

构建用于正则表达式的抽象JavaAPI计算机等级考试 PDF转换
可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022__E6_9E_84_E5_BB_BA_E7_94_A8_E4_c97_646199.htm 在我的经验中，大多数 Java 开发人员都需要解析某种文本。通常，这意味着他们最初要花一些时间使用象 indexOf 或 substring 那样的与 Java 字符串相关的函数或方法，并且希望输入格式永远不变。但是，如果输入格式改变，那么用于读取新格式的代码维护起来就会变得更复杂、更困难。最后，代码可能需要支持自动换行（word wrapping）、区分大小写等。由于逻辑变得更加复杂，所以维护也变得很困难。因为任何更改都可能产生副作用并使文本解析器的其它部分停止工作，所以开发人员需要时间修正这些小错误。有一定 Perl 经验的开发人员可能也有过使用正则表达式的经验。如果够幸运（或优秀）的话，这位开发人员能够说服团队其余的人（或至少是团队领导）使用这项技术。新的方法将取消编写用来调用 String 方法的多行代码，它意味着将解析器逻辑的核心委托出去，并替换为 regexp 库。接受了有 Perl5 经验的开发人员的建议后，团队必须选择哪个 regex 实现最适合他们的项目。然后他们需要学习如何使用它。在简要地研究了从因特网上找到的众多可选方案后，假设团队决定从人们更熟悉的库中选择一个使用，如属于 Jakarta 项目的 Oro。接下来，对解析器进行较大程度地重构或几乎重新编写，并且解析器最终使用了 Oro 的类，如 Perl5Compiler、Perl5Matcher 等。这一决定的后果很明显：代码与 Jakarta Oro 的类紧密地耦合在一起。团队承担了风险，因为不知道非功能性需求（如性能或线程模型）是否将得到

满足。团队已花费时间和财力来学习并重新编写代码，以使它使用 regex 库。如果他们的决定是错误的并且选择了新的库，则这一工作在成本上将不会有很大区别，因为将需要再次重新编写代码。即使库工作正常，如果他们决定应该迁移到全新的库（例如，包括在 JDK 1.4 中的库），怎么办？去耦的好处有没有办法使团队知道哪个实现最适合他们的需要呢（不仅现在能将来也能）？让我们试着寻找答案。避免依赖任何特定的实现 前面的情形在软件工程中十分常见。在有些情况中，这样的情形会导致较大的投资和较长的延期。当不了解所有后果就作出决定而且决策制定人不太走运或缺乏必需的经验时，就常常会发生这种情况。可将该情形概括如下：您需要某种提供者 您没有选择最佳提供者的客观标准 您希望能用最低的成本来评估所有的待选项 所作的决定不应将您束缚在所选的提供者上 这一问题的解决方法是使代码更加独立于提供者。这引入了新的层？同时去除客户机和提供者的耦合的层。在服务器端开发中，很容易找到使用该方法的模式或体系结构。下面引用一些示例：对于 J2EE，您主要关注如何构建应用程序而不是应用程序服务器的细节。数据访问对象（Data Access Object，DAO）模式隐藏了如何访问数据库（或 LDAP 服务器、XML 文件等）的细节和复杂性，因为它提供了访问抽象持久存储层的方法，而您则不需要在客户机代码中处理数据库问题（数据实际存储在哪里）。这不是四人组（Gang of Four，GoF）模式，而是 Sun 的 J2EE 最佳实践的一部分。在假想的开发团队示例中，他们正在寻找这样的层：抽象所有正则表达式实现背后的概念。团队就可以着重学习和理解这些概念。他们所学的可以应用到任何实现

或版本。支持新的库且没有副作用。基于插件体系结构，动态选择执行 regexp 模式的实际库，并且适配器不会被耦合。新库仅会引入对新适配器的需要。提供比较不同可选方案的方法。一个简单的基准实用程序就可以显示有趣的性能测量结果。如果对每个实现都执行这样的实用程序，团队就会获得有价值的信息并能选择最好的可选方案。听起来不错，但.....任何去耦方法都至少有一个缺点：如果客户机代码仅需要一个实现所提供的特定功能，怎么办？您不能使用任何其它实现，因此您最终将代码与该实现耦合。也许将来会在这方面有所改善，但您现在却束手无策。这样的示例并不象您想的那样少。在 regexp 领域中，一些编译器选项仅被某些实现支持。如果您的客户机代码需要这种特定的功能，那么这个一般层是不够的？至少从迄今对它描述来看是不够的。附加层是否应支持每个实现的所有非公共功能，并且如果选择了不支持该实现的附加层则抛出异常？那可以是一种解决方案，但它并不支持仅定义公共抽象概念这一最初目标。有一个 GoF 模式非常适合这种情形：职责链（Chain of Responsibility）。它在设计中引入了另一种间接方法。用这种方法，客户机代码向能处理其所发消息的实体列表发送消息或命令。列表项被组织成链，因此消息可按顺序被处理并且在到达链尾之前被用掉。在这种情况下，可以通过特殊类型的消息对仅被某些实现支持的特定功能建模。由链中的每一项根据其是否了解这些功能来决定是否将该消息传给下一项。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com