软件技术专业主要课程

软件技术专业的主要课程、课程的主要内容、教学要求如下表所示：

|  |  |  |
| --- | --- | --- |
| **主要课程** | **主要内容** | **教学要求** |
| Java面向对象程序设计 | 面向对象的核心原则，包括封装、继承和多态；类和对象的创建与使用，构造方法和析构方法；静态和非静态成员，访问控制；继承机制和接口的实现；Java集合框架的应用，包括List、Set、Map等；异常处理，自定义异常；Java I/O（输入/输出）系统，文件操作；多线程编程，线程的生命周期和同步机制；设计模式的基础，如单例、工厂、策略模式等；以及面向对象思想在实际编程中的应用和案例分析。课程旨在培养学生的面向对象分析和设计能力，为高级软件开发打下坚实基础。 | 强调学生必须掌握面向对象的基本概念和原则，能够熟练应用类和对象进行程序设计。学生应理解封装、继承和多态性在实际编码中的应用，能够使用接口和抽象类来实现代码的扩展性和灵活性。课程要求学生熟悉Java集合框架，能够进行异常处理和资源管理。此外，学生还需学习Java I/O操作，掌握多线程编程的基础，理解线程同步和并发控制。课程还包括设计模式的介绍，要求学生能够识别和应用常见的设计模式。最终，学生应能够综合运用所学知识，独立完成面向对象的程序设计任务，并具备良好的编程风格和代码组织能力。 |
| Java Web应用开发 | 涵盖了Servlet技术作为Web应用的核心，包括请求处理和响应生成；JSP页面技术，模板引擎的使用；MVC架构模式在Web开发中的应用；HTTP协议详解，包括请求和响应的生命周期；Web服务器如Tomcat的部署和管理；JDBC数据库访问技术；表达式语言（EL）和JSTL标准标签库的使用；会话管理机制，包括Cookies和HttpSession；Web应用安全性，涉及用户认证和授权；过滤器和监听器的创建与应用；Ajax技术实现异步数据交互；JSON数据格式处理；以及Web应用的性能优化和最佳实践。通过本课程，学生将掌握构建动态和交互式Web应用的必备技能。 | 要求学生深入理解Web应用的工作原理和开发流程。学生需掌握Java SE基础，熟练使用Servlet和JSP进行Web页面的生成与交互处理。要求学生能够运用MVC架构设计Web应用，理解HTTP协议及其在Web开发中的应用。学生应熟悉Tomcat等Web服务器的配置和使用，掌握JDBC进行数据库操作，以及使用JSTL和EL进行页面渲染。课程还要求学生学习Web应用的安全性知识，包括用户认证、授权和数据保护。此外，学生需了解Ajax、JSON与前端技术交互，以及Web应用的性能优化技巧。最终目标是培养学生能够独立开发和部署具有良好用户体验的Web应用。 |
| 基于SSM框架应用系统开发 | 课程涵盖了Spring框架基础，包括IoC容器和AOP原理；Spring MVC框架用于构建Web应用的方法；MyBatis作为数据持久层的操作和配置；SSM框架的整合技巧，实现数据的增删改查；RESTful API设计原则和实现；数据库连接池的使用和管理；Spring Security实现应用安全性；事务管理策略和多数据源配置；前端技术与后端服务的分离；单元测试和集成测试策略；以及项目的部署、性能调优和监控。通过本课程，学生能够掌握使用SSM框架开发企业级应用的完整流程和关键技术。 | 要求学生全面掌握Spring、Spring MVC和MyBatis框架的核心概念与应用。学生需熟练进行SSM框架的整合开发，实现数据的增删改查等基本操作。课程强调学生能够设计RESTful API，理解并应用数据库连接池技术，掌握Spring Security进行应用安全控制。学生应理解事务管理机制，学会使用单元测试和集成测试确保代码质量。此外，课程要求学生了解微服务架构的基本概念，掌握前端与后端分离的开发模式，以及Web应用的部署和性能优化技巧。最终目标是培养学生能够独立开发企业级Web应用系统。 |
| 数据结构与算法（Java） | 数据结构核心概念，线性结构（数组、链表、栈、队列）的实现与操作；树结构（二叉树）及其遍历算法；图的表示（邻接矩阵、邻接表）和遍历（深度优先、广度优先）；排序算法（冒泡、选择、插入、希尔、归并、快速排序）；搜索算法（线性、二分）；递归与分治策略，以及算法的时间和空间复杂度分析。课程通过理论讲解、代码实现和案例分析，培养学生解决实际问题的能力。 | 要求学生深入理解数据结构的基本概念和算法设计原理。学生需掌握数组、链表、栈、队列等线性数据结构，以及树、图等非线性结构的特性和应用。课程要求学生能够使用Java语言实现这些数据结构，并进行有效的数据操作。学生应熟悉递归、分治、贪心、动态规划等算法策略，并能够分析算法的时间和空间复杂度。此外，课程强调学生能够运用所学知识解决实际编程问题，具备良好的逻辑思维和问题解决能力。最终目标是培养学生具备扎实的数据结构和算法基础，为进一步的软件开发和算法设计打下坚实基础。 |
| 数据库应用技术(MySQL) | 数据库系统原理、MySQL架构与安装配置、SQL基础语法、数据定义语言(DDL)、数据操纵语言(DML)、数据控制语言(DCL)、复杂查询技术、索引优化、视图创建与使用、存储过程与函数、触发器的编写与应用、事务管理与隔离级别、锁机制、性能监控与调优、数据备份与恢复策略、安全性控制包括用户权限管理，以及实际数据库设计案例分析。通过本课程，学生将掌握MySQL数据库的管理和应用技能 | 要求学生系统掌握数据库的基本原理和MySQL数据库的高级应用。学生需熟练进行数据库设计、规范化和反规范化，理解ER模型和关系模型。课程要求学生能够使用SQL语言进行复杂查询、数据更新和事务处理。学生应掌握索引、视图、存储过程、触发器等高级功能，以及数据库的备份、恢复和优化技术。此外，课程强调学生学习数据库的安全性管理，包括用户权限控制和数据加密。最终目标是培养学生具备独立设计、管理和优化数据库的能力，以及解决实际数据库问题的专业技能。 |
| Web前端交互技术 | HTML/CSS布局与样式设计，JavaScript基础与高级编程，DOM操作与事件处理，响应式设计与移动优先策略，用户交互设计原则，表单验证与数据绑定，动画与过渡效果实现，以及Ajax和Fetch API进行异步数据交互。 | Web前端交互技术课程要求学生掌握HTML、CSS和JavaScript的基础知识，能够使用这些技术构建响应式和用户友好的网页。学生需学会DOM操作、事件处理、表单验证以及Ajax数据交互。课程强调实践能力，要求学生能够运用前端框架和库（如React或Vue.js）开发交互式应用，并理解前端性能优化和安全性原则。学生应具备良好的代码组织能力和团队协作精神。 |
| JavaScript程序设计 | 该课程涵盖基础语法、数据类型与操作、控制流程、函数定义与作用域、对象与原型链、数组与字符串处理、正则表达式、DOM操作、事件处理、BOM应用、异步编程（Callback、Promise、async/await）、模块化开发、错误与异常处理、性能优化技巧。课程通过实际案例，强化编程思维和问题解决能力，为前端开发打下坚实基础。 | 要求学生深入理解JavaScript语言的核心概念，包括基本语法、数据类型、流程控制、函数、对象和原型链。学生需掌握DOM操作、事件处理机制以及异步编程技术，如Callback、Promise和async/await。课程要求学生能够使用现代JavaScript开发工具和模块化开发方法，理解JSON数据处理和Ajax技术实现数据交互。学生应熟悉前端性能优化和代码调试技巧，能够编写高效、可维护的代码。此外，课程强调实践应用，要求学生能够独立完成具有实际应用价值的JavaScript项目，培养解决复杂问题的能力。 |
| Node应用系统开发 | Node.js环境搭建与基础概念，JavaScript在服务端的应用，异步编程与事件循环机制，Express框架的使用与中间件开发，RESTful API设计原则与实现，数据库集成（如MongoDB、MySQL），身份验证与授权策略（如JWT、OAuth），错误处理与日志记录，性能监控与优化，安全性实践（包括防范XSS、CSRF等攻击），测试（单元测试、集成测试），以及Docker容器化部署。课程通过实际项目案例，培养学生从系统设计到开发部署的全流程能力。。 | 要求学生全面掌握Node.js平台的核心概念和开发技能。学生需熟练使用JavaScript进行服务端编程，理解Node.js的事件驱动和非阻塞I/O模型。课程要求学生能够使用Express框架构建RESTful API，掌握数据库操作和ORM技术，如MongoDB或MySQL。学生应熟悉中间件的使用、身份验证与授权机制，以及错误处理和日志记录。此外，课程强调安全性、性能优化和测试驱动开发的重要性。学生需学习Docker容器化技术和项目部署策略，以及如何编写高效、可维护的Node.js应用。最终目标是培养学生具备独立开发和维护企业级Node.js应用的能力。 |
| Vue应用系统开发 | Vue 3的Composition API的核心概念与应用；组件化开发流程，创建可复用的UI组件；Vue Router 4用于单页面应用的路由管理；Pinia作为Vue 3的官方状态管理库，实现高效的状态组织和维护；Axios库在Vue 3项目中管理HTTP请求；Vite作为开发和构建工具，提高开发效率；TypeScript与Vue 3结合，提升项目类型安全性；单元测试和E2E测试，确保应用稳定性；以及性能优化技巧，包括懒加载、代码分割等。课程通过项目驱动教学，培养学生使用Vue 3和Pinia构建现代Web应用的实战能力。 | 要求学生掌握Vue 3的核心特性，特别是Composition API的使用，以及Pinia作为状态管理工具的应用。学生需熟悉组件化开发流程，能够设计和实现可复用的UI组件。课程要求学生能够使用Vue Router 4进行单页面应用的路由管理，并通过Axios进行API数据交互。学生应了解Vite在现代前端开发中的使用，掌握TypeScript在Vue项目中的集成，以增强代码的可维护性和类型安全性。此外，课程强调编写高质量的单元和集成测试，以及应用性能优化技巧。最终目标是培养学生具备独立开发和维护基于Vue 3和Pinia的前端应用的能力。 |
| UniApp应用系统开发 | UniApp框架的基本概念和跨平台开发的优势；使用Vue.js进行移动应用开发的核心原理；页面布局和组件化开发方法；条件编译和平台特有API的使用；数据绑定和事件处理机制；API调用和网络请求处理；本地存储和数据持久化技术；UI组件库的使用和自定义组件开发；页面路由和导航管理；性能优化和调试技巧；以及利用UniApp开发多端应用的实战项目。课程旨在培养学生掌握UniApp框架，能够高效开发iOS、Android、Web等多平台应用。 | 要求学生掌握跨平台移动应用开发的核心技术。学生需理解UniApp框架和Vue.js的基础知识，能够使用UniApp进行组件化开发和页面布局设计。课程要求学生熟悉条件编译和平台适配，能够调用UniApp提供的各种API，并处理网络请求和数据存储。学生应学会使用UI组件库和自定义组件，掌握页面路由和导航，以及进行应用的性能优化和调试。此外，课程强调实战项目开发，要求学生能够独立完成从设计到发布的全流程，培养解决实际开发问题的能力。 |
| 产品策划与推广 | 市场调研与分析方法，用户需求识别与分析；产品定位与概念开发；产品设计原则与用户体验(UX)设计；产品生命周期管理；品牌建设与品牌推广策略；营销渠道的选择与运用；数字营销与社交媒体策略；内容营销与故事叙述技巧；广告创意与公关活动；数据分析与市场反馈；以及产品推广的法律和伦理考量。课程通过案例研究和实践项目，培养学生的产品策划思维和推广能力，为学生提供在多变市场环境中成功推广产品的实际技能。 | 要求学生具备市场调研和用户分析的能力，能够准确识别目标市场和用户需求。学生需掌握产品定位、概念开发和用户体验设计，了解产品开发全过程。课程要求学生熟悉品牌建设、营销策略和数字营销工具，能够运用社交媒体和内容营销进行有效推广。学生应学会运用数据分析工具评估市场反馈，不断优化产品和推广策略。此外，课程强调创意思维、法律伦理意识和团队协作能力，培养学生成为能够综合运用多学科知识进行产品策划与推广的专业人才。 |
| 软件交互设计项目管理 | 交互设计的基本原则与概念； 界面元素与布局；交互设计的流程和工具；用户研究的方法与技巧；用户角色与场景构建；交互体验优化策略； 信息构架的设计原则；导航与搜索设计；任务流程与交互逻辑；视觉设计的原则与风格；色彩与字体选择；动画与过渡效果；Figma的基本操作；原型设计与交互动画制作；设计与协作工具的使用。 | 该课程旨在让学生掌握软件交互设计的基本原理，和项目管理的方法和技巧。培养学生从用户需求和交互体验出发，进行设计思考与项目管理的能力。对于任课老师来说，需主要讲授界面交互设计的理论基础、用户体验研究方法、设计思维流程、及具体软件的应用和实现。在课程的学习过程中，学生需要掌握的典型工作任务包括，对用户调研与需求分析、使用Figma实现界面交互设计和在线协同操作。 |
| 产品原型交互设计 | 设计思维与创新方法，用户研究与人物画像创建，用户需求分析与场景设计；交互设计原则，如一致性、反馈、错误防范等；原型设计工具的使用，如Sketch、Adobe XD、Figma等；低保真与高保真原型的创建，包括线框图、流程图和交互细节；用户测试方法，如启发式评估、可用性测试；设计迭代与优化策略；以及跨学科团队协作。课程通过实际案例分析和项目实践，培养学生的原型设计能力，以及将用户中心设计理念应用于产品开发全过程的技能。 | 强调学生对用户中心设计理念的深入理解与应用。学生需掌握设计思维、用户研究方法，能够创建人物画像和场景分析。课程要求学生熟练使用原型设计工具进行低保真到高保真的原型制作，理解并运用交互设计原则。学生应能够组织和执行用户测试，收集反馈，并根据反馈进行设计迭代。此外，课程强调沟通和协作能力，要求学生能够在跨学科团队中有效交流设计思路和解决方案，最终培养出能够独立完成从概念到原型的交互设计的综合能力。 |
| 用户界面设计与制作 | 视觉设计基础，色彩理论、排版和网格系统；UI设计原则，如一致性、可用性和美学；设计软件工具的使用，如Adobe Photoshop、Illustrator、Sketch等；图标和图形元素的设计；响应式和自适应设计技术；用户交互流程和动效设计；设计规范和系统；用户测试和反馈整合；以及跨平台设计策略，包括Web和移动界面设计。课程通过案例研究、实践项目和设计工作坊，培养学生的创意思维和专业UI设计能力。 | 要求学生具备良好的审美能力和创意思维。学生需熟练掌握UI设计原则和视觉设计基础，包括色彩搭配、排版和布局。课程要求学生能够使用专业设计软件进行图标、图形和界面元素的创作。学生应理解响应式设计的重要性，并学会为不同设备和屏幕尺寸设计界面。此外，课程强调用户测试的重要性，要求学生能够根据用户反馈优化设计。学生还需学习设计规范的制定和维护，以及如何在团队中协作，确保设计的一致性和可交付性。最终目标是培养学生能够独立完成高质量的用户界面设计项目。 |
| 低代码平台技术 | 低代码开发的概念与优势，低代码平台的架构与组件；可视化编程基础，拖放式界面构建技术；业务逻辑与流程自动化设计；集成API和第三方服务的方法；数据模型与数据库设计简化；用户身份验证与授权机制；前端与后端的分离以及低代码应用的部署；低代码平台的安全考虑；以及通过实际案例教授如何使用低代码工具快速开发企业级应用。课程旨在培养学生利用低代码技术提高开发效率，解决实际业务问题的能力。 | 要求学生理解低代码开发的核心理念，掌握使用低代码平台进行快速应用开发的技巧。学生需熟悉平台的架构组件和可视化编程方法，能够设计和实现业务逻辑与自动化流程。课程要求学生学会集成API、处理数据模型以及配置用户身份验证。学生应了解前端与后端的分离概念，掌握应用的部署和管理。此外，课程强调安全最佳实践和团队协作能力，培养学生评估和选择适合的低代码工具，以及运用这些工具解决复杂业务问题的能力。 |
| 企业项目实战 | 本课程是本专业的专创融合课程，涉及之前所学所有专业课程中的专业知识和技能进行梳理、总结、提升和综合应用。综合应用内容主要包括：行业企业工程项目、学科技能竞赛项目、国家大学生创新创业训练计划项目、教师科研项目等。 | 要求学生能够将理论知识应用于实际工作场景中。学生需掌握项目管理的全过程，包括项目策划、执行、监控和收尾。课程要求学生在团队环境中有效沟通和协作，解决跨学科问题。学生应熟悉敏捷开发流程和工具，能够进行技术选型和应用。此外，课程强调风险管理、质量保证和客户关系管理的重要性。学生需学会如何进行项目成果的展示和评估，以及如何收集和整合用户反馈进行产品迭代。最终目标是培养学生具备在企业环境中独立管理和执行项目的能力。 |
| 网页设计与制作 | 网页设计基础，如网页布局和视觉层次；HTML语言的语法结构，用于创建网页内容；CSS样式表的使用，实现网页的视觉表现和响应式设计；图像和多媒体元素的嵌入技术；网页导航设计和用户界面元素的使用；网站的可访问性和可用性原则；网站的版本控制和团队协作流程；以及网站性能优化技巧。课程通过项目实践，使学生能够独立完成静态网页的规划、设计和制作。 | 强调学生对网页设计和前端开发基础的掌握。学生需熟练应用HTML5来构建网页结构，使用CSS3实现网页的视觉设计和响应式布局。课程要求学生理解并运用网页设计原则，包括色彩搭配、排版和用户界面设计。学生应学会使用JavaScript或jQuery添加基本交互功能，并关注网站的可访问性和跨浏览器兼容性。此外，课程强调网站性能优化的重要性，要求学生掌握图片处理、代码压缩和SEO基础。学生还需学会使用版本控制工具，培养团队协作和项目管理能力。最终目标是使学生能够独立完成专业、美观且用户友好的静态网页设计和制作。 |

如有调整，以最新为准。