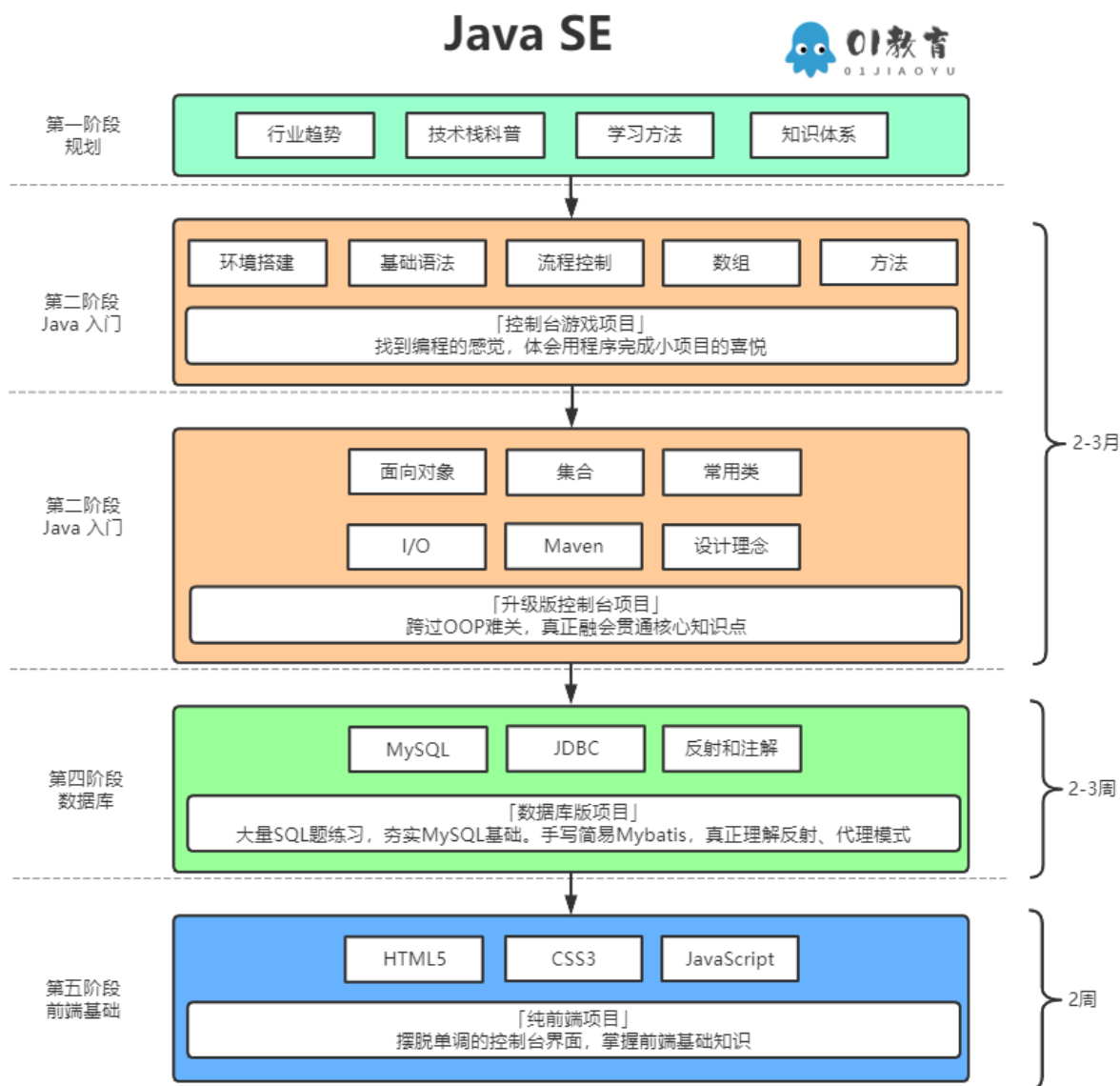


学习时长仅供参考

## Java SE



## 科普、规划

了解软件开发/互联网行业，科普各个技术栈，建立学习心态。

初学者最痛苦的就是没有方向，不知道学什么、学了不知道能干什么。

所以在正式学习前要学的知识有一个大概的了解（本路线会介绍的），**要知道每项技术是干什么的、学了有什么用、为什么要学**，这样才能在学习时不会两眼一抹黑。

## 第二阶段：Java 入门

### 环境搭建

[重点] JDK、Idea 安装与配置

### 基础语法

[重点] 变量、常量、注释

[重点] 数据类型

[熟悉] 运算符

[熟悉] 控制台输入、输出：println、Scanner

### 流程控制

[熟悉] 流程图

[重点] 分支：if、switch、逻辑短路

[重点] 循环：while、for、foreach、do...while

### 数组

[重点] 数组的创建和使用

[熟悉 难点] 冒泡排序

[熟悉 难点] 选择排序

[熟悉 难点] 二分查找法

[熟悉] 二维数组

### 方法

[重点] 方法创建、调用

[重点] 形参、实参

[熟悉 难点] 递归

### 「重点/难点讲解」

所有编程语言必然有其特性和基本语法，这是后续所有知识点的基石。

该阶段的 **两个排序和一个查找法**，属于基本的算法。

**程序就是对数据进行增删改查操作**，算法是操作数据、解决问题的一种方式，**程序 = 算法 + 数据结构**，所以算法会一直伴随着我们的开发生涯。

对于新手来说，目前的算法可能刚理解起来有点吃力，代码会写不出来。不过没关系，**先理解其算法的思想**（就是说这个算法是干什么的、有什么好处）就行了。

还有 **递归**，目前也只需要理解其作用（自己调用自己）即可，在某些情况下，使用递归会极大减少我们的代码量，后续随着我们的编码越来越多，你也会对递归越来越有感觉。

## 「基础控制台项目」

游戏面板等有趣的小项目。

用控制台进行简单的交互，实现增删改查等功能，将该阶段所有知识全部用上。

比如：

```
===欢迎进入零壹星球===
请输入数字选择相应操作：
1. 购买物品
2. 出售物品
3. 退出游戏

===商店===
以下是所有商品，请输入对应编码进行购买：
1. 小红药品
2. 小蓝药品
3. 木棍
4. 新手戒指
...
...

===背包===
以下是背包所有物品，请输入对应编码进行购买：
1. 小红药品
2. 小蓝药品
3. 空
4. 空
4. 空
```

该阶段最重要的是让自己找到编程的感觉，体会到用程序来完成功能的喜悦。

## 第三阶段：Java 核心

### 面向对象

**[重点]** 成员变量、成员方法、创建对象和使用对象

**[重点]** 封装、继承、多态

**[熟悉]** 静态变量、静态方法

**[重点]** 接口与抽象类

**[熟悉]** 内部类、匿名内部类

**[重点]** Lambda 表达式

唯一微信号：ddg\_1987

## 集合

[重点] List、Set、Map

[熟悉] 泛型

[熟悉] 迭代器

[重点 难点] 基本的数据结构，如数组、链表、Hash，等等

[重点] Stream

[熟悉] 相关工具类，如 Collections、Arrays 等

## 常用类 and 知识

[重点] String、StringBuilder、StringBuffer、StringJoiner

[熟悉] Math

[熟悉] Optional

[熟悉] LocalDateTime

[熟悉] 枚举

[重点] 日志

[重点] 异常

## I/O

[熟悉] File 相关操作

[熟悉] 序列化

[重点] 各种流的使用

## Maven

[熟悉] Maven 基本概念：项目管理和构建介绍、Maven 项目结构

[重点] 安装和配置（本地仓库配置，镜像配置）

[重点] 依赖管理

[重点] 构建流程

[熟悉] 模块管理（聚合项目）

## 设计理念

[重点] 面向对象设计原则：如开闭原则、单一职责原则、依赖倒置原则，等等

[重点] 三层架构

[重点] MVC

## 「重点/难点讲解」

该阶段是 Java 语言的核心特性、核心知识，和第一阶段一样，是后续所有知识的基石。

**面向对象** 这一块是初学者的第一大难关，如果感到学习困难是很正常的，不要气馁，只需要多练习项目就好了。**一定要多敲多练项目。**

然后就是 **数据结构** 这一块，数据结构就是组织数据的一种方式，合理的数据结构能极大提高程序性能。

前面也说了，程序就是算法 + 数据结构，这一块也会一直伴随着我们的编程生涯。

对于初学者来说，数据结构也是比较难以理解的，没关系，老套路，**现阶段只需了解一些基本数据结构的特性**（就是这个数据结构的特点是什么、什么场景使用最合理）即可。

## 「升级版控制台项目」

项目技术升级：比如数组改为集合、面向过程改为面向对象、内存保存改为序列化本地保存。

项目难度升级：增加业务复杂度，比如电影选座系统。

该阶段由于涉及到许多不得不讲解的核心知识，所以难度陡增，是初学者第一大难关。

为了降学习难度，所以去除了更晦涩难懂的的多线程、网络编程等技术，**只讲解在目前项目中会实际用到的技术。**

该阶段项目需要不断打磨、不断优化，以真正掌握这些知识点，而不是学到后面又忘了前面。

## 第四阶段：数据库

### MySQL

[了解] 数据库概念：为什么要有数据库、关系型数据库的概念、数据库的种类

[重点] 环境安装与配置

[熟悉] 数据定义语言（DDL）

[熟悉] 数据操作语言（DML）

[重点 难点] 数据查询语言（DQL）

[重点] 索引与 explain

**疯狂练习：各种 SQL 题（网上有50道SQL题）**

### JDBC

[熟悉] 驱动与基本语法

[熟悉] Statement、PreparedStatement

[熟悉] 实体与数据映射、Lombok

[熟悉] 事务

[熟悉] DAO 封装

## 反射和注解

[重点 难点] 反射

[重点] 注解

[重点 难点] 静态代理、动态代理

[重点 难点] 手写简易 MyBatis

### 「重点/难点讲解」

数据库最常用的操作就是查询，所以 **数据查询语言（DQL）** 毫无疑问是数据库的核心知识。这一块说难是因为知识点比较多，但这也是最容易解决的了。**无他，疯狂练习 SQL 题目即可。**

然后就是 Java 著名的难点知识 **反射和代理** 了，为什么难呢，因为它和面向对象理念是相悖的，所以让众多学习者折戟沉沙。

**反射是动态代理的基础，而动态代理则用来扩展对象的功能。**所以，这一块知识需要我们手动去扩展对象的功能，将其用在实际项目中才能更好地理解。即，用该只是封装、扩展 JDBC 原生功能，自己实现一个简易的 MyBatis 框架出来。

### 「数据库版控制台项目」

将之前持久化方案改成数据库存储，数据复杂度升级，手写分页。

数据库是另一种语言，和 Java 区别较明显，需要大量的练习来巩固知识。

把反射等知识放在这里是因为反射和注解在这刚好有用武之地，如果不实际使用，只干讲解技术，学员很快就会忘记。

该阶段直接手写简易的 MyBatis，锻炼封装能力的同时，真正掌握反射等难懂的知识。

## 第五阶段：前端基础

### HTML5

[熟悉] 各种常用标签，如图像、块、内联、列表、表格、表单，等等

### CSS3

[熟悉] 基础样式，如文本、字体，等等

[熟悉] 选择器

[熟悉] 常用布局模型

[了解] 常用样式库，如Bootstrap

### JavaScript

[熟悉] 基本语法

[熟悉] DOM

[熟悉] JQuery

[重点] Ajax 和JSON

## 「难点讲解」

界面是前端，Java 写的程序是服务器程序，称为后端。

作为后端程序员来说，也是要了解一些前端知识的，不过不用了解太多，只要大概知道那么回事就行了。

其中 Ajax 和 JSON 要着重了解，因为这是前后端分离、前后端传递数据的关键。

## 「纯前端项目」

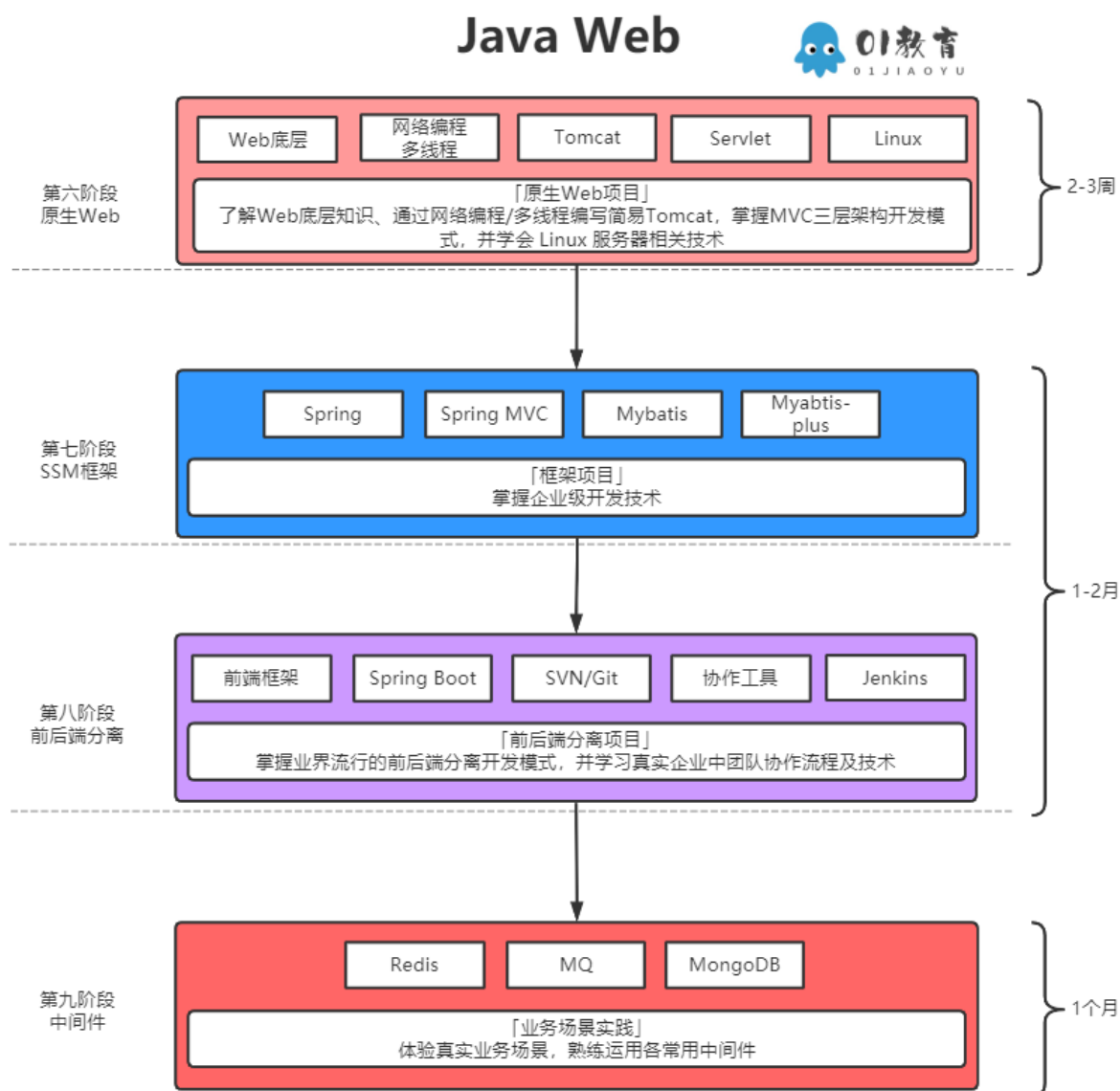
将之前控制台项目显示改为前端实现，只实现页面，无需实现后台交互。

我们重点是后端开发，所以前端只需有基本的掌握即可，无需花费太多时间。

表单数据传递、Ajax等，只用请求本地静态资源即可，后台交互留在后面 Web 项目中再掌握。

Java 有图形界面技术，但完全不用学，浪费时间。

# Java Web



## 第六阶段：原生 Web

## Web 底层

[熟悉] URL、HTTP

[熟悉] IP、DNS

[熟悉] 请求与响应模型

[熟悉] 服务器概念

## 网络编程和多线程

[了解 难点] 网络编程 Socket

[熟悉 难点] 多线程

[重点 难点] 手写一个简易 Web 服务器

## Tomcat

[重点] 安装与基本配置

[熟悉] 项目目录体系

[熟悉] 搭建静态页面服务器

## Servlet

[重点] Request、Response

[重点] Session、Cookie

[了解] JSP

[熟悉] 文件上传和下载

[重点] 转发与重定向、MVC 三层架构

[重点] Filter、Listener

[重点 难点] 手写简易 Spring MVC

## Linux

[了解] Linux 概念：Linux 发展、各发行版本

[重点] 虚拟机安装和使用

[熟悉] XFtp、XShell

[重点] 目录结构和常用命令

[重点] JDK、Tomcat、MySQL 安装

[重点] WAR 包部署

## 「重点/难点讲解」

**网络编程和多线程** 是晦涩难懂的知识点，**新手千千万万不要执着这一块**，不用觉得学不会而自闭。

网络编程属于 Java 中网络通讯最底层的 API，在实际开发中我们几乎不会用到，因为别人已经封装好了更高层、更易懂的 API 供我们使用。

唯一微信号：ddg\_1987



从初级工程师到高级工程师，多线程是 Java 程序员永恒的难点，对于初级开发者来说在实际开发中也极少用到。

正是因为实际开发中很少用到，所以许多初学者对于这一块感觉难以理解，不知道学了能干嘛。那么可以自己实现一个简易的 Web 服务器来加深这一块的理解。在实现过程中会碰到许多问题，不过没关系，也不用将功能实现得多完美，只需体会到知识点能实际运用起来即可。

然后就是 `Servlet`，这个是 Java Web 的基础和核心，后续的 Web 框架就是基于此而实现的。为了加深这一块的理解，我们可以手动实现 Spring MVC 框架。

最后是 `Linux` 这一块，我们后续写的程序基本都是部署在 Linux 上，所以作为一个后端程序员来说，要学会服务器的使用和部署。

## 「原生 WEB 项目」

将之前项目完全升级为 Web 项目。

该阶段开始使用 Maven 搭建项目，算是企业级开发的第一步，主要掌握动态网页、前后端传递的感觉，并且在该阶段要学习 Linux 知识，项目也要部署到 Linux 服务器上。在后续阶段会不断练习 Linux 相关技术。

该阶段为过度阶段，所以不用太多纠结 前端、JSP 相关知识。

## 第七阶段：SSM框架

### Spring

[重点] 理解控制反转和依赖注入

[重点] XML、注解、Java 三种配置方法

[熟悉] Servlet 和 Spring 整合

[重点] 面向切面编程、AspectJ

### Spring MVC

[熟悉] 基本使用方式（控制器、接口编写）

[重点] 常用注解

[了解] Thymeleaf

[重点] Spring 和 Spring MVC 整合

[重点] 拦截器

### MyBatis

[熟悉] 基础使用（增删改查）

[重点] Spring 整合，SSM 整合

[重点] 动态 SQL

[熟悉] 级联映射、缓存

[熟悉] PageHelper 插件

## MyBatis-Plus

[熟悉] 基础使用（增删改查）

[熟悉] 分页插件

[熟悉] 逻辑删除

[熟悉] 字段自动填充

[重点] 条件查询器

[重点] Lambda 查询

### 「重点/难点讲解」

框架，是方便我们开发的工具，可以提高我们的开发效率、系统稳定性。在企业级开发中，是不可能通过原生 API 开发的，一定会使用框架进行开发。

Spring 有两层意思，一个是 Spring 框架本身（提供控制反转功能，方便我们组织代码），一个是 Spring 生态体系（一系列的框架和库）。

那对于 Java 开发者来说，**Spring 是重中之重，毫不夸张地说，我们一切的企业级开发都会基于 Spring 框架！**所以要不断精进这一块的知识。

Spring MVC 就是 Spring 体系中的一员，用于方便 Web 开发。

MyBatis 是简化了 JDBC 的框架，MyBatis-Plus 则是对 MyBatis 功能增强的一个框架。

**要熟悉掌握这些框架，不断地练习项目即可。**

### 「框架项目」

将之前 Web 应用全面升级为 SSM 框架，JSP 页面全部改为 Thymeleaf。

其中前端页面许多数据渲染可以使用 JSON，提前熟悉前后端分离。

---

该阶段正式进入企业级开发模式，可以和其他人一起合作进行团队开发。

## 第八阶段：前后端分离

### 前端框架

[了解] React、Vue 二者选一

[了解] 各个前端界面框架：如 ElementUI、Ant 等

[了解] Node.js

### Spring Boot

[重点] 基本使用，整合 SSM

[重点] 常用注解

[重点] 自定义配置

[重点] Jar 打包与部署

[熟悉] RESTful

唯一微信号：ddg\_1987

[重点] 认证/授权详解、JWT

[熟悉] Spring Security

## 团队协作

[熟悉] SVN

[重点] Git

[重点] 项目管理流程和工具，如 TAPD、Ones，等等

[熟悉] Jenkins

## 「重点/难点讲解」

该阶段学起来反而会觉得轻松简单，因为 Spring Boot 就是简化我们开发的工具，它能让各个框架的使用变得极其轻松。

所以在该阶段，**重点就是锻炼自己的业务能力**，通过框架完成各种各样的功能，并且了解一些项目管理的工具。

Git 是我们程序员协作的重点，一定要好好熟练掌握。

## 「前后端分离项目」

将之前项目升级为前后端分离项目，即将之前的服务端渲染页面全部改为前端页面，前后端只准使用 JSON 进行数据传递。

该阶段项目学习完毕便可以完全胜任初级开发工程师岗位，重点是掌握现在流行的前后端分离开发模式，以及在企业中真实的开发流程。

学完该阶段我们应当有极强的自学能力了，即在编程行业可以自驱动成长和学习，碰到问题都能自行了解和解决。

## 第九阶段：中间件

### Redis

[重点] 环境安装与配置

[重点] 五大数据类型和基本操作

[熟悉] Java API

[重点] 应用场景

[重点 难点] 缓存穿透、击穿、雪崩、一致性问题

### MQ

Rabbit、Rocket、Kafka 三选一

[重点] 环境安装与配置

[熟悉] Java API

[重点] 应用场景

[重点 难点] 消息丢失、重复消费、高可用问题

唯一微信号：ddg\_1987

## MongoDB

[重点] 环境安装与配置

[重点] 数据类型和基本操作

[熟悉] Java API

[重点] 应用场景

### 「重点/难点讲解」

中间件就是用来解决具体某个需求场景的技术，所以熟悉中间件的唯一办法，就是实践各种业务场景。

Redis 是缓存数据库，用来提高数据响应速度。

MQ 是消息队列，用于异步、削峰、解耦，可以解决许多业务场景问题。

MongoDB 是文档数据库，方便检索数据。

其中 Redis 基本已经成为现在企业开发中的必会技能，和 MySQL 一样重要，所以应当好好钻研。

其他中间件，**不一定要全部学完，碰到某个需求了再学习具体技术会更好。**

该阶段标注的难点，都是属于中间件的核心问题，也是面试常问的问题，需要不断熟悉。

### 「业务场景实践项目」

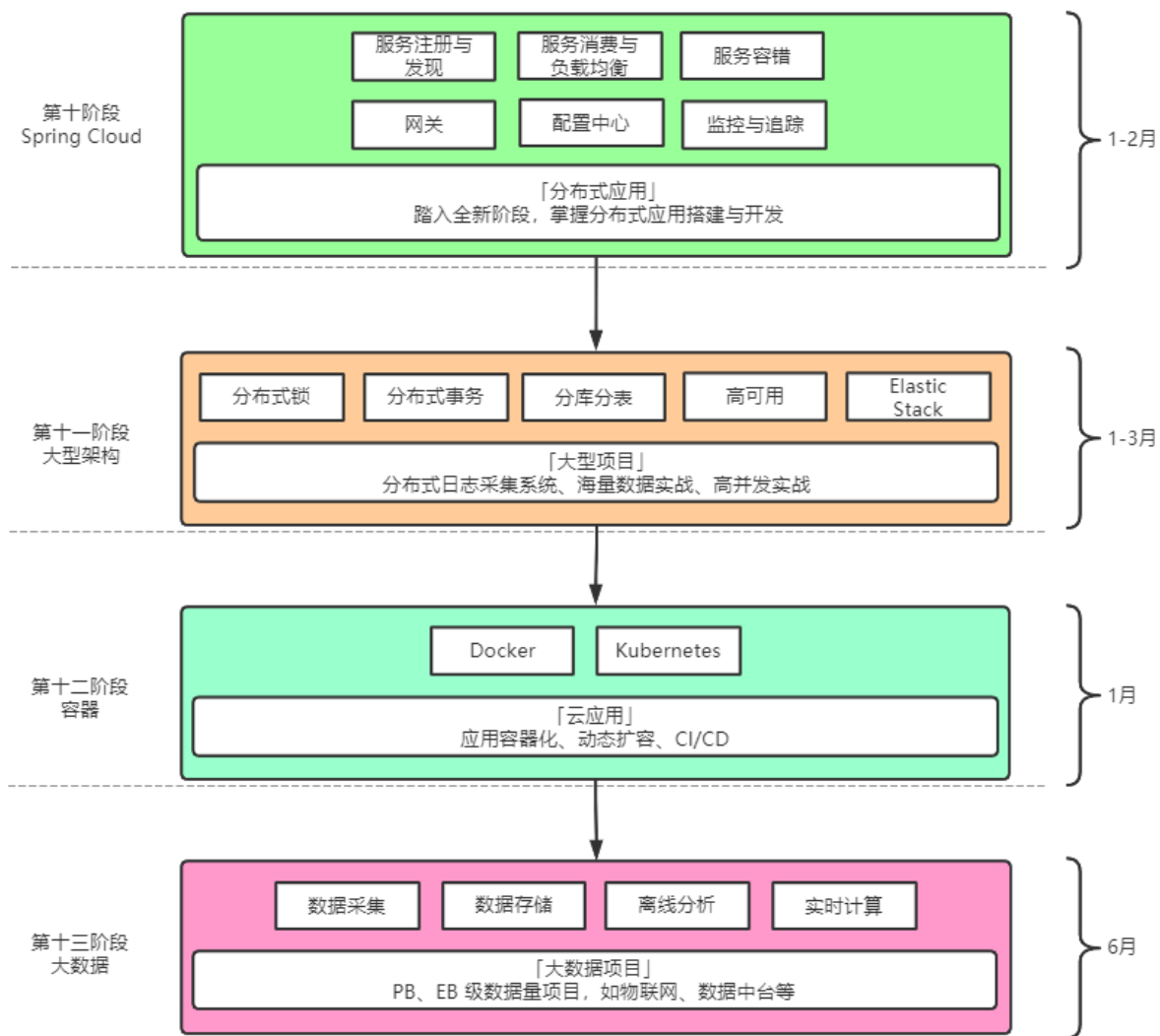
将各个中间件可以用到的应用场景，全部升级到之前的项目里

这三个中间件在单体应用中比较常见，所以在分布式之前学习。

## 分布式微服务架构

---

## 分布式微服务架构



## 第十阶段：Spring Cloud

### 服务注册与发现

[重点] Zookeeper

[重点] Nacos

### 服务消费与负载均衡

[重点] Dubbo RPC、Dubbo LB

[熟悉] RestTemplate、OpenFeign、Ribbon

### 服务容错

[了解] 服务降级、熔断、限流概念

[重点] Sentinel

## 网关

[了解] 网关概念

[重点] Gateway、Zuul 二选一重点了解

[熟悉] Nginx

## 配置中心

[了解] 配置中心概念

[重点] Zookeeper

[重点] Nacos

## 监控与追踪

[了解] 监控与追踪的重要性

[重点] SkyWalking

[重点] Arthas

[熟悉] Zipkin

## 「重点/难点讲解」

分布式是一个全新的阶段，难度陡增。**如果没有熟练开发单体应用，不要贸然学习该阶段知识。**

为什么要使用这么难的系统呢，因为随着系统数据量、访问量增大，传统的单体应用已经承载不了压力了，所以需要通过分布式来解决这些问题。就好像，一个人打不赢，我就喊上一堆人。

分布式大概就这些核心组成部分，每个部分有多个技术实现，重要的是了解其背后的思想，技术都是万变不离其宗的。

## 「分布式应用」

将之前项目各个模块拆分为微服务。

之前都是单体应用，许多理念并不能适用分布式，所以该阶段是让自己对分布式入门，一些非常难的点（如：分布式事务）放到后面学习。

## 第十一阶段：大型架构

### 分布式锁和事务

[熟悉] Redis 分布式锁

[熟悉] Zookeeper 分布式锁

[重点 难点] 2PC 和 3PC（分布式事务）

### 分库分表

[重点] 主从、读写分离

[熟悉] 分库分表概念（垂直、水平）

[重点] Sharding-JDBC

唯一微信号：ddg\_1987

## 高可用

[重点] Redis 集群高可用

[重点] MQ 集群高可用

[重点] Zookeeper 集群高可用

[熟悉 难点] 负责均衡集群高可用（LVS、F5）

## Elasticsearch

[熟悉] 核心概念：搜索引擎、反向索引

[重点] ES 安装和配置

[重点] ES 常用API

[熟悉] Kibana

[熟悉] Logstash

[熟悉] Filebeat

[熟悉] 分词器

[重点] 数据迁移与同步

[重点 难点] 集群搭建

### 「重点/难点讲解」

该阶段难点在于分布式系统本身（分布式事务、高可用等），不光需要大量实践，也需要坚实的底层知识（比如网络、分布式理论等）。

所以对于该阶段的学习，在练习项目的同时，也要多多查阅相关资料，夯实自己对分布式的理解。

ES 属于海量数据处理技术，是大型应用必不可少的组成部分。

### 「大型应用」

分布式日志采集系统、海量数据实战、高并发。

该阶段基本属于架构师水平，掌握后至少高级开发是没有任何问题的。

## 第十二阶段：容器

### Docker

[了解] 容器概念

[熟悉] 环境安装与配置

[重点] 镜像、容器基本命令

[重点] DockerFile、镜像制作与发布

[重点] Spring Boot 微服务打包 Docker 镜像

[重点] 容器化 CI/CD

[熟悉] Docker Compose

唯一微信号：ddg\_1987

## Kubernetes

[重点] 安装与配置

[重点 难点] 集群搭建与部署

[熟悉] UI 界面搭建及使用

### 「重点/难点讲解」

分布式微服务，如果使用传统的 Jar 包进行发布和部署，会极其的麻烦，因为服务器的各项环境和配置一旦有些不一样，就会出现服务不可用的问题。

并且一个服务往往会发布多个包，如果手动一个个发布，其工作量是不敢想象的，更不用谈自动扩容等操作了。

容器技术就是为了解决这些问题。**该阶段主要就是应用，不断练习即可。**

### 「云应用」

将之前项目全部上容器化管理，团队也要采用 CI/CD。

之前的中间件也可以尝试用容器管理。

---

有了容器技术，才能将分布式微服务架构优点发挥到极致。

该阶段目标事让学员能够真正驾驭高可用的大型项目。

## 第十三阶段：大数据

### 数据采集

[重点] Flume（日志搜集）

[重点] Kafka（消息队列）

[重点] Sqoop（数据传输）

[熟悉] Canal（MySQL数据同步）

### 数据存储

[重点] HDFS（分布式文件系统）

[重点] TiDB（分布式数据库，兼容 MySQL 语法）

[重点] HBase（分布式数据库，列式存储）

### 离线分析

[重点] MapReduce（离线计算框架）

[重点] Hive（数仓）

[重点] Spark（通用计算）



## 实时计算

[重点] Storm（实时流处理）

[重点] Flink（实时流处理）

### 「重点/难点讲解」

大数据，顾名思义，就是处理大量数据的技术。传统数据量一般低于 PB，而大数据处理的数据往往有 PB 甚至 EB。

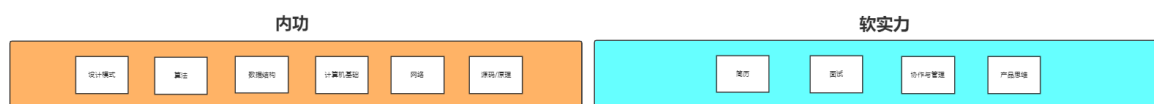
大数据可以算是一个单独的领域，到这一阶段时，你的 Web 水平或者说编程水平应该已经相当高了，所以只需花时间、花精力还是能攻克该阶段的。

### 「大数据项目」

各种大数据处理项目，如物联网、数据中台等。

其实之前第十一阶段已经可以处理海量数据了，在此基础上再进一步，全面掌握大数据技术。

## 其他



## 常用技术

Office文件操作：POI

第三方接入：短信、三方登录、支付等

任务调度：Schedule、XXL-Job

## 内功

### 设计模式

23 种常用设计模式

设计模式实践

### 算法和数据结构

常见数据结构

常见算法

### 计算机基础

计算机组成

内存 IO、磁盘 IO、网络 IO

操作系统

网络

唯一微信号：ddg\_1987

## 源码和原理

JDK 源码

Spring 源码

MQ 源码/原理

MySQL 原理

Redis 原理

ES 原理

## 软实力

---

简历

面试

协作与管理

产品思维