

java随笔

其实我们把一个类的构造方法私有化就是创造了单例模式（单纯的写法 而思想的话非常深入）

外面不能再直接构造了，

<http://blog.csdn.net/zuoxiaolong8810/article/details/9026775> 这个是我找到的一个关于设计模式的博客

这个是代理模式

<http://blog.csdn.net/zhengzhib/article/details/7190158> 这是关于依赖、关联、聚合和组合这四种关系之间区别

代理是什么

就是一个拦截器

就是一个切面

就是一个我们有功能的类，我们想访问它，但是直接访问总是太没用吗，我们有的最核心的功能，虽然是我们最需要的，但是我们不止是这些功能是吧，就算不能，加广告也不错的是吧

这个时候，我们就启动了代理，

我们不直接new这个核心功能，new拦截到这个核心功能的类，就是我们说的代理类，当然，我们还可以一层层的来代理代理，一层一层的切面

一层的一层加特殊的功能，就是传来传去

代理模式是用构造器来进行代理的，这个是没办法改变的，关于这个核心功能的方法是用什么来规范呢，当然首选是接口，当然我们也可以用继承的方式，但是为什么不常用继承的方式进行代理，这就是扯到

聚合好在哪了

好在这个构造器的设置

可以循环嵌套 想调就调 而且由于这个构造器

我们可以相对自由的实现，我们更换的服务对象的时候，更加的比这个继承的要简单，继承是写死的，我们需要

再继承一个类，用着个继承的新的服务对象来实现方法，聚合的话，直接传进去，比继承要省心，因为构造器不是从父类继承而来，而是重新设置，这样的话，就方便很多

idea的一个插件

sexy editor

设置背景

@Autowired 注解就是可以自动装配设置注入属性 自动装配构造器注入属性 直接注入属性 可以不用set方法

还有我们在bean中需要装配的方法

代码的世界总是有你想不到的东西 你没学就是等于不会 **不要给自己增加压力**

理性的思考同时不畏困难才是避免痛苦也高效成长的办法

纸上得来终觉浅 绝知此事要躬行

这是SSH构建项目的层次结构 只是个人搜索总结

entity层 这是最基本的实体层

Dao层 DAO是负责数据库的增删改查操作

Action层 典型的MVC 取代原来的Servlet

Service层 用来编写业务逻辑的和事物控制的

db层 连接数据库的层 如果有框架的放弃

关于数据库的想法

一个表如果我们想表示集合这一属性 但是比较难以表示 其实集合就是另外的一张表 而我们原来的表一个属性对应了这张表 可以由这张表的属性1,2,3, 4来查另外那张表的1,2,3,4行

模拟出在java中集合的效果

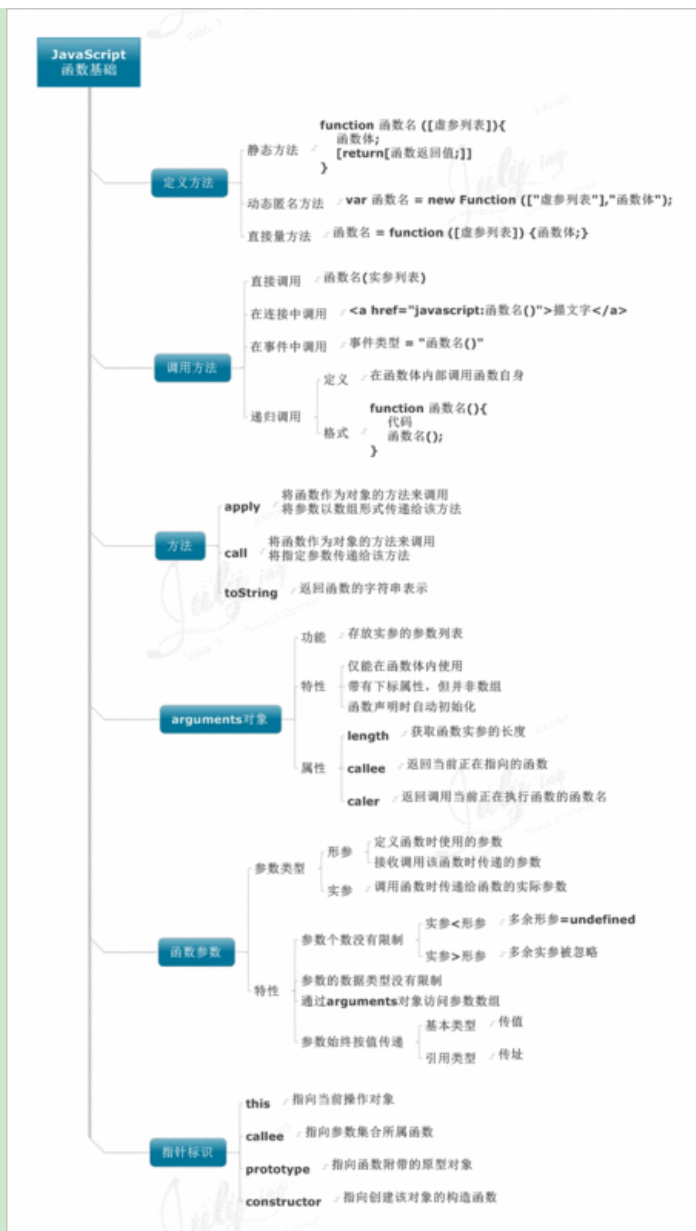
关于编码的密码这本书 很适合组成原理一无所知的人

逻辑门 继电器 二进制 布莱页盲文 摩尔斯电码 手电筒通信 布尔代数 电路的具体实现 关于进制的说法与说明

<http://droidyue.com/blog/2014/12/07/differences-between-stack-and-heap-in-java/> java中堆栈的区别

下面是关于JS的思维导图

<https://segmentfault.com/a/1190000000660786#articleHeader18>来源于这个博客



<http://imfei.blog.51cto.com/1849649/511689/>

<https://segmentfault.com/a/1190000003072424>

这是MySQL 索引相关的讲解

<http://www.bkjia.com/Javabc/1063046.html>

这里是关于多模块的东西 我不知道该占这个到底有没有意义

在学习一个新的框架，或者一门新的技术的时候 寻找一个好的demo 多敲几遍 比，看书理解是要快多的 这个demo应该有相关的说明 这大概就是文档 然后——对比 理解是非常快的

关于思想的训练 就是通过应对不同场景来进行解决的方案 语言只是工具 而工具是为了解决问题 设想不同的场景来进行不同的准备 在工作中需要注意

做之前想清楚 切记

关于SS的账号



服务器：sg103.ss99.ren

端口：30067

密码：370081

加密方式：aes-256-cfb

有效期：2017-08-04 18:00:50

在学习一门语言或者一门复杂的工具的过程中 总是想问自己这个功能有什么用 这个特性干什么 其实这都是没必要的 学习是有积累之后能照看全局之后 才去想这些事 没有积累纵使想出来了 也是花费大量的时间来达到一个原本积累之后能够轻而易举想出来的小目标 所以有的时候学习不要问自己为什么 而是去单纯的学习 问自己为什么的时候 是在自己有积累之后 能够掌握了这门语言的大局 这样去思考才能收获到很多 也是如同顺水依然理所当然般的水到渠成

人的思想是一个过程 看到一个东西自然就会去联想到以前学过的知识套路相关的逻辑 这个是一个具有局限的过程 但是这个局限是相对的局限

当思考的焦点从一片知识的另一个我们需要的点再次展开 就联想到其他的知识区域与相关的逻辑 我们的大脑的遗忘程度与只是与当时记忆的程度相关 而不是由主观臆断来决定（不要陷入主观否定与肯定的混乱之中） 为了保证效率 我们需要对每个点的知识掌握达到相对平均的水平 当然这是不太现实的 但是又是可以追求的 因为这样才会达到复习的时候效率化

优化IF/ELSE

1. 命令模式：比如把每个name（或者多个name组合）封装成一个个命令对象，那么程序的执行过程是，首先按照那么去查找command，比如command提供一个isMatch(String name)的方法，执行过程就比较简单了，command.execute(...)，这样增加name就是增加一个command类；
2. 使用Disptcher+Handler的模式，即所有的请求集中在Dispatcher，而Dispatcher根据规则（简单的Map或者复杂的规则都可以，根据实际情况），分发给不同的Handler；
3. 使用一个表（List、Set等）来存储name，这个最简单，貌似有人提过了。

总体上来说解决程序分支复杂性的关键是：将简单数据类型（比如String，int等）转化为对象类型进行处理（比如name --> command），这样可以将数据和数据的操作（对象类型中定义的）很好的结合在一起。