋值	a = 4, b = 3; a %= b;	a = 1, b = 3;

比较运算符用于对两个**数据**进行**比较**，比较返回的结果是**true**或**false**。

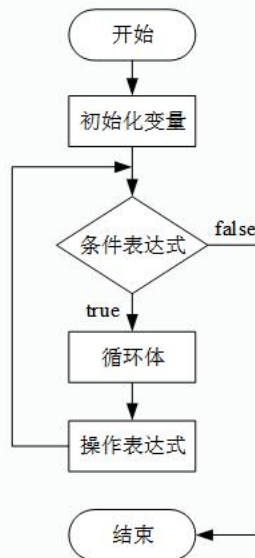
运算符	运算	示例	结果
>	大于	3 > 2	true
<	小于	3 < 2	false
>=	大于或等于	3 >= 2	true
<=	小于或等于	3 <= 2	false
==	等于	3 == 3	true
!=	不等于	3 != 3	false
===	全等	3 === 3	true
!==	不全等	3 !== 3	false

运算符	运算	示例	结果
&&	与	a && b	如果a的值为true, 则结果为b的值; 如果a的值为false, 则结果为a的值
	或	a b	如果a的值为true, 则结果为a的值; 如果a的值为false, 则结果为b的值
!	非	!a	如果a为true, 则结果为false; 如果a为false, 则结果为true

if...else if...else语句的运行流程如下图所示。



for语句的运行流程如下图所示。



3. 演示：使用简单案例演示相关运算的运算流程和结果
4. 实践：学生独立编程，教师提供多个小题目供选。
5. 竞赛：最先完成编程练习的同学讲解思路。

6. 总结：整体讲解过程中的常见问题；布置作业。

第 5 章 ArkTS 基础三（数组、枚举、函数）(4 课时)

教学目标

1、知识目标：

掌握数组(包括二维数组)相关的基本使用方法和场景；

掌握枚举关键字 `enum` 的使用方法和场景；

理解函数的概念和相关的使用方法，能编写普通函数、箭头函数、带默认参数函数等；了解常见的内置函数的使用方法；

理解变量的作用域和与闭包的概念

2、能力目标：

掌握 ArkTS 语言数组和函数的编写方法，能把给定的小功能目标通过编写对应的函数来做功能实现

3、素质目标：

培养“数据与行为分离”的函数式思维；

树立“枚举代替魔法字符串”的可读性意识。

4、课程思政目标：

在做功能实现拆分成小功能时，引入“工匠精神”，体会化整为零的解决模式。

重难点内容

重点：数组的定义和使用场景、各种类型函数的使用方法，`enum` 常量枚举。

难点：二维数组、箭头函数定义以及 `this` 指向。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

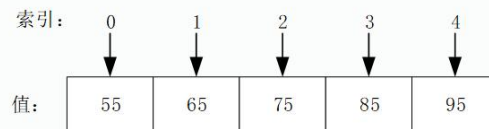
1. 导入：从常见的案例出发，如购物车列表中的物品总价计算，引出数组与函数需求。

2. 讲授：结合课件用图示说明数组的概念和使用方法；枚举的定义和声明方式；

函数的概念、自定义函数、函数作为值使用、箭头函数和常见的内置函数。

数组由0个或多个元素组成，数组中的每个元素由索引和值构成，其中，索引也称为下标，用数字表示，默认情况下从0开始依次递增，用于标识元素；值为元素的内容。

假设某个数组包含5个元素，这5个元素的值分别是55、65、75、85、95，该数组中索引和值的关系如下图所示。



1. 数组的基本使用方法

通过数组字面量创建数组的语法格式如下。

```
[值1, 值2, ...]
```

- 中括号“[]”中的值就是数组中的元素，元素之间使用英文逗号分隔，在值2的后面还可以有更多的值。值1的索引为0，值2的索引为1。
- 当数组中没有元素时，“[]”内不写任何内容。

枚举由一组命名值组成，这些命名值又称为枚举常量，用于预先定义一些值。

枚举的语法格式如下。

```
// 语法格式1
enum 枚举名1 { 枚举常量名1, 枚举常量名2, 枚举常量名3 }
// 语法格式2
enum 枚举名2 { 枚举常量名1 = 值1, 枚举常量名2 = 值2, 枚举常量名3 = 值2 }
```

- enum为声明枚举的关键字，建议将枚举名和枚举常量名中的每个单词首字母大写。
- 枚举常量的数量可以是0个或多个。
- 在语法格式1中有3个枚举常量，未对枚举常量赋值，它们的值默认为0、1、2。
- 在语法格式2中也有3个枚举常量，并为它们分别赋值。

定义了枚举后，使用“枚举名.枚举常量名”可以访问枚举常量的值。



在ArkTS中，函数分为**内置函数**和**自定义函数**。

- **内置函数**是指可以直接调用的函数。
- **自定义函数**是指用户自定义的实现某个特定功能的函数。

自定义函数在使用之前首先要**定义**，定义后才能**调用**，在要实现特定功能时可以**调用**相对应的**函数**。

3. 案例：现场演示把总价计算函数 `sumArray` 从普通函数一步步重构为箭头函数。
4. 实践：学生编写购物车商品总价计算功能
5. 总结：集中解决实践过程中的常见问题，布置课后作业。

第 6 章 ArkTS 基础四（类与对象）（2 课时）

教学目标

1、知识目标：

理解面向过程与面向对象思想差异；
掌握对象创建的方法，包括通过字面量创建对象和通过类创建对象；
掌握实例成员和静态成员的定义和访问方式。

2、能力目标：

掌握两种编程思想，能从两个不同的编程角度来实现同一个功能。
完成简单任务的面向对象封装

3、素质目标：

培养“对象抽象=属性+行为”的建模能力；
树立“静态成员属于类、实例成员属于对象”的清晰边界意识。

4、课程思政目标：

对于面向对象中所涉及的信息封装思想，可引出隐私信息保护，树立正确的数据伦理意识。

重难点内容

重点：面向过程与面向对象思想差异，对象创建的方法，实例成员和静态成员的定义和访问。

难点：面向过程与面向对象思想差异，实例成员与 `static` 成员的调用方式混淆。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：以日常生活中的做饭为例，引出面向过程的做法和面向对象式的做法。

表格		复制
维度	面向过程（自己做饭）	面向对象（叫对象帮忙）
核心单元	步骤（函数）	对象（冰箱、厨师、食材）
数据和动作	分离（鸡肉是数据，切是动作）	封装（鸡肉自带“被切”方法）
复用性	换菜需重写流程	换菜只需换对象参数或调用方法
现实映射	像“流水线”	像“分工合作”

2. 讲授：结合课件，讲解面向对象和面向过程各自的概念和思想异同；创建对象的两种方式以及实例成员和静态成员的定义和访问方式。

面向过程和面向对象是两种基本的程序设计思想。

在解决某个问题时：

- 面向过程以解决问题的过程为中心，强调的是函数的调用，程序被设计成一系列函数的集合。
- 面向对象以对象为中心，强调的是对象的属性和行为，程序被设计成一系列对象的集合。

下面对比面向过程和面向对象的优缺点，具体如下表所示。

分类	优点	缺点
面向过程	代码无浪费，无额外开销，适合对性能要求极其苛刻的情况和项目规模非常小、功能非常少的情况	不易维护、复用和扩展
面向对象	易维护、易复用和易扩展，适合业务逻辑复杂的大型项目	增加了额外的开销

3.2.1 通过字面量创建对象

通过字面量创建对象就是用大括号“{}”来标注对象成员，每个对象成员通过“键值对”的形式保存，即“key: value”的形式。

对象字面量的语法格式如下。

```
{ key1: value1, key2: value2, ... }
```

- key1和key2表示对象成员的名称，即属性名或方法名。
- value1和value2表示对象成员的值，即属性名对应的值或方法名对应的值。
- 多个对象成员之间使用逗号“,”隔开。
- 当对象中没有成员时，键值对可以省略，此时“{}”表示空对象。

定义类的语法格式如下。

```
class 类名 {  
    属性名: 类型 = 值;  
    方法名(参数1: 类型, 参数2: 类型, ...): 返回值类型 {  
        方法体  
    }  
}
```

- class是定义类的关键字。
- 类名与变量名类似，但是习惯上将单词的首字母大写。
- 在类的大括号“{}”中可以写0个或多个属性和方法，属性的写法类似给变量赋值，方法的写法类似函数定义。

定义类之后，通过类创建对象并赋值给变量的语法格式如下。

```
let 变量名: 类名 = new 类名;
```

- `new`关键字表示创建对象。
- 由于通过类创建对象时变量的类型可以被自动推断，所以可以省略上述语法中的“: 类名”。

另外，如果类名是一个函数，则表示通过构造函数创建对象，这是JavaScript中传统的创建对象的方式。

3. 实践：学生编码实现 `Student` 类，并创建两个对象，并实现对类内不同的成员属性和方法的调用。
4. 总结：总结实践过程中的常见问题，逐步实现并提供参考的示例编码；布置课后作业。

第 7 章 ArkTS 基础五（接口、泛型）（2 课时）

教学目标

1、知识目标：

理解接口 `interface` 的契约作用；
掌握不同的访问约束 `private/protected/public` 的可见性；
掌握类方法的实现和类的继承，以及接口的实现和继承；
理解泛型函数与泛型类的概念，以及掌握相关的使用方式

2、能力目标：

能够按任务要求，独立设定和定义相关类和接口的实现。
在编程时，能谨记高内聚、低耦合的原则，实现功能的同时，兼顾代码的可扩展性

3、素质目标：

培养“面向接口编程而非面向实现编程”的设计思维；
树立“强类型优于 `any`”的代码质量意识。

4、课程思政目标：

在讲解接口标准化时，引入中国制定 5G 标准的故事，激发民族自豪感与使命感。

重难点内容

重点：不同的访问约束 `private/protected/public` 的作用，类方法的实现和类的继承，以及接口的实现和继承；泛型函数与泛型类的概念。

难点：理解封装、继承、多态的实现和意义，并熟练运用。

(1) 封装

封装是指隐藏对象内部的实现细节，只对外开放操作接口。

- 接口是对象开放的属性和方法，无论对象的内部多么复杂，用户只需知道这些接口如何使用即可，而不需要知道内部的实现细节。
- 封装也有利于对象的修改和升级，无论一个对象内部的代码经过了多少次修改，只要不改变接口，就不会影响到使用这个对象时编写的代码。
- 通过合理设置访问控制修饰符，即可实现封装。

(2) 继承

继承允许一个类继承另一个类的属性和方法。

在实际开发中，使用继承不仅可以在保持接口兼容的前提下对功能进行扩展，而且可以增强代码的可复用性，为程序的修改和补充提供便利。

(3) 多态

多态是指同一个操作作用于不同的对象，会产生不同的执行结果。例如，项目中有一批相似的对象，如果它们的接口的命名、用法都是相同的，用户的学习成本就会比较低；而如果每种对象都有一套对应的接口，用户就需要学习每一种对象的使用方法，学习成本就会比较高。

多态的实现往往离不开继承，这是因为继承后多个对象就获取了相同的方法，然后可以根据每个对象的特点来改变同名方法的执行结果。

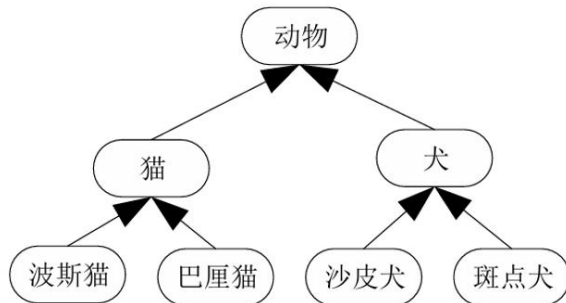
教学方法与过程

教学方法：讲授法 + 演示法 + 实践法

教学过程：

1. 导入：通过 USB 接口可以被多个品牌的各种产品使用的例子，引出接口的概念。
2. 讲授：利用课件讲解不同的访问约束 `private/protected/public` 的作用，类方法的实现和类的继承，以及接口的实现和继承；泛型函数与泛型类的概念等知识要点。

类的继承是指一个类继承另一个类的成员，并可以在不改变另一个类的前提下进行扩展。例如，猫和犬都属于动物，在程序中可以描述猫和犬继承自动物。同理，波斯猫和巴厘猫都继承自猫，沙皮犬和斑点犬都继承自犬，它们之间的继承关系如下图所示。



访问控制修饰符的作用范围如下表所示。

访问控制修饰符	同一个类内	子类	类外
public	允许访问	允许访问	允许访问
protected	允许访问	允许访问	不允许访问
private	允许访问	不允许访问	不允许访问

当在程序中访问了不允许访问的属性或方法时，DevEco Studio会使用红色波浪线标注错误代码，并且程序在编译时将会报错。

1. 定义泛型函数

在定义函数时，在函数名右侧添加“<泛型1, 泛型2, ...>”可以将函数定义成泛型函数。

“<>”中泛型的数量可以是1个或多个，它们相当于类型占位符，名称可以自定义，实际代表的类型需要在调用函数时指定，或者由ArkTS自动推断类型。

这些泛型可以在函数中作为类型使用，例如函数的参数类型、返回值类型、函数内变量的类型等。

定义泛型函数并调用该函数的示例代码如下。

```
function demo<T, U>(a: T, b: U): U {
    return b;
}
let str = demo<number, string>(1, '2');
```

2. 定义泛型类

在定义类时，在类名右侧添加 “<泛型1, 泛型2, ...>” 可以将类定义成泛型类，其用法与泛型函数类似。

```
class Demo<T, U> {  
    a: T;  
    b: U;  
    constructor(a: T, b: U) {  
        this.a = a;  
        this.b = b;  
    }  
}  
  
const demo = new Demo<number, string>(1, '2');
```

演示：对上述实现类的继承等知识要点进行相应的代码示例演示

4. 实践：学生自行实现相关知识点相应的代码示例，并尝试编写 USB 接口和相应的实现类，如 phone_usb，并测试调用。

5. 总结：统一梳理实践过程中的常见问题，布置作业。

第 8 章 ArkTS 基础六（内置对象、模块、错误处理）(6 课时)

教学目标

1、知识目标：

掌握常见的内置对象的基本功能和调用方法，如 Math, Number, Date, Array, String, JSON 等

掌握模块的基本导出和导入方式

掌握 try...catch 异常处理方式。

2、能力目标：

掌握基本的常见内置对象的应用，并能通过查询相关的说明文档了解进阶的功能调用方式。

编写能有效处理异常错误的、鲁棒性更强的应用程序

3、素质目标：

培养“模块即复用单元”的工程思维；

树立“异常不隐藏”的责任意识。

4、课程思政目标：

在异常处理的学习中，强调“异常不隐藏”的责任意识，树立实事求是，解决问题而不是隐藏问题的处事理念。

重难点内容

重点：常见的内置对象的基本功能和调用方法；模块 `import/export` 路径规则、`try/catch` 捕获运行时错误。

难点：模块 `import/export` 路径规则、`try/catch` 捕获运行时错误。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：通过“随机验证码”这个功能需求，引出 `Math.random` 这个常见内置对象方法。
2. 讲授：通过 课件+演示 逐个讲解常见内置对象的一些基本说明和使用方式，模块的导入导出方法以及 `try...catch` 异常处理方法。

`Math`对象表示数学对象，用于进行与数学相关的运算，其常用属性和方法如下表所示。

属性和方法	作用
<code>PI</code>	获取圆周率，结果为3.141592653589793
<code>abs(x)</code>	获取x的绝对值
<code>max(...values)</code>	获取所有参数中的最大值
<code>min(...values)</code>	获取所有参数中的最小值
<code>pow(x, y)</code>	获取基数 (x) 的指数 (y) 次幂，即x的y次幂
<code>sqrt(x)</code>	获取x的平方根，若x为负数，则返回NaN
<code>ceil(x)</code>	获取大于或等于x的最小整数，即向上取整
<code>floor(x)</code>	获取小于或等于x的最大整数，即向下取整
<code>round(x)</code>	获取x的四舍五入后的整数值
<code>random()</code>	获取大于或等于0且小于1的随机值

`...values`表示参数数量不固定。

3.6.2 Number对象

`Number`对象的常用方法如下表所示。

方法	作用
<code>toFixed(fractionDigits?)</code>	获取使用定点表示法表示给定数字的字符串。可选参数 <code>fractionDigits</code> 表示小数点后的位数，取值范围0~100，如果省略则被视为0
<code>toString()</code>	获取数字转换为字符串的结果

- 定点表示法是计算机中表示数字的一种方法，其小数点总是固定在指定的某一位置。
- `toString()`方法不仅在`Number`对象中可以用，任何可以转换成字符串的对象都可以调用该方法。

注意

受浮点数精度影响，`toFixed()`方法并不是严格按照四舍五入的规则进行处理的，它更适用于实现将不足的小数位补零。

3.6.3 Date对象

Date对象常用的获取日期和时间的方法如下。

方法	作用
getFullYear()	获取表示年的4位数字，如2023
getMonth()	获取月，取值范围为0~11（0表示1月，1表示2月，以此类推）
getDate()	获取月中的某一天，即获取日，取值范围为1~31
getDay()	获取星期，取值范围为0~6（0表示星期日，1表示星期一，以此类推）
getHours()	获取小时数，取值范围为0~23
getMinutes()	获取分钟数，取值范围为0~59
getSeconds()	获取秒数，取值范围为0~59
getMilliseconds()	获取毫秒数，取值范围为0~999
getTime()	获取从1970-01-01 00:00:00（UTC）到Date对象中存放的时间经历的毫秒数

3.6.3 Date对象

Date对象常用的设置日期和时间的方法如下。

方法	作用
setFullYear(year, month?, date?)	设置年为year，可选参数month和date可分别设置月和日
setMonth(month, date?)	设置月为month，可选参数date可设置日
setDate(date)	设置日为date
setHours(hours, min?, sec?, ms?)	设置时为hours，可选参数min、sec和ms可分别设置分、秒、毫秒
setMinutes(min, sec?, ms?)	设置分为min，可选参数sec和ms可设置秒、毫秒
setSeconds(sec, ms?)	设置秒为sec，可选参数ms可设置毫秒
setMilliseconds(ms)	设置毫秒为ms
setTime(time)	通过从1970-01-01 00:00:00（UTC）开始计时的毫秒数来设置时间

此外，Date()构造函数还有一个常用的静态方法now()，通过Date.now()的方式调用。该方法与getTime()方法作用类似，获取的时间是当前时间。

3.6.4 Array对象

在创建数组时，除了通过数组字面量“[]”创建数组外，还可以通过new Array()的方式创建数组，具体语法格式如下。

```
new Array(元素1, 元素2, ...)
```

- “元素1, 元素2, ...”是指数组中实际保存的元素，元素的数量可以是0个或多个，各元素之间使用英文逗号分隔。
- 若元素的数量是0个，则表示创建一个空数组。

注意

如果只传入一个number数据类型的参数，则表示创建一个拥有指定数量的空位的数组，空位是指在数组中预留了元素的空间，但没有给元素赋值，空位会被计算在数组长度内。

3.6.5 String对象

下面通过代码演示字符串搜索、匹配、替换和大小写转换，示例代码如下。

```
let str = 'Hello World';
// 搜索字符串中的 "World"
console.log(str.search(/World/));           // 输出结果: 6
// 匹配字符串中的 "Hello"
console.log(str.match(/Hello/));           // 输出结果: Hello
// 将字符串中的 "World" 替换为 "!"
console.log(str.replace(/World/, '!'));     // 输出结果: Hello !
// 获取字符串的小写形式
console.log(str.toLowerCase());             // 输出结果: hello world
// 获取字符串的大写形式
console.log(str.toUpperCase());             // 输出结果: HELLO WORLD
```

JSON对象的常用方法如下表所示。

方法	作用
JSON.parse(text, reviver?)	解析JSON字符串，返回解析后的值或对象。text表示解析的字符串；可选参数reviver是一个函数，用于在返回前对所得到的对象执行操作
JSON.stringify(value, replacer?, space?)	将一个对象或值转换为JSON字符串。value表示要转换的对象或值。可选参数replacer可以是一个函数或数组，如果是函数则用于选择性地替换值，如果是数组则可选择性地仅包含数组指定的属性；可选参数space指定缩进用的字符串，用于美化输出，如果参数是个数字，代表有多少的空格，上限为10

3. 实践：提供多个编程小题目(均可通过常见内置对象实现)供学生现场练习，过程中要求使用异常处理，完成后尝试把实现的功能模块导出。
4. 总结：校对答案，统一讲解实践过程中的常见问题；布置作业。

第9章 ArkUI 基础一（组件概念与自定义）（2课时）

教学目标

1、知识目标：

组件的概念与分类，以及组件的基本使用方法
通过查阅鸿蒙开发文档的方式学习 ArkUI 组件的基本流程
自定义组件的基本方式

2、能力目标：

掌握 ArkUI 相关的基本概念，理解基本的 UI 界面构建和交互功能实现的基本逻辑和方式，为后续进一步学习做基础。

3、素质目标：

培养“像搭积木一样搭界面”的组件化思维；
树立“一个组件只做一件事”的单一职责意识。

4、课程思政目标：

在组件命名时强调尊重文化差异，如避免使用带有宗教色彩的词汇，树立包容性设计观。

重难点内容

重点：组件的概念与分类，以及组件的基本使用方法；自定义组件的基本方式
难点：常用装饰器如@Component、@Entry、@State、@Preview 的功能和使用方式

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：通过日常生活中的乐高积木的搭建，引出组件化概念。
2. 讲授：结合课件+演示，讲解组件的概念与分类，以及组件(包括自定义组件)的基本使用方法。

组件可以按照其功能和用途进行分类，ArkUI中的组件分类如下。

- ① **基础组件**：提供了构建UI的基础，如按钮、单行文本输入框、标签等，它们通常是构建其他复杂组件的基础。
- ② **容器组件**：帮助开发者构建不同样式的页面布局，如栅格系统、弹性布局、网格布局等，使页面的布局更加灵活。
- ③ **媒体组件**：用于处理和展示音视频内容，丰富UI的视觉和交互体验。
- ④ **绘制组件**：用于绘制图形，例如线条、矩形、圆形等，通常用于实现涂鸦板、绘图工具等应用场景。
- ⑤ **画布组件**：提供了一个可绘制的画布，开发者可以在画布上绘制各种图形、文本等内容。这些组件通常用于创建自定义的图形界面或实现特定的绘图功能。
- ⑥ **其他组件**：包括操作块组件、弹出框组件、气泡组件、工具栏组件等。

ArkUI为组件提供了属性和事件，包括所有组件通用的属性和事件，以及每个组件特有的属性和事件。关于属性和事件的介绍如下。

- ① **属性**用于定义组件的样式、数据、特性等。其中，定义组件样式的属性与网页中的CSS样式属性类似，从而能让Web前端开发者快速上手鸿蒙应用的UI开发。
- ② **事件**用于定义组件的交互逻辑。开发者可以为组件的事件设置事件处理程序，当用户的交互触发了事件时，程序就会执行事件处理程序，从而处理用户的操作。

下面列举一些ArkUI中常用的通用属性和通用事件。

类型	名称	说明
通用属性	width	设置宽度
	height	设置高度
	margin	设置外边距
	padding	设置内边距
	background	设置背景
	backgroundColor	设置背景颜色
	border	设置边框的样式
	borderRadius	设置边框的圆角半径
通用事件	onClick	点击事件

在entry/src/main/ets/pages目录下，每个文件对应一个页面，在每个文件中可以定义多个自定义组件。打开该目录下的Index.ets文件，会看到里面的代码主要包括@Entry、@Component、struct Index {}、@State和build()，示例代码如下。

```
@Entry
@Component
struct Index {
    @State message: string = 'Hello World';
    build() {
        ..... (此处省略一些代码)
    }
}
```

1. 装饰器

装饰器用于装饰类、结构、方法以及变量，并赋予特殊的含义。装饰器既可以与被装饰的内容写在同一行，用空格分隔，也可以单独写在一行，写在被装饰内容的上方。多个装饰器可以同时使用，用空格或换行分隔即可。

常用的装饰器有：

- @Component
- @Entry
- @State
- @Preview

3. 实践：学生自行打开已有项目中相关目录和文件，结合上述演示样，尝试实现相关知识点相应的代码，并尝试编写自定义一个简单的组件，如一个个人名片，即在界面简单显示个人的姓名、班级、学号等信息，并测试运行。

4. 总结：统一梳理实践过程中的常见问题，布置作业。

第 10 章 ArkUI 基础二（常用布局与组件）（4 课时）

教学目标

1、知识目标：

常用组件的使用方式，包括：Column, Row, Image, Text, TextInput, Button, List 等。

2、能力目标：

能综合运用 Column、Row、TextInput、Button 完成简单页面 UI（无业务逻辑），能在 List 中用 ForEach 渲染图文列表。

3、素质目标：

培养“先画草图、再写布局”的可视化思维；
树立“输入框需提示隐私政策”的用户保护意识。

4、课程思政目标：

在输入信息采集时，讨论提示信息的必要性，树立“一个好产品应有尽可能低的学习和使用成本”的产品思维和服务意识。

重难点内容

重点：常用组件的使用方式，特别是：Column, Row。

难点：Column/Row 主轴方向、List 的 ForEach 使用。

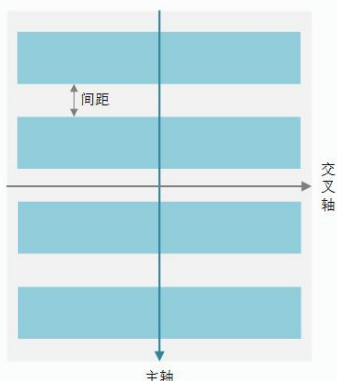
教学方法与过程

教学方法：讲授法+演示法+实践法

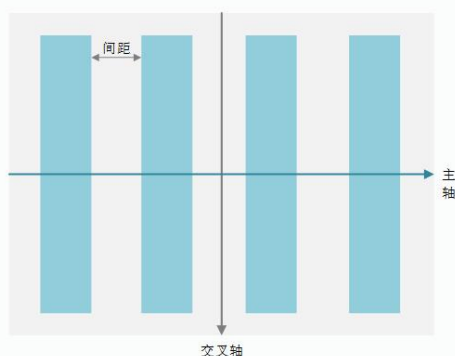
教学过程：

1. 导入：从常用的应用 app 的界面展示，如微信，或者淘宝等的一些相关界面，引出“用哪几个组件可拼成”的问题。
2. 讲授：结合课件+演示，展示各个常用组件的使用方法。

Column容器（Column组件创建的容器）内的子组件会按照代码中的顺序依次在垂直方向上排列。如果Column容器中的子组件的内容超出了父容器组件的宽度或高度，超出部分不会换行也不会出现滚动条。



Row容器（Row组件创建的容器）内的子组件会按照代码中的顺序依次在水平方向上排列。如果Row容器中的子组件的内容超出了父容器组件的宽度或高度，超出部分不会换行也不会出现滚动条。



Image组件的常用属性如下表所示。

属性	说明
alt	用于设置图像加载时显示的占位图
objectFit	用于设置图像的填充效果
interpolation	用于设置图像的插值效果，可缓解图像在缩放时的锯齿问题。SVG格式的图像不支持使用该属性
objectRepeat	用于设置图像的重复样式，从中心点向两边重复，剩余空间不足放下一张图像时图像会被截断。SVG格式的图像不支持使用该属性
renderMode	用于设置图像的渲染模式为原色或黑白。SVG格式的图像不支持使用该属性
fillColor	用于设置填充颜色，设置后填充颜色会覆盖在图像上。该属性仅对SVG格式的图像生效，设置后会替换图像的填充颜色

Text组件除了设置文本内容外，还可以通过设置各种属性来自定义文本的外观和行为。
Text组件的常用属性如下表所示。

属性	说明
textAlign	用于设置文本段落水平方向的对齐方式
textOverflow	用于设置文本超长时的显示方式
lineHeight	用于设置文本的行高
fontSize	用于设置文本的字号
fontColor	用于设置文本的字体颜色
fontWeight	用于设置文本的字体粗细
letterSpacing	用于设置文本字符间距
textCase	用于设置文本大小写
decoration	用于设置文本装饰线样式及其颜色

TextInput组件是单行文本输入框组件，其语法格式如下。

```
TextInput(value)
```

value参数是一个TextInputOptions类型的对象，该对象包含以下属性。

- placeholder属性：用于设置单行文本输入框无输入时的提示文本。
- text属性：用于设置单行文本输入框当前的文本内容。

下面通过代码演示如何设置单行文本输入框当前的文本内容，示例代码如下。

```
TextInput({ text: '请输入账号或手机号' })
```

Button组件是按钮组件，可快速创建不同样式的按钮，其语法格式如下。

```
Button(label, options)
```

- label参数用于设置按钮的文本内容。
- options参数为一个对象，包含type属性和stateEffect属性，具体会在后面进行讲解。

当不传递label参数时，可以在Button组件内嵌套其他组件，语法格式如下。

```
Button() {  
  // 其他组件  
}
```

4. 实践：学生独立复刻实现一个简单登录页，并预览器上试看效果，教师巡视纠偏。
5. 总结：统一梳理实践过程中的常见问题，布置作业。

第 11 章 ArkUI 基础三（条件与循环渲染、组件导入和导出）(2 课时)

教学目标

1、知识目标：

条件渲染的实现方式和使用场景
循环渲染的实现方式和使用场景
组件的导入和导出

2、能力目标：

能根据任务场景选择特定的实现方式，并通过编码的形式完成相应的 UI 渲染实现。

3、素质目标：

培养“数据驱动视图”的响应式思维；
树立“状态提升”的复用意识。

4、课程思政目标：

在编码任务实现的过程中，强调选用低复杂度的实现方式，从细节处体现节能减排，倡导节约型社会价值观。

重难点内容

重点：条件渲染的实现方式和使用场景；循环渲染的实现方式和使用场景。

难点：ForEach 的使用；@State 变量更新后的 UI 刷新机制

ForEach()函数的语法格式如下。

```
ForEach(  
  arr: Array,  
  itemGenerator: (item: 类型, index?: number) => void,  
  keyGenerator?: (item: 类型, index?: number): string => string  
)
```

- 参数arr表示数据源，它是一个数组。
- 参数itemGenerator表示组件生成函数。
- 参数keyGenerator表示键生成函数。

在这两个函数中，item参数表示数组中元素的值，index参数表示数组中元素的索引。

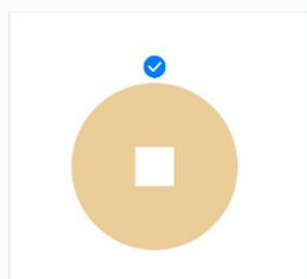
教学方法与过程

教学方法：讲授法+演示法+实践法

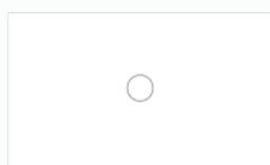
教学过程：

1. 导入：从购物 App 中多个格式重复的物品清单列表的筛选，引出条件渲染和循环渲染的必要性。
2. 讲授：结合课件+演示讲解条件渲染和循环渲染的使用方式和具体的实现效果。

在开发中，有时需要根据某个条件决定是否渲染某个组件，此时可以使用条件渲染语句。条件渲染语句包括if语句、if...else语句、if...else if...else语句。



渲染图像



不渲染图像

循环渲染语句主要是通过ForEach()函数实现的。

使用ForEach()函数可以基于数组进行循环渲染，在渲染过程中，系统会为每个数组元素生成一个唯一且持久的键，用于标识对应的组件。

当这个键发生变化时，ArkUI将视为该数组元素已被替换或修改，并会基于新的键创建一个新的组件。



3. 实践：学生实现单一条件渲染和循环渲染的使用方式，教师提供断点调试指导。
4. 总结：布置作业：购物 App 中多个格式重复的物品清单列表的筛选。

第 12 章 路由（页面跳转与返回控制） (2 课时)

教学目标

1、知识目标：

掌握页面跳转和返回的原理和操作方式

掌握页面返回前询问的方法

了解跨模块的页面跳转

2、能力目标：

能对任务进行拆分，使用多页面方式设计，并通过编码完成页面间的跳转和返回逻辑

3、素质目标：

培养“页面即服务、参数即契约”的接口思维；

树立“用户数据需加密传输”的安全意识。

4、课程思政目标：

在讨论页面路由拦截时，引入“未成年人防沉迷”国家政策，引导学生思考技术如何服务社会治理。

重难点内容

重点：页面跳转和返回的原理和操作方式，页面返回前询问的方法。

难点：`pushUrl()`方法和`replaceUrl()`方法，页面跳转时的参数传递。

教学方法与过程

教学方法：讲授法+演示法+实践法

教学过程：

1. 导入：通过购物输入的页面跳转样例，引出路由的概念。

页面跳转是指从一个页面导航到另一个页面，并且可以将数据从一个页面传递到另一个页面。下面演示一个购物软件的页面跳转过程，如下图所示。



2. 讲授：通过课件+演示的方式，讲解页面跳转相关的概念和逻辑，使用的函数接口以及具体的操作方法

在鸿蒙应用中，路由负责管理不同页面之间的跳转和参数传递，它基于轻量级的栈式管理结构，每个页面都有唯一的标识符，所有被访问过的页面都会被放置在页面栈中。

每当页面跳转时，路由会根据标识符对页面进行入栈或出栈操作，实现页面跳转和管理。

router对象提供了pushUrl()方法和replaceUrl()方法，这两个方法都可以实现页面跳转，它们的区别在于目标页是否会替换当前页。

方法	说明
pushUrl()	目标页不会替换当前页，而是将当前页压入页面栈以保留当前页的状态，跳转后可以通过返回操作回到当前页。这种模式适用于常规的页面导航，例如从一个列表页跳转到详情页
replaceUrl()	目标页会替换当前页，当前页会被销毁并释放资源。跳转后无法通过返回操作回到当前页。这种模式适用于需要替换当前页的场景，例如在登录成功后直接跳转到主页

`router`对象的`back()`方法用于实现页面返回功能，这个方法会将用户导航到上一个页面，而且不会重新初始化上一个页面，因此上一个页面的状态和数据都会被保留。

6.1.3 在页面返回前询问

1. 使用内置询问对话框

`router`对象的`showAlertBeforeBackPage()`方法用于实现在返回时弹出询问对话框的功能。

在调用`back()`方法之前，可以通过调用`showAlertBeforeBackPage()`方法启用询问对话框，启用后，当调用`back()`方法时，询问对话框就会自动弹出。

6.1.3 在页面返回前询问

2. 使用自定义询问对话框

自定义询问对话框可以使用`promptAction`对象实现。

使用该对象的`showDialog()`方法可以实现类似使用`showAlertBeforeBackPage()`方法启用询问对话框的效果，并可以获取用户点击的是哪个按钮，从而在用户点击后执行相应的逻辑。

3. 实践：完成两个简单的页面 A 和 B，并实现页面间的跳转和返回，以及中间参数传递。
4. 总结：统一梳理实践过程中的常见问题，布置作业。

第 13 章 生命周期与状态管理(2 课时)

教学目标

1、知识目标：

掌握自定义组件的生命周期方法
掌握 UIAbility 的生命周期方法
理解状态管理的概念
组件状态共享的实现方式

2、能力目标：

能利用生命周期方法，实现多个页面间跳转返回时确认
掌握使用状态变量来控制相关的 UI 变化的原理逻辑，并能熟练用此方式实现一些相对复杂的任务

3、素质目标：

培养“生命周期是应用的脉搏”的系统观；
树立“全局状态慎用、局部状态优先”的设计原则。

4、课程思政目标：

调试解读日志时，强调“不打印身份证号等敏感信息”，培养守法合规意识。

重难点内容

重点：自定义组件的生命周期方法，状态管理的概念及组件状态共享的实现方式
难点：组件状态共享的实现方式

教学方法与过程

教学方法：讲授法+演示法+实践法

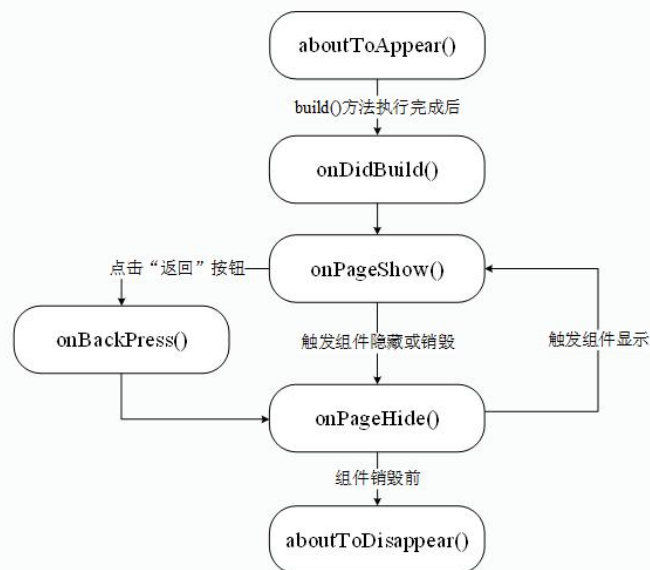
教学过程：

1. 导入：从日常手机应用的启动和关闭，引申到内部应用的生命周期概念。
2. 讲授：通过课件+演示的方式，讲解自定义组件和 UIAbility 的生命周期方法、状态管理的概念及组件状态共享的实现方式。

自定义组件包括入口组件和非入口组件，入口组件的生命周期方法如下表所示。

方法	说明
aboutToAppear()	在创建自定义组件时，在build()方法执行前执行。该方法允许更改状态变量，更改将在后续执行的build()方法中生效
onDidBuild()	在build()方法执行完成之后执行。不建议在该方法中更改状态变量、调用animateTo()等，这可能会导致不稳定的UI表现
aboutToDisappear()	在自定义组件销毁之前执行。该方法不允许更改状态变量，若修改可能会导致应用程序行为不稳定
onPageShow()	页面每次显示时执行一次，包括页面切换、应用进入前台等场景。该方法在build()方法之后触发
onPageHide()	页面每次隐藏时执行一次，包括页面跳转、应用进入后台等场景
onBackPressed()	当用户点击“返回”按钮 < 时触发。当返回值为true时表示页面自己处理返回逻辑，不进行页面路由；当返回值为false时表示使用默认的路由返回逻辑；不设置返回值则按照返回值为false处理

入口组件的生命周期方法的流程如下图所示。

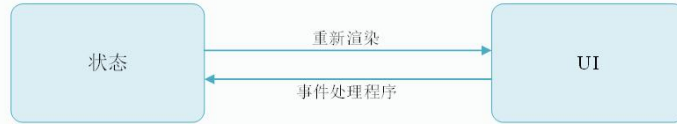


当用户打开、切换和返回到应用时，应用中的UIAbility会在其生命周期的不同状态之间转换。UIAbility提供了一系列生命周期方法，具体如下表所示。

方法	说明
onCreate()	当UIAbility被创建时调用，可以进行页面初始化操作，例如定义变量、资源加载等
onDestroy()	当UIAbility被销毁时调用，可以执行资源释放、清理等操作
onNewWant()	当UIAbility启动时调用，可以处理启动UIAbility的逻辑

ArkUI提供了状态管理机制，实现了当状态变量的值发生改变时重新渲染UI。如果不使用状态变量，UI只会在初始化时渲染，后续不会再渲染。

用户可以通过触发组件的事件来执行事件处理程序，事件处理程序可以改变状态，状态的变化会引起UI的重新渲染。



7.3.1 @Prop装饰器

@Prop装饰器用于装饰子组件中的状态变量，这个状态变量的值由父组件传递。使用@Prop装饰器后，当父组件的状态改变时，子组件的状态会立即改变。

在使用@Prop装饰器时应注意以下两点。

- ① @Prop装饰器只适用于非入口组件，不适用于入口组件。
- ② 被@Prop装饰器装饰的状态变量可以设置初始值，也可以不设置初始值，由父组件传递的值会覆盖初始值。

7.3.2 @Link装饰器

@Link装饰器用于装饰子组件中的状态变量，这个状态变量的值由父组件传递并保持同步。使用@Link装饰器后，当父组件的状态改变时，子组件的状态会立即改变；当子组件的状态改变时，父组件的状态也会立即改变。

在使用@Link装饰器时应注意以下两点。

- ① @Link装饰器只适用于非入口组件，不适用于入口组件。
- ② 被@Link装饰器装饰的状态变量可以设置初始值，也可以不设置初始值，由父组件传递的值会覆盖初始值。

7.3.3 @Provide和@Consume装饰器

① @Provide装饰器用于装饰上游组件（或称为祖先组件）中的状态变量，作为状态的“提供者”。被@Provide装饰器装饰的状态变量必须设置初始值。

② @Consume装饰器用于装饰下游组件（或称为后代组件）中的状态变量，作为状态的“消费者”，被@Consume装饰器装饰的状态变量接收和使用来自上游组件的状态，不允许设置初始值。

使用@Provide和@Consume装饰器后，当上游组件中的状态改变时，下游组件中的状态会立即改变；当下游组件中的状态改变时，上游组件中的也会立即改变。

3. 实践：编码实现以下功能：创建 LifePage1.ets 和 LifePage2.ets 两个页面，在这两个页面中编写生命周期方法，实现页面跳转前询问。
4. 总结：统一梳理实践过程中的常见问题，布置作业。