上云 30天计划

Java基础部分

- 语法
  - System 类简介
  - 字符串
    - String
    - String 构造器
    - String 属性简介
    - String 常用方法
    - 字符串与其他引用数据类型/
    - 引用数据类型 equals 方法
    - 引用 String 测试
  - StringBuilder 类 - 解决字符串拼接 expensive
  - StringBuffer
  - StringBuilder 与 StringBuffer 区别
  - 小结：内部自我学习与实践
  - 基本类型包装类型
    - Number 数字抽象类方法
    - Integer
      - 创建整数类实例
      - Integer 常用属性
      - Integer 常用常量
      - 自定义数值比较规则
      - 常用实用算法
      - String、Integer、int 三种类型的转换
    - Long
    - Boolean
    - Byte
    - Short
    - Float
    - Double
    - Character
  - 枚举类型
- 时间类
  - 旧的 Date
  - 新时间 API
  - 时间类型
    - 日期
    - 时间
    - 日期时间
    - 时间戳
    - 时区
    - 日期时间格式化
    - 日期时间解析
  - 日期格式
    - y 年 （四位年）
    - M 月 （数字或文本，长或短）
    - d 日 或 D 一年中的第几天
    - H 时 一天中的小时数（0-23）
    - h 时 一天中的小时数（1-12）
    - m 分 一小时中的分钟数
    - s 秒 一分钟中的秒数
    - S 毫秒
    - E 星期几
    - z 时区
    - Z 时区 -0700 或 -07:00
    - 单引号 '' 内的字符不会被解析为模式字符，例如 'at' 会原样输出
    - 单引号本身需要用 '' 表示，例如 'o''clock' 会输出 o'clock
    - 模式字母区分大小写，M 和 m 含义不同
    - 模式字母重复通常会影响输出格式，例如 yyyy 输出完整年份，yy 输出两位年份
  - Date
    - 创建时间 Date 实例 - 默认为当前时间，可通过 getTime() 获取毫秒数
    - 日期格式化 - 使用 SimpleDateFormat 类将 Date 对象转换为特定格式的字符串
    - 字符串转回 Date 对象 - 使用 SimpleDateFormat 的 parse() 方法，需处理 ParseException 异常
    - 日期计算 - 通过 getTime() 获取毫秒数进行计算，或使用 Calendar 类
  - Calendar
    - 创建日期实例
    - 修改日期时间字段
    - 日期日期计算
  - 工具类设计
    - 设计日期工具类
    - 设计时间工具类
    - 设计日期时间工具类
  - 多线程
  - 线程安全概念
  - 线程状态概念
  - 小结：内部自我学习与实践
- 异常处理
  - 异常概念
  - 异常类型
    - 异常的语法结构
    - 异常的抛出
    - 异常的捕获与处理
  - 自定义异常
    - 自定义异常类
    - 抛出自定义异常
    - 捕获自定义异常
  - 异常链
    - 异常的传递
    - 异常的包装
  - 异常与性能
    - 异常的开销
    - 性能考虑
  - 最佳实践
    - 异常的命名
    - 异常的文档
    - 异常的测试
  - 小结：内部自我学习与实践
- 集合
  - 集合概念
  - 集合类型
    - List
    - Set
    - Queue
    - Map
  - 集合操作
    - 添加元素
    - 删除元素
    - 修改元素
    - 查找元素
    - 遍历集合
    - 排序
    - 过滤
    - 转换
  - 集合工具类
    - Collections 工具类
    - 常用工具方法
    - 自定义工具方法
  - 小结：内部自我学习与实践
  - Map 接口
    - Map 接口与哈希表
    - HashMap 概念
      - HashMap 的工作原理
      - 哈希冲突处理
      - 扩容机制
      - 性能优化
    - LinkedHashMap
    - Properties
    - TreeMap
    - HashSet
    - LinkedHashSet
    - TreeSet
  - 小结：内部自我学习与实践
- IO 流
  - 流概念
  - 字节流
    - 字节流
    - 字节流的使用
    - 字节流的性能
  - 字符流
    - 字符流
    - 字符流的使用
    - 字符流的性能
  - 缓冲流
    - 缓冲流
    - 缓冲流的使用
    - 缓冲流的性能
  - 文件流
    - 文件流
    - 文件流的使用
    - 文件流的性能
  - 对象流
    - 对象流
    - 对象流的使用
    - 对象流的性能
  - 数据流
    - 数据流
    - 数据流的使用
    - 数据流的性能
  - 管道流
    - 管道流
    - 管道流的使用
    - 管道流的性能
  - 小结：内部自我学习与实践
- 多线程
  - 线程概念
  - 线程创建
    - 继承 Thread 类
    - 实现 Runnable 接口
    - 使用 Callable 和 Future
    - 线程池创建
  - 线程生命周期
    - 线程状态
    - 状态转换
    - 阻塞与唤醒
  - 线程同步
    - 锁机制
    - synchronized 关键字
    - 死锁问题
  - 线程通信
    - 等待与通知
    - 生产者-消费者模式
  - 线程安全
    - 线程安全问题
    - 解决方案
    - 最佳实践
  - 线程池
    - 线程池的优势
    - 创建线程池
    - 线程池的配置
  - 小结：内部自我学习与实践
- 反射与动态代理
  - 反射概念
  - 反射的应用
    - 获取类信息
    - 调用方法
    - 修改字段
  - 反射的性能
    - 性能开销
    - 优化方法
  - 动态代理
    - 代理模式
    - 实现动态代理
    - 应用场景
  - 小结：内部自我学习与实践
- 注解与结构化
  - 注解概念
  - 内置注解
    - @Override
    - @Deprecated
    - @SuppressWarnings
  - 自定义注解
    - 定义注解
    - 注解的属性
    - 元注解
    - 自定义注解的使用场景
    - 自定义注解的使用方法
    - 注解的解析
    - 注解的最佳实践
  - 结构化编程
    - 结构化编程的原则
    - 结构化编程的优势
  - 小结：内部自我学习与实践
- 网络编程
  - 网络概念
  - 协议
    - IP
    - TCP
    - UDP
    - HTTP
    - HTTPS
  - 网络模型
    - 分层模型
    - 数据封装
    - 数据解封
  - 网络编程
    - 套接字
    - 客户端
    - 服务端
    - 数据传输
  - 小结：内部自我学习与实践
- Java 进阶总结

数据结构与算法

- 时间复杂度
  - 大O表示法
  - 常见时间复杂度
  - 最坏情况、平均情况、最好情况
  - 实际应用
- 空间复杂度
  - 内存使用
  - 空间换时间
  - 实际应用
- 递归
  - 递归概念
  - 递归的优点
  - 递归的缺点
  - 递归与迭代
  - 递归的应用场景
  - 递归的实现
  - 递归的调试
  - 递归的优化
  - 递归的最佳实践
- 数组
  - 数组概念
  - 数组的特点
  - 数组的操作
  - 数组的应用
  - 数组的实现
  - 数组的优化
  - 数组的最佳实践
- 链表
  - 链表概念
  - 单向链表
  - 双向链表
  - 循环链表
  - 链表的应用
  - 链表的实现
  - 链表的优化
  - 链表的最佳实践
- 栈
  - 栈概念
  - 栈的应用
  - 栈的实现
  - 栈的优化
  - 栈的最佳实践
- 队列
  - 队列概念
  - 队列的应用
  - 队列的实现
  - 队列的优化
  - 队列的最佳实践
- 哈希表
  - 哈希表概念
  - 哈希函数
  - 哈希冲突
  - 哈希表的应用
  - 哈希表的实现
  - 哈希表的优化
  - 哈希表的最佳实践
- 树
  - 树概念
  - 二叉树
  - 二叉搜索树
  - 平衡二叉树
  - 树的应用
  - 树的实现
  - 树的优化
  - 树的最佳实践
- 图
  - 图概念
  - 图的表示
  - 图的遍历
  - 图的应用
  - 图的实现
  - 图的优化
  - 图的最佳实践
- 排序
  - 排序概念
  - 冒泡排序
  - 选择排序
  - 插入排序
  - 快速排序
  - 归并排序
  - 堆排序
  - 排序的应用
  - 排序的实现
  - 排序的优化
  - 排序的最佳实践
- 搜索
  - 搜索概念
  - 线性搜索
  - 二分搜索
  - 搜索的应用
  - 搜索的实现
  - 搜索的优化
  - 搜索的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践
- 动态规划
  - 动态规划概念
  - 动态规划的应用
  - 动态规划的实现
  - 动态规划的优化
  - 动态规划的最佳实践
- 贪心算法
  - 贪心算法概念
  - 贪心算法的应用
  - 贪心算法的实现
  - 贪心算法的优化
  - 贪心算法的最佳实践
- 回溯算法
  - 回溯算法概念
  - 回溯算法的应用
  - 回溯算法的实现
  - 回溯算法的优化
  - 回溯算法的最佳实践
- 分治算法
  - 分治算法概念
  - 分治算法的应用
  - 分治算法的实现
  - 分治算法的优化
  - 分治算法的最佳实践