**《高级程序设计》教学大纲**

**编写人：赵晓茹**

1. **为什么学习本门课程：**

**学习的关键是培养学生综合的职业能力，为企业培养高素质的技能型人才。在日常教学，尤其是专业课程教学中，教师在教授学生专业知识的同时，要着力提高学生的职业素养和综合能力。情境教学以能诱发学生学习主动性，强化学生学习感受性，学习过程贯穿实践性，关注学生发展的原则，成为提高学生职业素养的有效途径。**

似乎每天都有新的JavaScript库或框架发布，有能力从众多的库或框架中进行筛选是件好事，一个应用程序中如果包含了太多的脚本，对于维护来说却是件坏事。随着应用程序中脚本数量的增加，脚本间会产生依赖关系，当我第一次听说AngularJS时，它就立刻引起了我的注意，因为它只通过一个独立的框架就可以构建动态、交互密集型的客户端应用。通过进一步的研究，我确信这个第一判断是正确的，于是开始迷上了这个框架。AngularJS提供了一系列健壮的功能，以及将代码隔离成模块的方法，这对提高可复用性、可维护性和可测试性都是非常有益的。它的核心功能包括DOM操作、动画、模板、双向数据绑定、路由、历史管理、Ajax和测试，等等。

1. **学习本门课程的方法**

**本课程在教学过程中，将采用任务驱动配合案例教学，在解决困难以后，提出“任务需求—分析任务思路—引出解决问题”的方法，层层深入，从语法到案例，最后引导学生自主完成任务。**

1. **学习本门课程的方法**

新一轮基础教育课程改革,不仅表现在课程目标、结构、内容等方面的变化, 尤其集中地表现在对学生学会学习的倡导.本文以历史上教育家对培养学生掌握学习方法的重视以及这一思想的时代意义为切入点,针对当前教育——无论是理论层 面,还是实践层面——对培养学生掌握学习方法极度忽视的现实,提出和论述了将学习方法作为一门课程在实施的现实意义,并进一步论述学习方法这门课程的实施与评价的可能方式。

1. **教学目标**
2. **知识目标（学什么知识）**
3. 掌握git及gitHub的使用
4. 掌握构建工具gulp的使用
5. 掌握Angular.js的表达式、模型
6. 掌握Angular.js的作用域
7. 掌握Angular.js的控制器与过滤器
8. 掌握Angular.js关于表单的操作
9. 掌握Angular.js的服务
10. 掌握Angular.js的依赖注入
11. 掌握Angular.js的路由
12. 掌握ionic的使用
13. **能力目标（产出物是什么）**
14. 使用Angular.js结合bootstrap实现一套后台管理系统的开发，能够进行增删改查的操作
15. 使用Angular.js结合ionic实现一套webApp的开发
16. 使用gulp作为构建工具
17. 将项目提交到gitHub上进行管理
18. **参考教材**

**《AngularJs权威教程》[美] 勒纳（Ari Lerner） 著；赵望野，徐飞，何鹏飞 译**

**人民邮电出版社出版**

1. **课程内容安排**
2. **gitHup的使用**
   1. git的简介与安装
      1. 什么是git
      2. 如何安装git
   2. 远程仓库
      1. 如何添加远程仓库
      2. 如何从远程仓库克隆

1.2.3 SSH协议与HTTPS协议

1. **gitHub关于分支**

2.1 分支管理

2.1.1 如何创建分支

2.1.2 如何合并分支

2.1.3 分支管理策略

2.2 多人协作

2.2.1 如何解决代码冲突

1. **gulp的使用**
   1. 关于gulp

3.1.1 什么是gulp

3.1.2 gulp的安装

* 1. gulp语法
     1. task()
     2. pipe()
     3. src()
  2. 构建任务
     1. 拷贝文件
     2. 合并文件
     3. 压缩文件
     4. 编译SASS
     5. 压缩CSS
     6. 启动服务
     7. 监测文件

1. gulp的其他应用
   1. 模块化
      1. gulp实现js模块化
   2. mock数据
      1. mock数据的配置
   3. 版本控制

4.3.1 生成MD5文件

4.3.2 让html文件自动将入口文件的文件名替换为md5加密之后的名称

1. **Angular的基本指令**
   1. 常用指令的使用
      1. ng-app
      2. ng-bind
      3. ng-model
      4. ng-click
      5. ng-class
      6. ng-if
      7. ng-show
      8. ng-hide
      9. ng-repeat
      10. ng-init

5.2 angular表达式

5.3 控制器

5.3.1 如何创建控制器

5.3.2 $scope

5.3.3 $rootScope

1. Deferred对象与过滤器
   1. 关于deferred对象

6.1.1 什么是deferred对象

6.1.2 如何创建一个deferred对象

* 1. Deferred对象的常用方法
     1. when()
     2. then()
     3. done()
     4. fail()
     5. promise()
     6. resolve()
     7. reject()

6.3 angular的过滤器

6.3.1 currency

6.3.2 uppercase

6.3.3 lowercase

6.3.4 date

6.3.5 limitTo

6.3.7 filter

1. orderBy排序与自定义指令
   1. orderBy
      1. 实例讲解orderBy的各种排序方式
      2. 实例讲解升序降序
   2. 自定义指令
      1. 实例讲解如何使用filter自定义指令
2. **Angular服务**
   1. 关于服务
      1. 什么是服务
      2. 为什么要使用服务
   2. Angular的常用服务
      1. $http
      2. $timeout
      3. $interval

8.3 自定义服务

8.3.1 如何使用service自定义服务

8.3.2 如何注入服务

1. **依赖注入**
   1. 什么是依赖注入
   2. Angular的依赖注入机制
      1. value
      2. factory
      3. service
      4. provider
      5. constant
   3. 总结

9.3.1 各种依赖注入机制之间的区别

1. **angular的路由**
   1. 关于路由
      1. 什么是路由
      2. Angular.js的路由机制
   2. 路由设置对象
      1. $routeProvider对象
      2. when()方法
   3. when()方法的参数配置
      1. template
      2. templateUrl
      3. controller
      4. controllerAs
      5. redirectTo
      6. Resolve

10.4 传递参数

10.4.1 如何在路由跳转时传递参数

10.4.2 如何获取传递过来的参数

1. **ui-router**
   1. 关于ui-router
      1. 什么是ui-router
      2. 如何使用ui-router
   2. ui-router的state服务
      1. 如何配置无参数的路由
      2. 如何配置有参数的路由
      3. 如何实现路由的嵌套
   3. 总结
      1. ui-router与ng-route的区别
2. **路由事件与$location服务**
   1. 路由事件

12.1.1 $stateChangeError

12.1.2 $stateChangeStart

12.1.3 $stateChangeSuccess

12.1.4 $stateNotFound

12.2 路由的其他方法与属性

12.2.1 go()

12.2.2 reload()

12.2.3 get()

12.2.4 current

12.2.5 params

12.3 $location

12.3.1 url()

12.3.2 search()

12.4 内置事件

12.4.1 $locationChangeStart

12.4.2 $locationChangeSuccess

1. **使用Angular.js结合bootStrap实现后台管理系统**
   1. 使用bootstrap实现页面布局
   2. 使用路由实现页面跳转
   3. 功能模块的分类

13.2.1 信息添加

13.2.2 信息修改

13.2.3 信息删除

13.2.4 信息搜索

1. **使用Angular.js实现在线翻译**
   1. Angular与表单
   2. 封装一个ajax请求的服务
   3. 控制器别名
   4. 表单提交
2. **ionic的安装与使用**
   1. ionic的安装

15.1.1 cordova的安装

15.1.2 ionic的安装

15.1.3 使用ionic初始化项目

* 1. css-components

15.2.1 header

15.2.2 footer

15.2.3 content

15.2.4 buttons

15.2.5 tabs

15.2.6 list

15.2.7 forms

1. **ionic的js组件一**
   1. content
      1. ion-content
      2. ion-refresher
      3. Ion-pane
   2. scroll
      1. ion-scroll
      2. ion-infinite-scroll
      3. $ionicScrollDelegate
   3. Tabs
      1. ion-tabs
      2. ion-tab

16.3.3 $ionicTabsDelegate

* 1. Slide Menus
     1. ion-slide-menus
     2. ion-slide-menu-content

16.4.3 ion-slide-menu

16.4.4 menu-toggle

16.4.5 $ionicSideMenuDelegate

1. **ionic的js组件二**
   1. Navigation

17.1.1 ion-nav-view

17.1.2 ion-view

17.1.3 ion-nav-bar

17.1.4 ion-nav-buttons

* 1. Lists

17.2.1 ion-list

17.2.2 ion-item

17.2.3 ion-delete-button

* 1. Form inputs

17.3.1 ion-checkbox

17.2.2 ion-radio

17.2.3 ion-toggle

* 1. Slide box

17.4.1 ion-slide-box

17.4.2 $ionicSlideBoxDelegate

* 1. Action sheet

17.5.1 $ionicActionSheet

* 1. Popup

17.5.1 $ionicPopup

* 1. Loading

17.5.1 $ionicLoading

1. **webApp开发（上）**
   1. 使用SASS编译样式
   2. 使用HTML5+CSS3搭建页面结构
   3. 使用ionic-font实现字体图标
   4. 解决一像素bug
2. **webApp开发（中）**
   1. 使用angular.js实现路由配置、页面跳转
   2. 使用angular.js实现http请求、数据渲染
   3. 使用angular.js实现服务的定义
3. **webApp开发（下）**
   1. 使用ionic的scroll实现区域滚动
   2. 使用ionic的tabs实现选项卡
   3. 使用ionic的actionSheet实现操作表
   4. 使用ionic的popup实现弹出窗口
   5. 使用ionic的loading实现加载动画