

微信小程序开发 教案

专业：计算机应用技术

任课老师：薛晓桂

指导思想：

继续以建设高水平院校为引领，以课证融通为抓手，在教务处、系部的领导下，坚持教学改革、加强内涵建设，以质量求生存，以创新求发展，加强队伍建设，加强教学管理，开展教学改革，为建设高水平院校而努力。

第 1 章 uni-app 框架入门

【学习目标】

- 理解跨平台开发的必要性与 uni-app 的技术定位。
- 掌握 HBuilderX 开发工具的核心功能与项目创建流程。
- 对比原生小程序与框架开发的差异（课程思政：国产开源框架的生态价值）。

【重点难点】

- 重点：uni-app 项目结构解析（pages、static、components）。
- 难点：条件编译语法在多端适配中的应用。

【学习过程】

- 理论讲解（1 学时）
 - 对比原生小程序、混合开发、跨平台框架的技术差异
 - 分析 uni-app"一套代码多端发布"的技术实现原理
 - 介绍 DCloud 生态体系（插件市场、uniCloud、开发者社区）
- 实践操作（2 学时）
 - 安装 HBuilderX 最新版并配置开发环境
 - 创建默认模板项目，解析目录结构与配置文件
 - 实现"Hello World"示例并运行到微信开发者工具

【小结】

通过本章学习，学生理解了跨平台开发的价值与 uni-app 的技术优势，掌握了开发环境搭建与基础项目结构。需注意条件编译语法在处理平台差异时的正确使用，避免多端兼容性问题。

第 2 章 Vue.js 基础与组件化开发

【学习目标】

- 掌握 Vue.js 核心概念（数据响应式、组件化、生命周期）。
- 理解 uni-app 对 Vue 语法的扩展与优化。
- 熟练使用官方组件与自定义组件开发。

【重点难点】

- 重点：组件通信机制（props、\$emit、event bus）。
- 难点：混合开发模式下的组件设计（原生组件与 web 组件协同）。

【学习过程】

- **理论讲解（1 学时）**
 - Vue 响应式原理与数据绑定语法
 - 组件生命周期钩子函数（created、mounted、destroyed）
 - uni-app 组件规范与原生组件差异
- **实践操作（2 学时）**
 - 开发商品列表组件，实现数据循环渲染
 - 创建父子组件通信示例（父传子 props，子传父 \$emit）
 - 使用 event bus 实现非父子组件间通信

【小结】

本章深入学习了 Vue.js 基础与组件化开发模式。学生需重点掌握组件通信机制的适用场景：props 用于单向数据流，\$emit 用于事件触发，event bus 用于复杂场景。注意避免在大型项目中过度使用 event bus 导致代码可维护性下降。

第 3 章 路由与状态管理

【学习目标】

- 掌握 uni-app 路由机制（navigateTo、redirectTo 等）。
- 理解 Vuex/Pinia 状态管理模式，实现全局数据共享。
- 实现复杂业务场景的路由设计（如登录拦截、TabBar 配置）。

【重点难点】

- 重点：动态路由与路由守卫的应用。
- 难点：大型项目的状态分层设计与数据持久化。

【学习过程】

- 理论讲解（1 学时）
 - uni-app 路由配置与导航方式
 - Vuex 核心概念（state、getters、mutations、actions）
 - 路由守卫实现权限控制的原理
- 实践操作（2 学时）
 - 配置包含 TabBar 的多页面路由
 - 实现登录拦截逻辑（未登录用户跳转登录页）
 - 使用 Vuex 管理用户信息与购物车数据

【小结】

本章学习了路由系统与状态管理的核心技术。学生需理解路由守卫在控制页面访问权限中的作用，以及 Vuex 单向数据流的设计理念。在大型项目中应采用模块化的状态管理方案，并结合 localStorage 实现数据持久化。

第 4 章 界面设计与交互实现

【学习目标】

- 掌握 uni-app 布局方案（flex、grid、rpx 适配）。
- 实现常见交互效果（下拉刷新、上拉加载、滚动监听）。
- 理解视觉设计原则，优化小程序 UI 体验。

【重点难点】

- 重点：自定义导航栏与沉浸式设计。
- 难点：复杂动画效果实现（如转场动画、交互动画）。

【学习过程】

- **理论讲解（1 学时）**
 - 移动端响应式设计原则
 - 常见交互模式的用户体验设计
 - uni-app 动画 API 与性能优化
- **实践操作（2 学时）**
 - 使用 flex 布局实现商品展示网格
 - 开发下拉刷新与上拉加载更多功能
 - 实现页面转场动画与元素交互动画

【小结】

本章聚焦于界面设计与交互实现。学生需掌握 flex 布局的核心属性与 rpx 适配方案，理解自定义导航栏的实现原理。在实现动画效果时，应优先使用 transform 与 opacity 属性以提高性能，避免过度复杂的动画影响用户体验。

第 5 章 API 调用与数据交互

【学习目标】

- 掌握 uni-app 统一 API 体系（网络请求、文件操作、设备信息）。

1. 封装 API 请求，实现拦截与错误处理。
2. 理解 WebSocket 实时通信原理与应用场景。

【重点难点】

- 重点：请求并发控制与数据缓存策略。
- 难点：跨域问题处理与接口安全设计。

【学习过程】

1. 理论讲解（1 学时）
 - uni-app API 分类与使用规范
 - HTTP 请求原理与跨域解决方案
 - WebSocket 长连接的适用场景
2. 实践操作（2 学时）
 - 封装统一的 API 请求模块（含拦截器）
 - 实现请求并发控制与 loading 状态管理
 - 开发基于 WebSocket 的实时聊天功能

【小结】

本章学习了 API 调用与数据交互技术。学生需掌握 API 请求封装与错误处理机制，理解跨域问题的解决方案（配置白名单、使用代理服务器）。在处理大量数据请求时，应合理使用缓存策略并控制并发数，避免性能问题。

第 6 章 uniCloud 云开发进阶

【学习目标】

- 深入理解云开发模式（云函数、云数据库、云存储）。
- 实现复杂业务逻辑（如权限控制、事务处理）。

- 掌握云开发性能优化与监控方法。

【重点难点】

- 重点：云函数的安全调用与资源隔离。
- 难点：大数据量的云数据库查询优化。

【学习过程】

1. 理论讲解（1 学时）

- 云开发架构与传统后端开发对比
- 云数据库索引设计与查询优化
- 云函数权限控制与安全策略

2. 实践操作（2 学时）

- 开发带权限验证的云函数
- 设计并优化复杂查询的云数据库索引
- 实现云存储文件的安全访问控制

【小结】

本章深入学习了 uniCloud 云开发技术。学生需理解云函数、云数据库与云存储的协同工作模式，掌握数据库索引优化方法。在处理敏感业务逻辑时，应通过云函数实现权限控制，避免前端直接操作数据库带来的安全风险。

第 7 章 微信小程序特有功能开发

【学习目标】

- 掌握微信小程序开放能力（支付、分享、订阅消息）。
- 实现小程序与微信生态的深度集成（公众号关联、企业微信互通）。
- 理解小程序安全规范与审核机制。

【重点难点】

- 重点：微信支付全流程实现与订单管理。
- 难点：用户隐私保护与敏感数据处理。

【学习过程】

1. 理论讲解（1 学时）

- 微信开放能力 API 分类与使用限制
- 支付流程安全规范与风控措施
- 小程序审核标准与常见拒绝原因

2. 实践操作（2 学时）

- 实现微信支付统一下单与回调处理
- 配置小程序分享功能与自定义分享卡片
- 开发订阅消息推送功能

【小结】

本章聚焦于微信小程序特有功能开发。学生需严格遵守微信支付安全规范，确保支付流程的完整性与数据安全性。在获取用户敏感信息时，应遵循最小必要原则并明确告知用户用途。注意审核标准中的内容规范，避免因违规导致小程序下架。

第 8 章 插件与扩展开发

【学习目标】

- 掌握 uni-app 插件市场的使用与插件开发流程。
- 开发自定义原生插件（Android/iOS）。
- 集成第三方服务（如地图、推送、AI 能力）。

【重点难点】

- 重点：原生插件的生命周期管理与通信机制。
- 难点：多端插件的兼容性设计。

【学习过程】

1. 理论讲解（1 学时）

- uni-app 插件架构与开发规范
- 原生插件与前端的通信原理
- 第三方服务集成的认证与授权机制

2. 实践操作（2 学时）

- 从插件市场集成 uView UI 组件库
- 开发简单的原生插件（如获取设备信息）
- 集成高德地图 SDK 实现位置服务

【小结】

本章学习了插件开发与第三方服务集成技术。学生需掌握插件市场的使用方法，理解原生插件的开发流程与通信机制。在集成第三方服务时，应注意 API 版本兼容性与服务条款合规性，避免因服务变更导致功能异常。

第 9 章 性能优化与质量保障

【学习目标】

- 掌握小程序性能优化方法（分包加载、图片处理、代码压缩）。
- 理解测试流程与自动化测试方案。
- 实现异常监控与错误日志收集。

【重点难点】

- 重点：首屏加载速度优化与长列表渲染优化。
- 难点：端到端测试框架的集成与使用。

【学习过程】

1. 理论讲解 (1 学时)

- 小程序性能瓶颈分析方法
- 自动化测试策略与工具选择
- 错误监控系统的设计与实现

2. 实践操作 (2 学时)

- 配置分包加载并分析加载速度变化
- 使用 Jest 编写单元测试用例
- 集成 Sentry 实现前端错误监控

【小结】

本章学习了小程序性能优化与质量保障技术。学生需掌握性能分析工具的使用，理解分包加载的配置原则。在实施自动化测试时，应根据项目特点选择合适的测试框架与策略。错误监控系统应覆盖前端与云函数，确保问题可追溯。

第 10 章 项目实战与发布上线

【学习目标】

- 综合运用所学知识开发完整商业级小程序（如电商、教育、政务应用）。
- 掌握多环境配置与持续集成/部署（CI/CD）流程。
- 理解小程序运营策略与用户增长方法。

【重点难点】

- 重点：大型项目的架构设计与团队协作规范。

- 难点：灰度发布与线上问题快速修复机制。

【学习过程】

1. 理论讲解（1 学时）

- 商业项目开发流程与敏捷方法论
- 多环境配置（开发/测试/生产）策略
- 小程序推广与用户增长策略

2. 实践操作（2 学时）

- 设计并开发完整的微信小程序项目
- 配置 CI/CD 流程实现自动化部署
- 完成小程序上线前的审核准备

【小结】

本章通过实战项目整合所学知识。学生需掌握大型项目的架构设计原则，理解团队协作规范与版本控制策略。在上线前应进行全面测试，制定应急预案以应对可能出现的线上问题。注意收集用户反馈，为后续迭代优化提供依据。