**Java知识整理**

1. **面向对象的思想**
   1. **基本概念**
      1. **封装**

将客观事物封装成抽象的类，这些类可以把自己的数据与方法让可信的类或对象操作，对不可信的类进行信息隐藏。

* + 1. **继承**

使用现有类的所有的功能，在无需重新编写原来的类的情况下对这些功能进行拓展。Java中只支持类的单继承性。一个类只能有一个直接的父类。

* + 1. **多态**

一个类的实例的相同方法在不同情况下有不同的表现形式。最常见的多态就是将子类传入父类参数中，运行时调用父类方法时通过传入的子类决定具体的内部结构或行为。

实现多态的三个条件

继承

重写（子类对象重写父类的方法）

父类引用指向子类对象

* 1. **设计原则**
     1. **单一职责原则**
     2. **里氏替换原则**
     3. **依赖倒转原则**
     4. **迪米特法则**
     5. **聚合复用原则**
     6. **开放封闭原则**
     7. **接口隔离原则**

1. **Java重点基础知识**
   1. **java常用类、接口**
   2. **java IO**
      1. **BIO体系**
      2. **BIO分类**
      3. **BIO基本用法**
      4. **NIO基本用法**
      5. **开源IO工具**
   3. **Java多线程**
      1. **一些概念**
      2. **线程基本用法**
      3. **同步**
      4. **死锁**
      5. **synchronized与volatile**
      6. **锁**
      7. **JUC**
   4. **Java网络编程**
      1. **一些概念**
      2. **Socket基本用法**
      3. **爬虫原理**
      4. **Netty**
   5. **Java集合**
      1. **泛型**
      2. **集合体系结构**
      3. **基本用法**
      4. **java.util.concurrent包下的容器**
   6. **Java异常**
      1. **异常机制**
      2. **异常体系结构**
      3. **异常处理原则**
   7. **Java反射**
      1. **反射的作用**
      2. **反射的基本用法**
      3. **自定义注解**
      4. **JDK与CGLIB动态代理**
      5. **Java动态编译**
   8. **JVM**
      1. **JVM内存结构**
      2. **Java类加载机制**
      3. **垃圾回收**
      4. **JVM监控工具**
      5. **JVM优化**
   9. **其他**
2. **设计模式**
   1. **总览**
   2. **单例模式**
3. **协议**
   1. **TCP**
   2. **UDP**
   3. **HTTP**
   4. **RPC**
4. **框架**
   1. **mybaits**
   2. **spring**
   3. **spring mvc**
   4. **spring boot**
   5. **spring cloud**
   6. **dubbo**
5. **微服务架构**
   1. **概念**
   2. **微服务基本组件**
   3. **Spring cloud netflix**
   4. **Spring cloud**