# 亿路编程技术培训企划书

**目录**

1. 客户需求分析-----------------------------------------------------------2
2. 不同APP类型横向对比---------------------------------------------3
3. 不同开发语言横向对比-----------------------------------------------4
4. 课程方案-----------------------------------------------------------------5
5. 附录-----------------------------------------------------------------------6
   1. 课程大纲
      1. Web 静态网站------------------------------------------------6
      2. Web App------------------------------------------------------7，8
      3. 原生安卓App--------------------------------------------------9
      4. 原生IOS App--------------------------------------------------10
      5. 混合型App--------------------------------------------------11，12
      6. 小程序-----------------------------------------------------------13
      7. 桌面应用-----------------------------------------------------14，15

### 需求分析

**目标客户：**棚瀬さん（たなせ)

**年龄：**40~50

**职业：**

**需求：**想了解开发APP的相关技术

**学习动机：**与本社有业务往来，产生了自己开发APP的想法

**学习所需的工具与材料：**一台电脑，语言相关的基础知识书籍

### 不同APP类型横向对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **优点** | **缺点** | **开发语言** |
| **Web 应用** | **Web APP**  **（例：www.jd.com）** | 1. 支持范围广，多设备兼容， 2. 开发成本低，周期短，可以即时上线 3. 用户通过浏览器访问，开发者更新仅需要通过服务器更新即可 4. 有一定数量的成熟开发框架 | 1. 和原生App相比，性能和体验相对较差 2. 不能访问手机的所有功能 3. 没有网络的情况下不能使用 4. 使用新型浏览器的时候容易出现兼容问题 | 浏览器端：CSS，HTML5，JavaScript…  服务器端：Python，Java，MySQL，sqlite… |
| **手机应用** | **原生APP**  **（例：IOS Calendar）** | 1. 拥有手机的各种权限 2. 运行速度快，用户体验在这三者中最好 3. 可以快捷调用设备的接口 4. 在没有网络的情况下依旧可以访问以前下载过的数据 5. 官方提供大量开发工具帮助支持开发 | 1. 开发成本高，周期长 2. 无法跨平台部署，安卓和IOS都需要单独开发，而且支持的设备有限 3. APP每次更新需要向每个商城提交申请，审核周期长 | 安卓：Java，Kotlin，C++，C#，JavaScript…  IOS：Objective-c，swift |
| **混合APP**  **（例：微信）** | 1. 支持范围广，多设备兼容 2. 部分内容不需要联网也能获取 3. 可访问的手机功能比Web App多，但比不上原生App | 1. 性能和速度依旧比不上原生App 2. 技术相对不够成熟 | JavaScript，CSS，HTML，Web View |
| **小程序**  **（例：健康码）** | 1. 开发成本低廉，过程相对简单，维护简易 2. 相比H5，功能与应用场景丰富，相比Web APP可以调用更多手机功能 3. 支持范围广，可以跨平台部署 4. 前端代码托管在微信服务器上，渲染加载速度更快 | 1. 客户群体数量受限于微信用户数量 2. 微信小程序有整包不大于8M，单包不大于2M的大小限制 3. 程序内容受腾讯审核限制 4. 大多数情况下仅仅是作为一个APP部分功能的整体嫁接 | 服务器端：HTML5，CSS，JavaScript  服务器端：JavaScript |
| **桌面应用** | **桌面应用**  **（例：桌面微信）** | 1. 可实现的功能非常多  2. 部分应用下载后不需要联网 | 1. 有比较长的开发周期和较高的开发难度 | Java，Python…… |

### 不同语言的横向对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **语言** | **学习周期** | **难度** | **优点** | **缺点** |
| ***HTML + CSS***  人气：5 | 6天 | **1** | 1. 很高的兼容性，可以调用很多设备功能  2．同时支持安卓和IOS  3. 追加了本地数据库，支持开发离线应用  4. 出色的动画，游戏动态效果 | 1. 并不是所有浏览器都能完美支持HTML5，IE9以下均不支持  2. 网站表现很大程度上取决于网络情况 |
| ***Javascript***  人气：4 | 20天 | **2** | 1. 难度相对较低，容易上手  2. 支持跨平台 | 1. 安全性较差（在此场景下并不构成一个主要缺点） |
| ***Python***  人气：5 | 10天 | **1** | 1. 非常的简单，容易上手 2. 有大量的库来辅助开发 | 1. 有相当数量的外库，掌握这些库需要很多时间和精力 |
| ***Java***  人气：5 | 20天 | **3** | 1. 开发群体基数大而且活跃，遇到问题也能在网络平台找到解决方案 | 1. 学习曲线较长，不是最容易上手的语言，但也不是最难学的 |
| ***Objective-c***  人气：4 | 25天 | **4** | 1. 适合UI变成  2. 运行速度相对较快  3. 方便与C/C++混用 | 1. 仅用于开发苹果原生APP  2. 学习曲线相对陡峭 |
| ***Swift***  ***人气：4*** | 25天 | **4** | 1. 语法简洁  2. 性能出色，报错精准 | 1. 第三方库支持较少  2. 仅用于开发苹果原生APP |
| ***C++***  人气：3 | 30天 | **5** | 1. 跨平台性非常好  2. 运行效率高  3. C++在C的基础上增加OOP特点 | 1. 对比Java缺少垃圾回收机制，容易内存泄露  2. 学习曲线陡峭，学习比较难 |

注：以上表格中的**入门**，**全部掌握**，**精通**分别对标以下技术能力

**入门：**能够跟着教程顺利地创建一个完整的可运行项目

**掌握全部内容：**具备独立开发相对较复杂项目的能力

**精通：**对该语言相关知识融会贯通，能够开发维护体量庞大的项目

客户的目标能力暂时定位在入门与掌握全部内容之间

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **方案** | **目标** | **学习内容** | **课程所需** | **所需课时** | **费用** |
| **WEB** | **静态网站开发** | 一个静态网站 | CSS3，HTML，JavaScrip，Nginx | * + - 1. 一台能够运行相应配置环境的电脑       2. 所调用的API的对应开发者账户       3. 对应开发语言的相关文档 | 待定 | 待定 |
| **Web App** | 一个简单的ToDoList应用 | HTML, JavaScript, CSS  Java，sqlite，J-Query，Boot-Strap | 待定 | 待定 |
| **手机APP** | **原生App** | 一个能在客户手机上运行的轻量APP（ToDoList） | Java，Android SDK，  API调用方法  IOS：Swift (OC) | 待定 | 待定 |
| **混合型App** | 一个ToDo List应用 | JavaScript，VUE，Uni-App | 待定 | 待定 |
| **小程序** | 一个能够部署在微信上的小程序 | JavaScript，HTML5，CSS，WXSS，WXML | 待定 | 待定 |
| **桌面应用** | **桌面应用** | 一个能够在windows系统运行的增删改查（本地保存） | Java + Swing；  WPF + C-sharp | 待定 | 待定 |

### 课程方案

**备注：课时，一节课一小时，一天上限两节课**

### 附录

**课程大纲（暂定）**

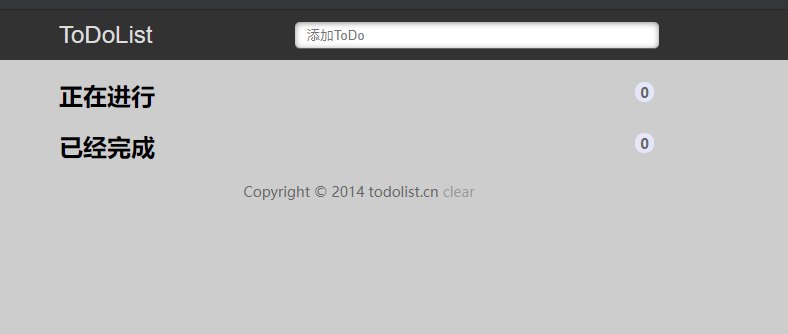
|  |  |
| --- | --- |
|  | **课程安排** |
| **WEB静态网站** | **HTML + CSS** |
| HTML简介以及基础标签 |
| HTML高级标签 （上下） |
| CSS引入，基础选择器与选择器权重 |
| CSS复杂选择器，权重计算问题，CSS基础属性 |
| 实例：居中五环，两栏布局，BFC |
| 包裹浮动元素，开发实例，文字溢出处理，背景图处理 |
| CSS要点总结说明，项目演练（此课时可选） |
| **JavaScript** |
| 变量，值运算，运算符） |
| 条件语句，循环语句 |
| 条件语句补充，初识引用值 |
| typeof，类型转换 |
| 函数与作用域 （上下） |
| 递归与预编译 （上下） |
| 对象与包装类 （上下） |
| 原型，原型链，call/apply （上中下） |
| 继承模式，对象枚举，命名空间（上下） |
| 三目运算符 |
| 项目开发 |
| Arguments，克隆，数组 |
| try...catch，es5标准模式 |
| dom选择器，节点类型 |
| dom继承树，基本操作（上下） |
| date对象，定时器 |
| 获取窗口属性，dom尺寸，脚本化CSS（上下） |
| 事件event，事件监听（上中下） |
| Nginx |
| 学习服务器与网站的关系，学习配置Nginx服务器，实际部署网站 |
| **网站实际设计与开发** |
| 网页UI的视觉分镜设计，布局与设计 |
| 导航设计，导航与网站的内容结构 |
| 网站的可访问性 |

**http://www.rococo.co.jp**

|  |  |
| --- | --- |
|  | **课程安排** |
| **WEB应用** | **HTML + CSS** |
| HTML简介以及基础标签 |
| HTML高级标签 （上下） |
| CSS引入，基础选择器与选择器权重 |
| CSS复杂选择器，权重计算问题，CSS基础属性 |
| 实例：居中五环，两栏布局，BFC |
| 包裹浮动元素，开发实例，文字溢出处理，背景图处理 |
| CSS要点总结说明，项目演练（此课时可选） |
| **JavaScript** |
| 变量，值运算，运算符） |
| 条件语句，循环语句 |
| 条件语句补充，初识引用值 |
| typeof，类型转换 |
| 函数与作用域 （上下） |
| 递归与预编译 （上下） |
| 对象与包装类 （上下） |
| 原型，原型链，call/apply （上中下） |
| 继承模式，对象枚举，命名空间（上下） |
| 三目运算符 |
| Arguments，克隆，数组 |
| try...catch，es5标准模式 |
| dom选择器，节点类型 |
| dom继承树，基本操作（上下） |
| date对象，定时器 |
| 获取窗口属性，dom尺寸，脚本化CSS（上下） |
| 事件event，事件监听（上中下） |
| **Java** |
| IDEA的下载，安装与使用，Hello World |
| 基本语法与数据类型 |
| 基本运算与运算符 |
| 顺序结构，选择结构，循环结构（if，switch，while，do-while，for） |
| 方法与数组（附带最简单的冒泡排序） |
| 类与对象 |
| 匿名对象与构造器 |
| 封装 |
| 继承思想 |
| 多态 |
| 代码块与final关键字 |
| 抽象类和接口 |
| 泛型 |
| Map的基本操作与常用方法（增删，遍历） |
| 异常的抛出与处理，自定义异常 |
| 多线程与并发处理 |
| **sqlite** |
| 关系型数据库结构，sqlite基本查询语句，JDBC连接数据库 |
| **J-Query** |
| jQuery基础，选择器，属性与样式 |
| jQuery筛选，DOM操作 |
| jQuery事件，动画（淡出入，滑动，自定义动画） |
| jQuery Ajax |
| **Boot-strap** |
| 框架介绍与应用 |
|  |

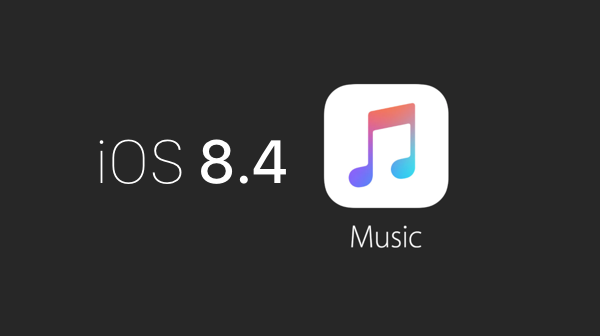
**http://www.todolist.cn**

**在线ToDoList**

****

|  |  |
| --- | --- |
|  | **课程安排** |
| **原生安卓APP开发** | **Java** |
| IDEA的下载，安装与使用，Hello World |
| 基本语法与数据类型 |
| 基本运算与运算符 |
| 顺序结构，选择结构，循环结构（if，switch，while，do-while，for） |
| 方法与数组（附带最简单的冒泡排序） |
| 类与对象 |
| 匿名对象与构造器 |
| 封装 |
| 继承思想 |
| 多态 |
| 代码块与final关键字 |
| 抽象类和接口 |
| 泛型 |
| Map的基本操作与常用方法（增删，遍历） |
| 异常的抛出与处理，自定义异常 |
| 多线程与并发处理 |
| **Android SDK** |
| 开发环境的配置，认识Android工程目录 |
| 基本布局的使用方法 |
| 基本控件的使用方法 |
| Activity和Fragment的生命周期 |
| 列表类控件和Adapter的使用方法 |
| Activity的启动模式，四大基本组件的作用及使用场景 |
| Android自带的SharePreferences存储 |
| 文件存储方法 |
| 了解HTTP相关协议（如果需要的话） |
| 三种动画的使用 |
| 自定义view |
| 了解API，如何调用API |
| 项目开发 |
|  |

|  |  |
| --- | --- |
|  | **课程安排** |
| **原生IOS APP开发** | **Swift** |
| IOS开发所需系统与环境 |
| playground文件，常量，变量与数据类型 |
| 函数和元组的声明和使用 |
| 条件语句，循环语句和控制转移语句 |
| 可选类型的含义，声明和使用场景 |
| 枚举、类和结构体的声明和使用 |
| 集合、数组和字典的声明与使用 |
| Xcode创建项目，模拟器运行，编辑器中编辑应用，使用控件 |
| 设置控件属性，标记关联控件，IOS的事件驱动机制 |
| Outlet Collection管理多个控件，运用数组和for循环语句实现应用功能 |
| 搭建视图：Assets.xcassets管理应用图片、UIImageView控件 |
| 学习通过segue实现界面跳转 |
| 自动布局与约束，APP ICON |
| 应用显示名称与真机调试 |

**Apple Music**

|  |  |
| --- | --- |
|  | **课程安排** |
| **混合APP** | **JavaScript** |
| 变量，值运算，运算符），条件语句，循环语句 |
| typeof，类型转换 |
| 函数与作用域 （上下） |
| 递归与预编译 （上下） |
| 对象与包装类 （上下） |
| 原型，原型链，call/apply （上中下） |
| 继承模式，对象枚举，命名空间（上下） |
| 三目运算符 |
| Arguments，克隆，数组 |
| try...catch，es5标准模式 |
| dom选择器，节点类型 |
| dom继承树，基本操作（上下） |
| date对象，定时器 |
| 获取窗口属性，dom尺寸，脚本化CSS（上下） |
| 事件event，事件监听（上中下） |
| **Vue** |
| 什么是Vue，Vue的常用指令 |
| Vue的data，作用域，双向绑定、事件、事件对象获取方式 |
| Vue冒泡、阻止默认事件、键盘事件 |
| Vue.JS的模板、属性、class和style |
| Vue的计算属性和侦听属性 |
| Vue的过滤器详解，Vue的实例方法及cloak，Vue的自定义指令 |
| Vue的组件封装，组件模板使用，父子级数据交互 |
| Vue组件阴影效果及高级选项卡的组件封装（实战案例） |
| 什么叫路由，vue-router的下载与使用，router的理解及搭建 |
| sub-Routers的理解与使用 |
| 介绍Vue脚手架，vue-cli的下载与使用， |
| simple、webpack、webpack-simple脚手架介绍与应用 |
| 简易组件在脚手架中的使用以及封装 |
| **Uni-App** |
| Uni-App的环境搭建，创建项目运行，了解目录 |
| 全局外观配置，创建新页面和新页面的配置 |
| tabbar，condition启动模式配置 |
| text，view，image组件的基本使用 |
| 数据绑定，注册事件和传参以及获取到事件对象，生命周期函数 |
| 组件之间的通讯方式，uni-ui组件库的基本介绍和使用 |

**Tik Tok**



|  |  |
| --- | --- |
|  | **课程安排** |
| **小程序** | **HTML + CSS** |
| HTML简介以及基础标签 |
| HTML高级标签 （上下） |
| CSS引入，基础选择器与选择器权重 |
| CSS复杂选择器，权重计算问题，CSS基础属性 |
| 实例：居中五环，两栏布局，BFC |
| 包裹浮动元素，开发实例，文字溢出处理，背景图处理 |
| CSS要点总结说明，项目演练（此课时可选） |
| **JavaScript** |
| 变量，值运算，运算符） |
| 条件语句，循环语句 |
| 条件语句补充，初识引用值 |
| typeof，类型转换 |
| 函数与作用域 （上下） |
| 递归与预编译 （上下） |
| 对象与包装类 （上下） |
| 原型，原型链，call/apply （上中下） |
| 继承模式，对象枚举，命名空间（上下） |
| 三目运算符 |
| Arguments，克隆，数组 |
| try...catch，es5标准模式 |
| dom选择器，节点类型 |
| dom继承树，基本操作（上下） |
| date对象，定时器 |
| 获取窗口属性，dom尺寸，脚本化CSS（上下） |
| 事件event，事件监听（上中下） |

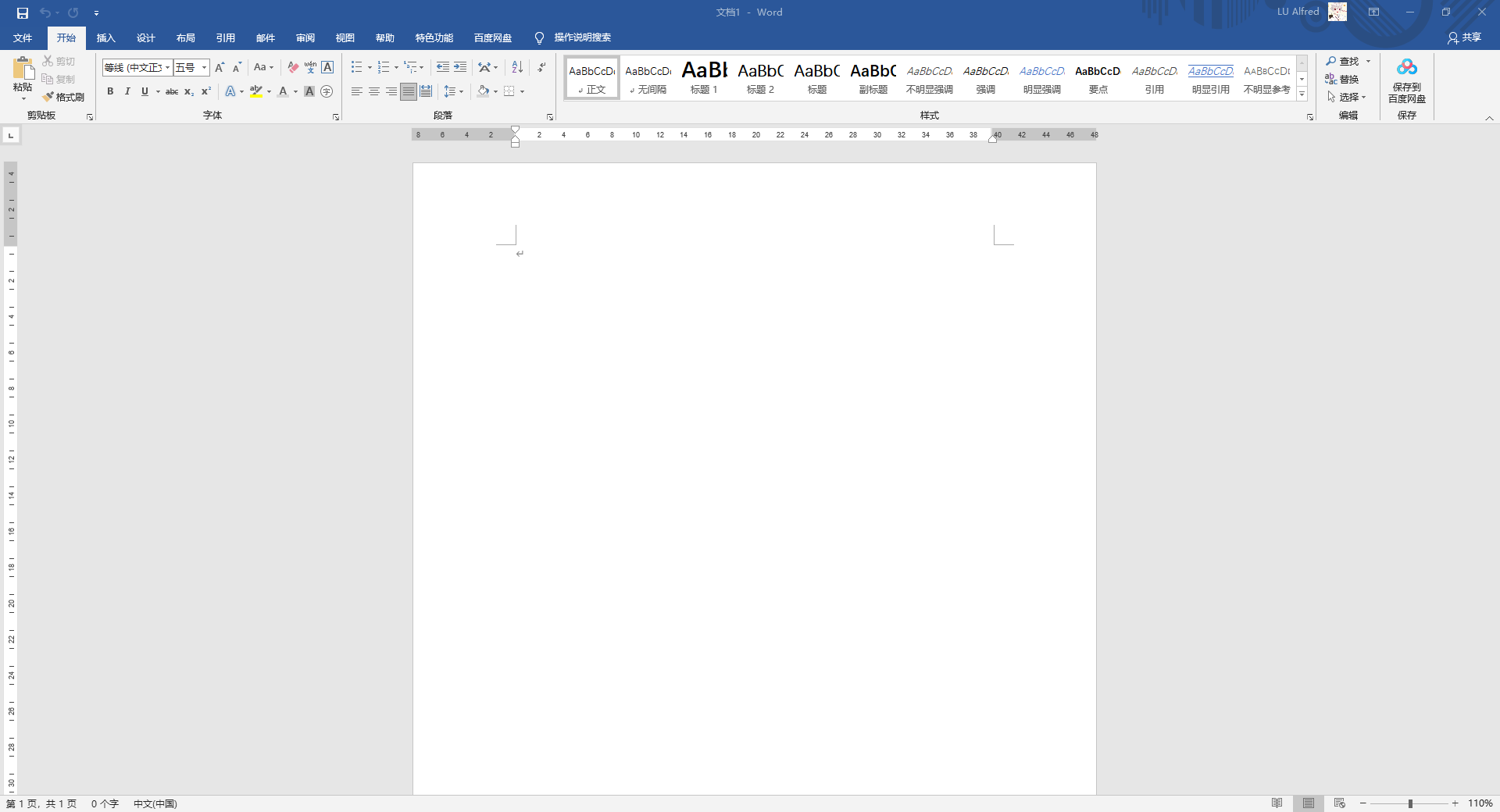
****

**初声日语**

|  |  |
| --- | --- |
|  | **课程安排** |
| **桌面应用（Java + Swing）** | **Java** |
| IDEA的下载，安装与使用，Hello World |
| 基本语法与数据类型 |
| 基本运算与运算符 |
| 顺序结构，选择结构，循环结构（if，switch，while，do-while，for） |
| 方法与数组（附带最简单的冒泡排序） |
| 类与对象 |
| 匿名对象与构造器 |
| 封装 |
| 继承思想 |
| 多态 |
| 代码块与final关键字 |
| 抽象类和接口 |
| 泛型 |
| Map的基本操作与常用方法（增删，遍历） |
| 异常的抛出与处理，自定义异常 |
| 多线程与并发处理 |
| **Swing** |
| 基本控件，布局管理器的使用 |
| 自定义布局器，图片的加载与显示 |
| 菜单与工具栏，自定义绘制，JList列表框和自定义显示 |
| JTable表格与自定义显示，Java Swing的应用程序打包和发布 |
|  |

|  |  |
| --- | --- |
|  | **课程安排** |
| **桌面应用（C# + WPF）** | **C#** |
| Visual Studio的安装（下用VS代替Visual Studio）与使用 |
| C#的语法结构（变量与常量，运算符与表达式，流程控制语句） |
| OOP类和对象（类和对象概念，具体实现，成员方法与构造方法） |
| OOP类和对象（值类型和引用类型，静态属性和方法，方法重载） |
| 结构体，C#的封装和索引器 |
| C#中的事件处理 |
| 面向对象的继承与多态 |
| 抽象类和接口 |
| C#中的常用类 |
| 集合与泛型 |
| 调试与异常处理 |
| **WPF** |
| WPF中的常用布局 |
| 熟悉常用控件的使用 |
| 学习创建模板，自定义控件 |
| 绑定界面元素，绑定方法·和模式以及绑定原理 |

**Microsoft Office Word**

****

**JavaScript**（<https://ke.qq.com/course/231577?tuin=31cf0205&taid=2841395744442521>）

**CSS + HTML**（<https://study.163.com/course/courseMain.htm?courseId=1004165002>）

**Java**（<https://study.163.com/course/courseMain.htm?courseId=1005537028>）

**Vue**（<https://study.163.com/course/introduction.htm?share=1&shareId=1450550286&courseId=1004979009&_trace_c_p_k2_=6830c33a4c3a4a12a95fabf6d4e039ed>）

**C#**（<https://study.163.com/course/courseMain.htm?courseId=1209746822>）

**WPF**（<https://blog.csdn.net/Pentoncos/article/details/106017722>）

**Swift**（<https://study.163.com/course/introduction/1006380014.htm>

<https://study.163.com/course/introduction.htm?courseId=1006531042>）