

# Java知识点总结

## 目录

- 1. 技术社区
- 2. 常用工具
  - 2.1. 编译器
  - 2.2. IDEA插件
  - 2.3. UML绘制
  - 2.4. HTTP请求
  - 2.5. 前端开发编辑
  - 2.6. 思维导图软件
  - 2.7. 编辑器
  - 2.8. 调优工具
  - 2.9. 压测工具
  - 2.10. 版本管理
  - 2.11. 其他
- 3. 计算机基础
- 4. 算法
- 5. 网络TCP
- 6. 网络HTTP
- 7. Unix环境编程
- 8. Shell
  - 文件相关
- 9. Git
- 10. Maven
- 11. Jenkins
- 12. Docker
- 13. Java基础
- 14. JVM
- 15. MySQL
- 16. MyBatis
- 17. Netty
- 18. Dubbo
- 19. Redis
- 20. Spring

- 21. SpringBoot
- 22. SpringCloud
- 23. Dubbo
- 24. RocketMQ
- 25. RabbitMQ
- 26. Kafka
- 27. Nginx
- 28. Tomcat
- 29. elasticsearch
- 30. FastDfs
- 31. Quartz
- 32. 设计模式
  - 32.1. 设计原则
    - 32.1.1. 单一职责原则
    - 32.1.2. 里氏替换原则
    - 32.1.5. 依赖倒置原则
    - 32.1.4. 接口隔离原则
    - 32.1.3. 迪米特法则
    - 32.1.6. 开闭原则
  - 32.2. 单例模式
  - 32.3. 工厂方法模式
  - 32.4. 抽象工厂模式
  - 32.5. 模板方法模式
  - 32.6. 建造者模式
  - 32.7. 代理模式
  - 32.8. 原型模式
  - 32.9. 中介者模式
  - 32.10. 命令模式
  - 32.11. 责任链模式
  - 32.12. 装饰模式
  - 32.13. 策略模式
  - 32.14. 适配器模式
  - 32.15. 迭代器模式
  - 32.16. 组合模式
  - 32.17. 观察者模式
  - 32.18. 门面模式
  - 32.19. 备忘录模式
  - 32.20. 访问者模式
  - 32.21. 状态模式

- [32.22. 解释器模式](#)
- [32.23. 享元模式](#)
- [32.24. 桥梁模式](#)
- [33. Zookeeper](#)
- [34. 分布式系统](#)

# 1. 技术社区

- [博客园](#)
- [开源中国](#)
- [stackoverflow](#)
- [CSDN](#)
- [掘金](#)
- [技术文档下载](#)
- [leetcode](#)
- [牛客网](#)

[目录](#)

# 2. 常用工具

## 2.1. 编译器

- [IDEA](#) （个人觉得IDEA比较好用）
- [Eclipse](#)
- [MyEclipse](#)

## 2.2. IDEA插件

- [Lombok](#)：使用注解，免去实体类中get /set/构造器等代码，需要引入lombok包
- [Alibaba Java Coding Guidelines](#)：《阿里巴巴Java开发规约》扫描插件,安装后右键单击选择扫描，会提示哪些地方不符合规范
- [MyBatis plugin](#)：可以从mybatis mapper接口的方法直接跳转到mapper.xml配置文件
- [jclasslib: JClassLib](#)不但是一个字节码阅读器而且还包含一个类库允许开发者读取,修改,写入Java Class文件与字节码

## 2.3. UML绘制

- <https://www.draw.io/> : 在线UML绘制工具，可以保存到本地或者Github

## 2.4. HTTP请求

- POSTMAN (<https://www.getpostman.com/>) :一款功能强大的网页调试与发送网页HTTP请求的工具
- curl指令

## 2.5. 前端开发编辑

- [vscode](#)
- IDEA
- Brackets 代码提示，颜色区分
- sublime\_text\_3 这个Linux下安装后，不支持中文输入，网上有解决办法。

## 2.6. 思维导图软件

- MindMaster 多平台，免费，可以导出为图片。

## 2.7. 编辑器

- VIM
- GEDIT

## 2.8. 调优工具

- visualVm(<https://visualvm.github.io/>):是一款免费的,集成了多个 JDK 命令行工具的可视化工具,它能为您提供强大的分析能力,对 Java 应用程序做性能分析和调优.

## 2.9. 压测工具

- Jmeter 可以开多个线程对接口进行压测，并分析压测结果。

## 2.10. 版本管理

- git
- svn
- 码云
- Git在线练习 [try git/git real/Learn Git Branching](#)

## 2.11. 其他

- SpringBoot Banner 字符生成 [根据图片生成/根据文字生成](#)

[目录](#)

# 3. 计算机基础

[目录](#)

# 4. 算法

[目录](#)

# 5. 网络TCP

[目录](#)

# 6. 网络HTTP

[目录](#)

# 7. Unix环境编程

[目录](#)

# 8. Shell

## 文件相关

[目录](#)

- 路径
  - / 根路径
  - . 当前目录
  - .. 上一级目录
  - ~/ 当前用户目录

- cd path 进入路径
  - cd - 返回上一级目录
  - 文件列表
    - ls 只列出当前目录文件名称
    - ls -l 包含文件权限等信息
    - ls -a 包含"."开头的隐藏文件
    - ls -F 区分文件和目录
    - ls -R 递归显示目录
    - ls -l aa 只显示带aa的列表, "?"单个字符, "\*"多个字符
    - pwd 显示当前绝对路径
  - 处理文件
    - touch 创建文件
    - mkdir 创建目录
    - cp source dest 复制文件, 加 -r 用于复制目录
    - rm 删除, -r 删除目录 -f 强制删除文件
    - mv source dest 重命名/复制文件
  - 查看文件
    - file xxx 查看文件类型
    - cat 打印文件数据, -n 加上行号, -b 只给有文本的加上行号
  - 文件权限
    - 文件权限包括用户(U)-组(G)-其他用户(O),包括读(4-r)写(2-w)执行(1-x)权限。
    - chmod u+x file 用户添加执行权限
    - chmod u-x file 用户去除执行权限
    - chmod 111 file 所有的都为执行权限
    - chmod 721 file 用户有读写执行, 组用户有写, 其他用户有执行权限
- rw-rw-r-- 1 lgj lgj 1215779 Jun 28 18:08 1.mp4
- 文件链接：给文件创建虚拟副本
    - 符号链接:仅保存引用, 文件内容和原文件不一样, 和源文件同步更新
    - 硬链接: 保存引用和文件信息, 和源文件同步更新
    - 使用cp 复制链接文件时, 复制的文件仍然保留和源文件之间的链接关系, 一般不这么做。
    - ln source dest ,加-s 时创建的是软连接。

目录

## 9. Git

目录

## 10. Maven

[目录](#)

## 11. Jenkins

[目录](#)

## 12. Docker

[目录](#)

## 13. Java基础

[目录](#)

## 14. JVM

[目录](#)

## 15. MySQL

[目录](#)

## 16. MyBatis

[目录](#)

## 17. Netty

[目录](#)

## 18. Dubbo

[目录](#)

## 19. Redis

[目录](#)

## 20. Spring

## 21. SpringBoot

[目录](#)

## 22. SpringCloud

[目录](#)

## 23. Dubbo

[目录](#)

## 24. RocketMQ

[目录](#)

## 25. RabbitMQ

[目录](#)

## 26. Kafka

[目录](#)

## 27. Nginx

[目录](#)

## 28. Tomcat

[目录](#)

## 29. elasticsearch

[目录](#)

## 30. FastDfs

[目录](#)

## 31. Quartz

[目录](#)



# 32. 设计模式

[目录](#)

## 32.1. 设计原则

[目录](#)

### 32.1.1. 单一职责原则

[目录](#)

- 每个类只负责单一的功能，避免出现复杂臃肿的类。造成维护困难。

### 32.1.2. 里氏替换原则

- 父类出现的地方子类也可以出现。方便实现类替换，减少代码修改。

### 32.1.5. 依赖倒置原则

- 高层模块不应该依赖细节，而应该依赖其抽象
- 也就是说定义类属性时，使用接口或者抽象类来定义，而不是实现类。

### 32.1.4. 接口隔离原则

- 创建单一的接口，不要创建臃肿的接口。每个接口应该只是负责单一的功能，避免增加新抽象方法时，造成较多的实现类需要修改。

### 32.1.3. 迪米特法则

- 一个类应当降低对其他类的依赖,避免其他类修改时，使用类也要进行修改。

### 32.1.6. 开闭原则

- 对修改关闭，对扩展开放
- 当需要增加新功能时，应当避免修改原类的代码，应当通过一些设计模式来进行功能扩展。比如装饰模式，这样可以达到对其他使用类的影响。
- 以上的原则都是施行指南，开闭原则是最终目标。

## 32.2. 单例模式

## 32.3. 工厂方法模式

[目录](#)

## 32.4. 抽象工厂模式

[目录](#)

## 32.5. 模板方法模式

[目录](#)

## 32.6. 建造者模式

[目录](#)

## 32.7. 代理模式

[目录](#)

## 32.8. 原型模式

[目录](#)

## 32.9. 中介者模式

[目录](#)

## 32.10. 命令模式

[目录](#)

## 32.11. 责任链模式

[目录](#)

## 32.12. 装饰模式

[目录](#)

## 32.13. 策略模式

[目录](#)

## 32.14. 适配器模式

[目录](#)

## 32.15. 迭代器模式

[目录](#)

## 32.16. 组合模式

[目录](#)

## 32.17. 观察者模式

[目录](#)

## 32.18. 门面模式

[目录](#)

## 32.19. 备忘录模式

[目录](#)

## 32.20. 访问者模式

[目录](#)

## 32.21. 状态模式

[目录](#)

## 32.22. 解释器模式

[目录](#)

## 32.23. 享元模式

[目录](#)

## 32.24. 桥梁模式

[目录](#)

## 33. Zookeeper

[目录](#)

## 34. 分布式系统

[目录](#)

[目录](#)