2025年夏季《移动软件开发》实验报告

姓名：邓林 学号：23020007014

姓名和学号	**邓林，23020007014**
本实验属于哪门课程?	中国海洋大学25夏《移动软件开发》
实验名称?	实验5：第一个 HarmonyOS 应用
博客地址?	2025年夏季《移动软件开发》实验报告5-CSDN博客
Github仓库地址?	https://github.com/xixiyhaha/2025Mobile-software-development.git

（备注：将实验报告发布在博客、代码公开至 github 是 **加分项**，不是必须做的）

一、实验目标

1、掌握如何构建 HarmonyOS应用；

2、掌握应用程序包结构、资源文件的使用；

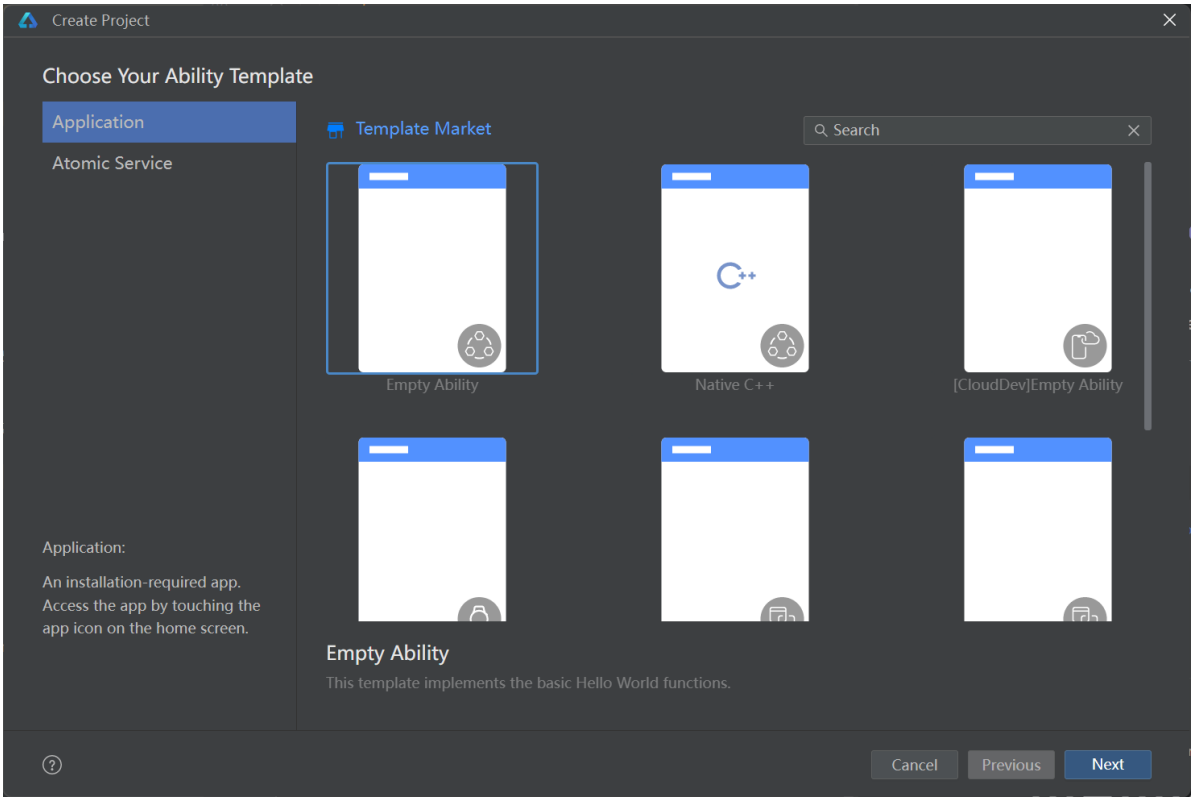
3、掌握ArkTS的核心功能和语法等基础知识，为后续的应用开发奠定基础。

二、实验步骤

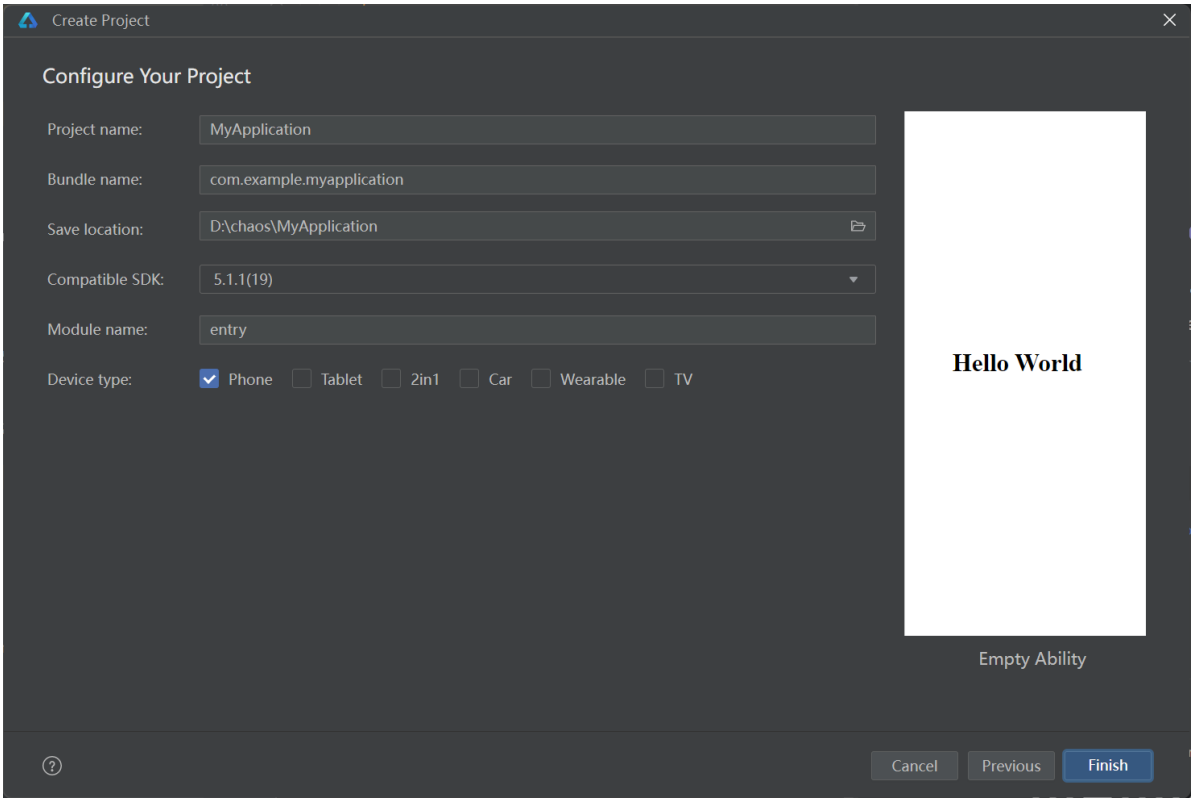
1、下载并安装最新版本 - 下载中心 - 华为开发者联盟；

2、打开DevEco Studio，单击Create Project创建一个新工程；

3、选择Application应用开发（本文以应用开发为例，Atomic Service对应为元服务开发），选择模板Empty Ability，单击Next进行下一步配置；



4、进入配置工程界面，Compatible SDK表示兼容的最低API Version，此处以选择5.1.1(19)为例，其他参数保持 默认设置即可， 单击Finish，工具会自动生成示例代码和相关资源，等待工程创建完成；



5、工程同步完成后，在Project窗口，单击entry > src > main > ets > pages，打开Index.ets文件，将页面从 RelativeContainer相对布局修改成Row/Column线性布局，代码如下：

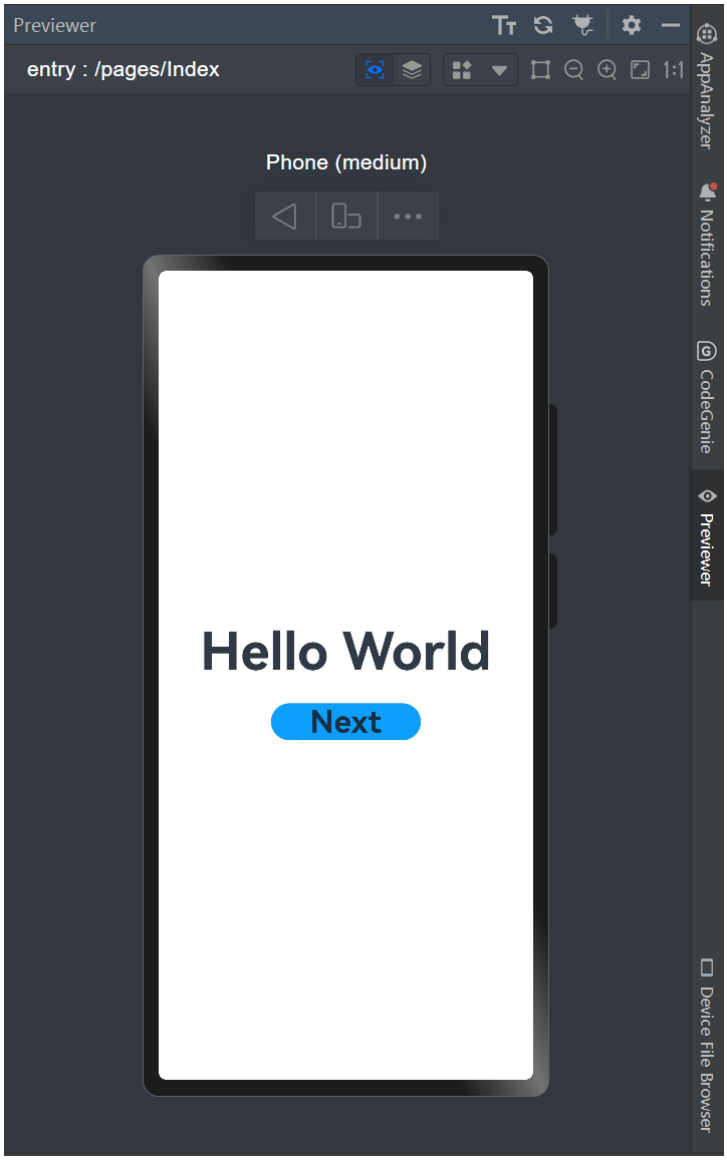
```
// Index.ets
@Entry
@Component
struct Index {
  @State message: string = 'Hello World';

  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        }
      .width('100%')
    }
    .height('100%')
  }
}
```

6、在默认页面基础上，我们添加一个Button组件，作为按钮响应用户onClick事件，从而实现跳转到另一个页 面。代码如下：

```
// Index.ets
@Entry
@Component
struct Index {
  @State message: string = 'Hello World';

  build() {
    Row() {
      Column() {
```

```
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        // 添加按钮，以响应用户onClick事件
        Button() {
          Text('Next')
            .fontSize(30)
            .fontWeight(FontWeight.Bold)
        }
        .type(ButtonType.Capsule)
        .margin({
          top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        }
      .width('100%')
    }
    .height('100%')
  }
}
```

7、在编辑窗口右侧工具栏，单击Previewer，打开预览器。第一个页面效果如下图所示：



8、新建第二个页面文件。在Project窗口，打开entry > src > main > ets，右键单击pages文件夹，选择New > ArkTS File，命名为Second，单击回车键；

9、配置第二个页面的路由。在Project窗口，打开entry > src > main > resources > base > profile，在main_pages.json文件中的"src"下配置第二个页面的路由"pages/Second"。示例如下：

```
{
     "src": [
        "pages/Index",
        "pages/Second"
    ]
}
```

10、参照第一个页面，在第二个页面添加Text组件、Button组件等，并设置其样式。Second.ets文件的示例如 下：

```
// Second.ets
@Entry
@Component
struct Second {
  @State message: string = 'Hi there';

  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        Button() {
          Text('Back')
            .fontSize(30)
            .fontWeight(FontWeight.Bold)
        }
        .type(ButtonType.Capsule)
        .margin({
          top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        }
      .width('100%')
    }
    .height('100%')
  }
}
```

11、实现页面间的跳转。在第一个页面中，跳转按钮绑定onClick事件，单击按钮时跳转到第二页。Index.ets文件的示例如下：

```
// Index.ets
import { BusinessError } from '@kit.BasicServicesKit';

@Entry
@Component
struct Index {
```

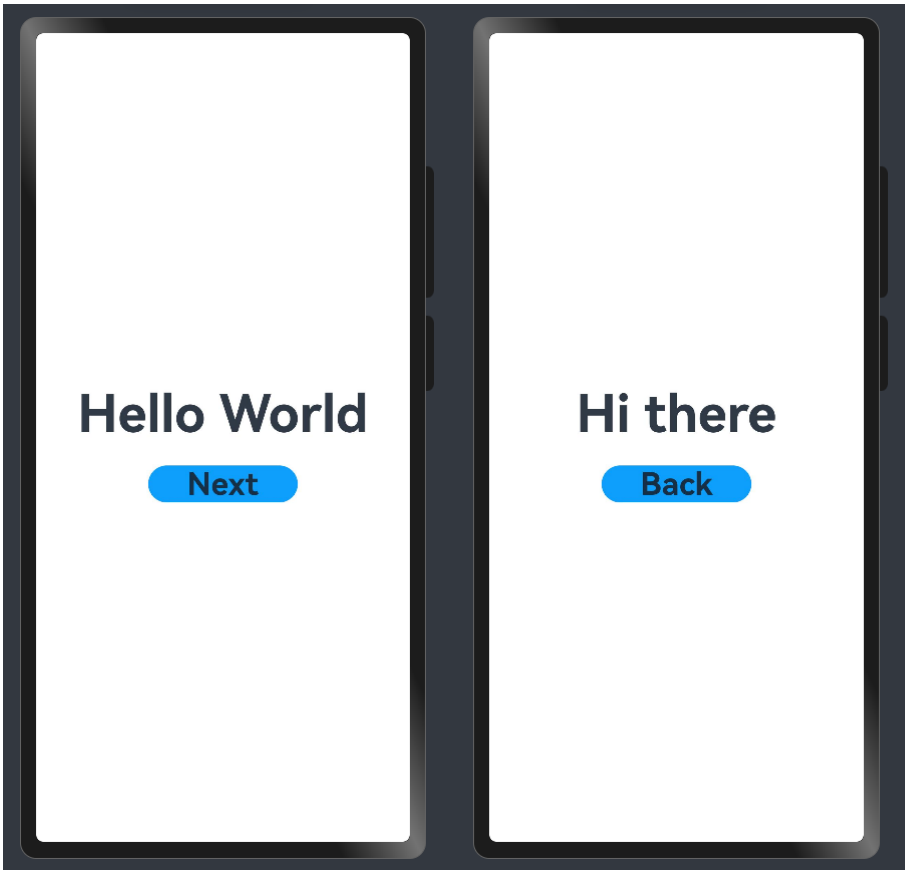
```
  @State message: string = 'Hello World';

  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        // 添加按钮，以响应用户onClick事件
        Button() {
          Text('Next')
            .fontSize(30)
            .fontWeight(FontWeight.Bold)
        }
        .type(ButtonType.Capsule)
        .margin({
          top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        // 跳转按钮绑定onClick事件，单击时跳转到第二页
        .onClick(() => {
          console.info(`Succeeded in clicking the 'Next' button.`)
          // 获取UIContext
          let uiContext: UIContext = this.getUIContext();
          let router = uiContext.getRouter();
          // 跳转到第二页
          router.pushUrl({ url: 'pages/Second' }).then(() => {
            console.info('Succeeded in jumping to the second page.')
          }).catch((err: BusinessError) => {
            console.error(`Failed to jump to the second page. Code is
${err.code},
message is ${err.message}`)
          })
        })
        }
      .width('100%')
    }
    .height('100%')
  }
}
```

12、在第二个页面中，返回按钮绑定onClick事件，单击按钮时返回到第一页。Second.ets文件的示例如下：

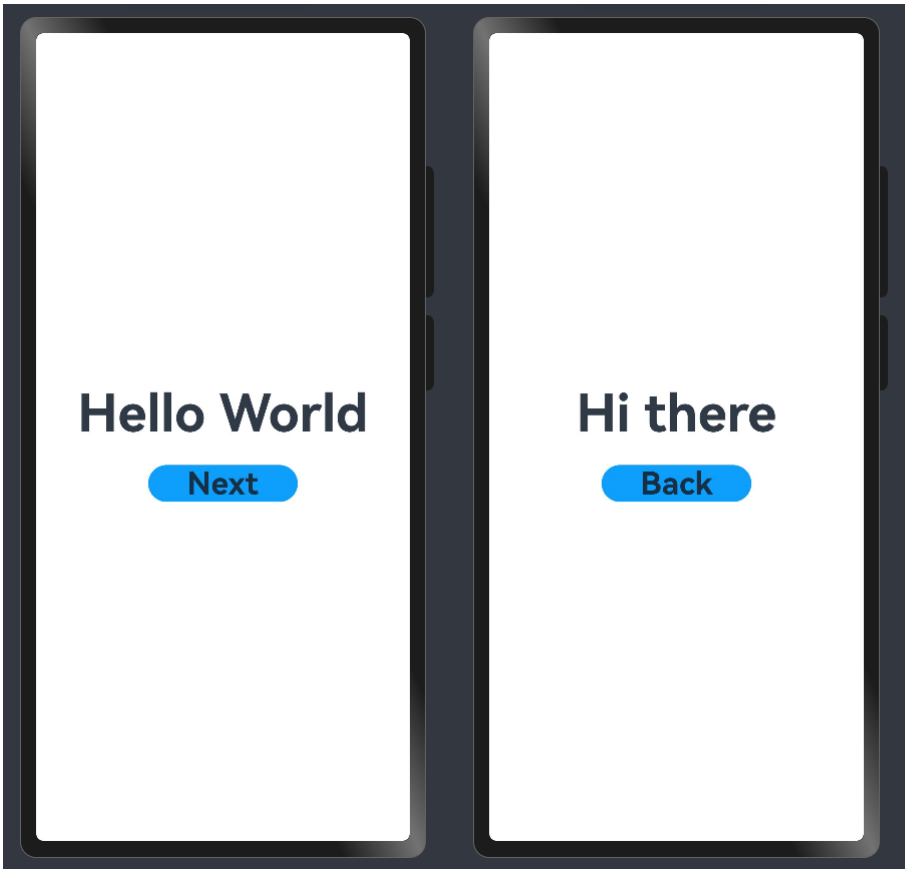
```
// Second.ets
import { BusinessError } from '@kit.BasicServicesKit';

@Entry
@Component
struct Second {
  @State message: string = 'Hi there';
  build() {
    Row() {
      Column() {
        Text(this.message)
```

```
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
        Button() {
          Text('Back')
            .fontSize(30)
            .fontWeight(FontWeight.Bold)
        }
        .type(ButtonType.Capsule)
        .margin({
          top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('5%')
        // 返回按钮绑定onClick事件，单击按钮时返回到第一页
        .onClick(() => {
          console.info(`Succeeded in clicking the 'Back' button.`)
          // 获取UIContext
          let uiContext: UIContext = this.getUIContext();
          let router = uiContext.getRouter();
          try {
            // 返回第一页
            router.back()
            console.info('Succeeded in returning to the first page.')
          } catch (err) {
            let code = (err as BusinessError).code;
            let message = (err as BusinessError).message;
            console.error(`Failed to return to the first page. Code is ${code},
message is ${message}`)
          }
        })
        }
      .width('100%')
    }
    .height('100%')
  }
}
```

13、打开Index.ets文件，单击预览器中的按钮进行刷新。效果如下图所示：



三、程序运行结果



四、问题总结与体会

描述实验过程中所遇到的问题，以及是如何解决的。有哪些收获和体会，对于课程的安排有哪些建议。

收获和体会：

通过本次“ 第一个 HarmonyOS 应用 ” 实验，我从对鸿蒙开发的零基础认知，逐步实现了从概念理解到实践落地的突破。实验中，我不仅掌握了 DevEco Studio 的使用、ArkTS 工程的创建与目录结构梳理，还通过构建含页面跳转和返回功能的应用，实践了基于 ArkTS 的声明式开发范式、Row/Column 线性布局的运用，以及页面路由 router 模块的导入与事件绑定，成功完成从 Index 页面到 Second 页面的跳转与返回逻辑编写。整个过程让我直观感受到鸿蒙开发中 “数据驱动 UI 更新” 的核心思想、模块化的工程设计理念，以及 ArkTS 语法的简洁高效，不仅夯实了应用开发的基础技能，更让我对 HarmonyOS 生态的兼容性与开发便捷性有了深刻认知，为后续探索更复杂的鸿蒙功能奠定了扎实基础。