

2025年夏季《移动软件开发》实验报告

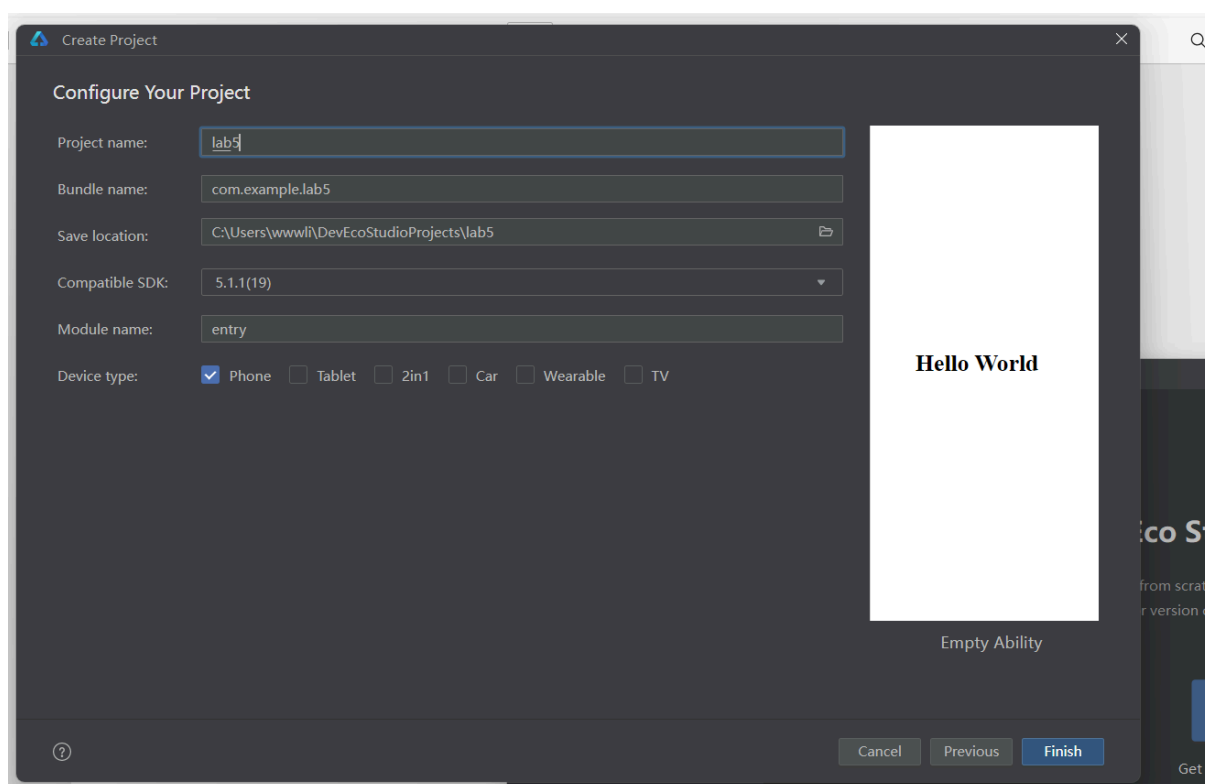
一、实验目标

1、学习使用快速启动模板创建小程序的方法；2、学习不使用模板手动创建小程序的方法。

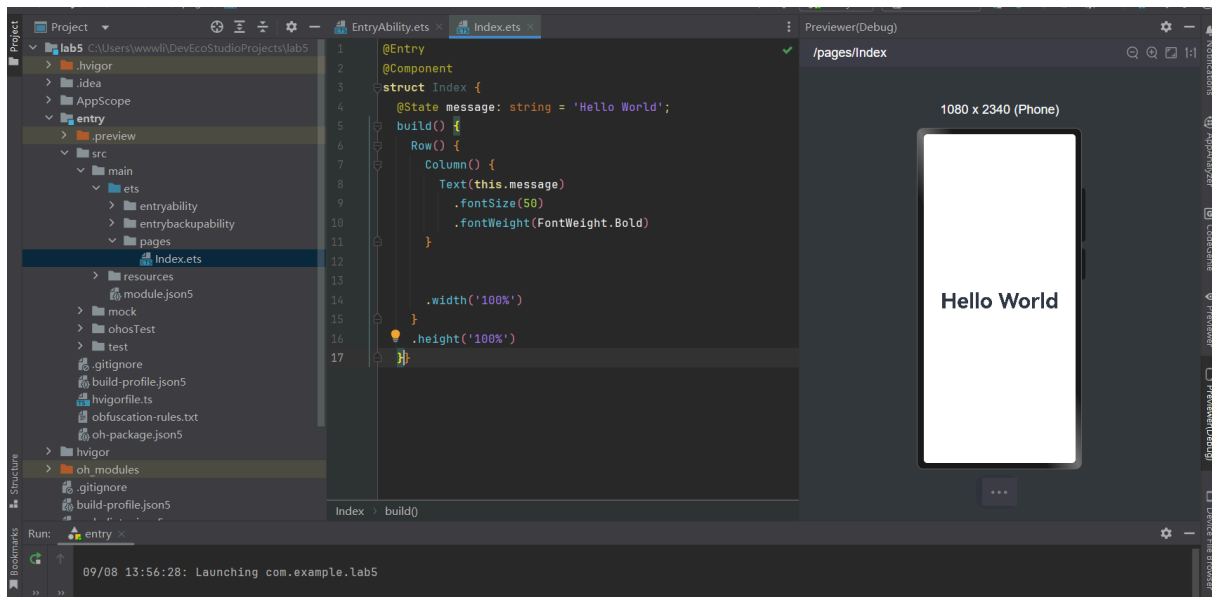
二、实验步骤

列出实验的关键步骤、代码解析、截图。

打开DevEco Studio，单击Create Project创建工程。选择模板Empty Ability，单击Next进行下一步配置，修改项目名称为lab5。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。



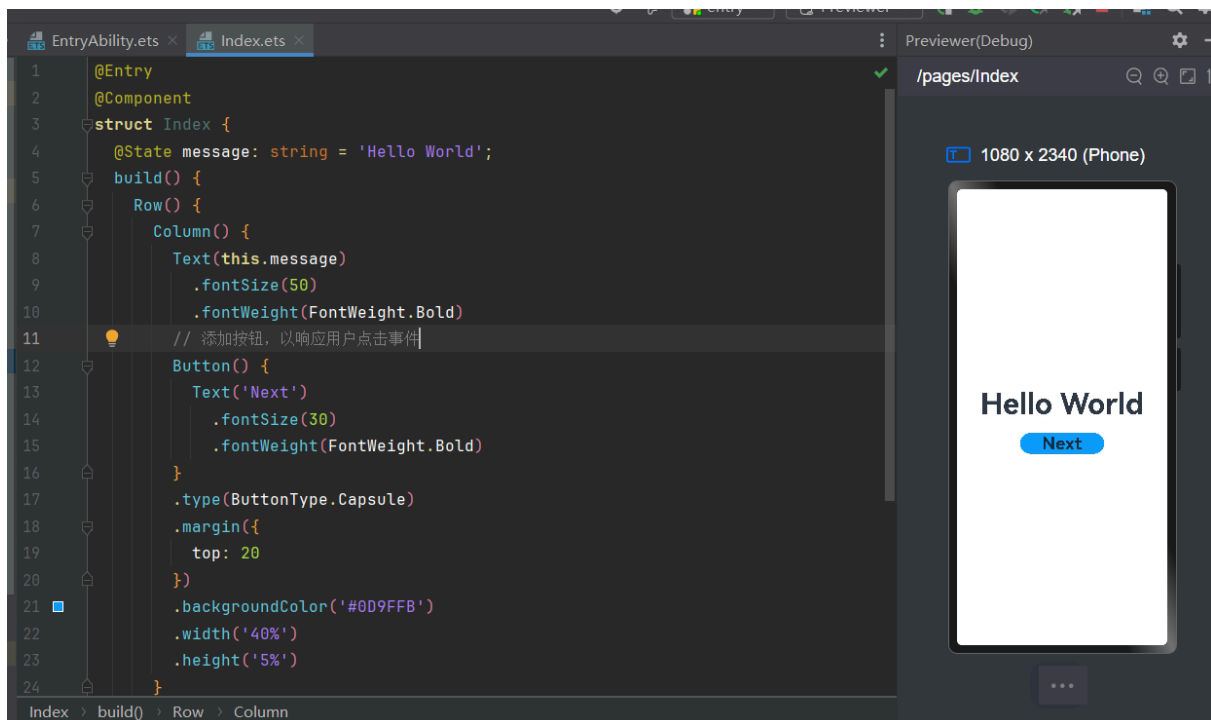
在Project窗口，单击entry > src > main > ets > pages，打开Index.ets文件，将页面RelativeContainer相对布局修改成Row/Column线性布局。代码和实验结果如下：



2.添加按钮

在Index.ets文件中，继续添加代码，添加一个Button组件，作为按钮响应用户的点击操作，从而跳转到另一个界面。代码和实验结果如下：

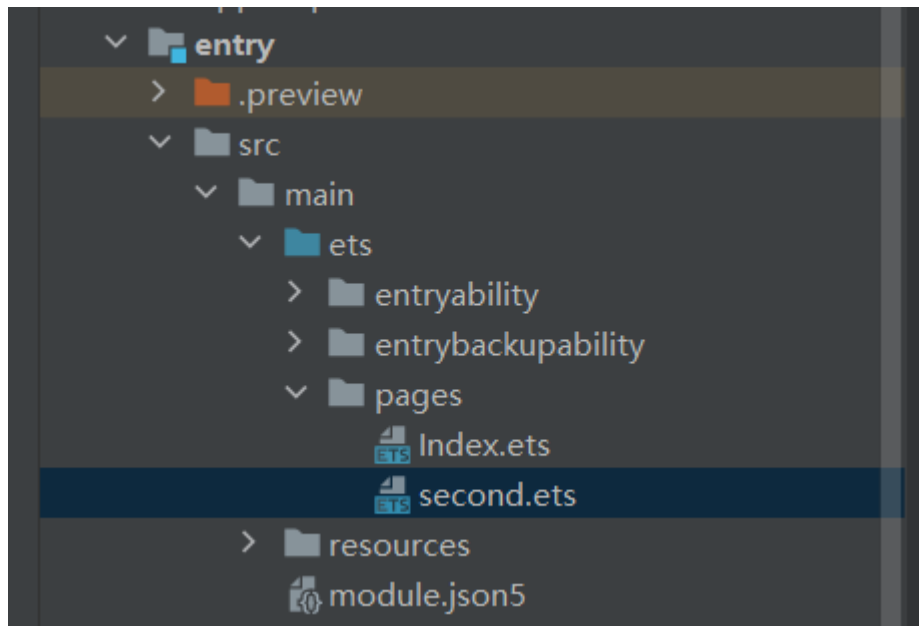
```
Button() {
  Text('Next')
  .fontSize(30)
  .fontWeight(FontWeight.Bold)
}
.type(ButtonType.Capsule)
.margin({
  top: 20
})
.backgroundColor('#0D9FFB')
.width('40%')
.height('5%')
```



3.创建第二个界面

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。并且在entry > src > main > resources > base > profile中main_pages.json文件中的"src"下配置第二个页面的路由"pages/second"。

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



2. 添加文本和按钮

和第一个页面一样，在第二个页面添加Text组件、Button组件等，并设置其样式。代码如下：

```
@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%')
  }
}

```

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Next按钮时跳转到第二页。

```

    // 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}')
      })
    })

```

在第二个页面中，返回按钮绑定onClick事件，单击Back按钮时返回到第一页。

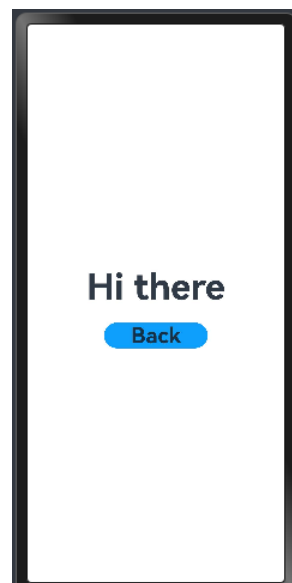
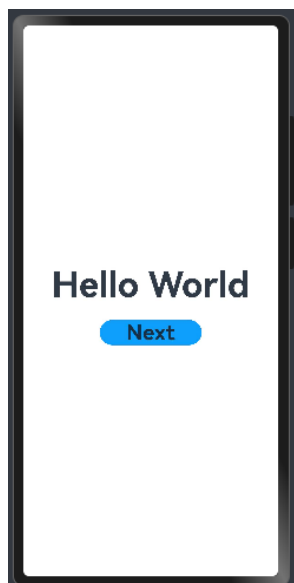
```

    // 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
    })

```

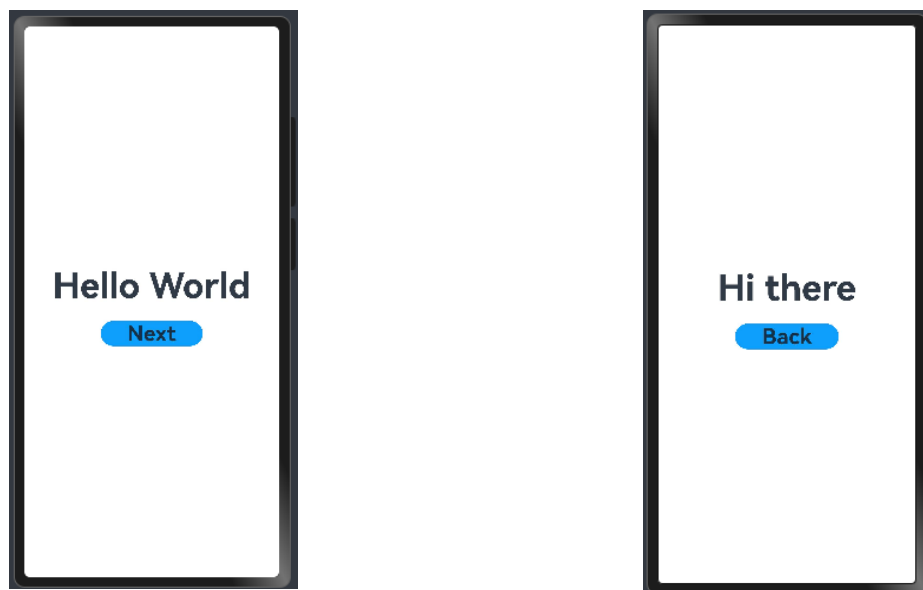
```
let uiContext: UIContext = this.getUIContext();
let router = uiContext.getRouter();
try {
  // 返回第一页
  router.back()
  console.info('Succeeded in returning to the first page.')
} catch (err) {
  let code = (err as BusinessError).code;
  let message = (err as BusinessError).message;
  console.error(`Failed to return to the first page. Code is ${code},
message is ${message}`)
}
})
```

最后当代码书写完全后，最后的结果如下：



三、程序运行结果

列出程序的最终运行结果及截图。



四、问题总结与体会

本问题的实验教程非常的详细，没有遇到任何问题。

本次实验让我深刻体会到 HarmonyOS 应用开发的高效与便捷。ArkTS 语言的声明式范式，结合其 `@Entry`、`@Component`、`@State` 等装饰器，极大地简化了 UI 结构和状态管理，使开发过程更加直观和自动化。HarmonyOS 的 ArkUI 框架拥有灵活的布局方式和丰富的组件，更容易构建出美观的界面，同时，DevEco Studio IDE 提供的代码编辑、实时预览和调试等一体化功能，更加方便和便捷。