

信 息 化 培 训 教 材 及 资 料

车辆购置税征收管理系统软件 (V1.1)

核心技术手册 (技术篇)

本书编写组 编

 中国税务出版社


中税出版

车辆购置税征收管理系统软件 (V1.1)
核心技术手册 (技术篇)

本书编写组 编

ISBN 978-7-80235-196-7



9 787802 351967 >

总定价：498.00元

信息化培训教材及资料

车辆购置税征收管理系统软件

V1.1 核心技术手册

(技术篇)

本书编写组 编

中国税务出版社

图书在版编目 (CIP) 数据

车辆购置税征收管理系统软件 V1.1 核心技术手册. 技术篇 / 本书编写组编. —北京: 中国税务出版社, 2008.9
(信息化培训教材及资料)

ISBN 978-7-80235-196-7

I. 车… II. 本… III. 车辆-消费税-税收管理-管理信息系统-中国-技术手册 IV. F812.423-39

中国版本图书馆 CIP 数据核字 (2008) 第 126196 号

版权所有·侵权必究

书 名: 车辆购置税征收管理系统软件 V1.1 核心技术手册 (技术篇)

作 者: 本书编写组 编

责任编辑: 王迎新

责任校对: 于 玲

技术设计: 刘冬珂

出版发行: 中国税务出版社

北京市西城区木樨地北里甲 11 号 (国宏大厦 B 座)

邮编: 100038

http: //www. taxation. cn

E-mail: taxp@tom.com

发行部电话: (010) 63908889/90/91

邮购直销电话: (010) 63908837 传真: (010) 63908835

经 销: 各地新华书店

印 刷: 北京外文印刷厂

规 格: 787×1092 毫米 1/16

印 张: 9.75

字 数: 226000 字

版 次: 2009 年 1 月第 1 版 2009 年 1 月北京第 1 次印刷

书 号: ISBN 978-7-80235-196-7/F·1116

总 定 价: 498.00 元

如发现印装错误 本社发行部负责调换

序

本书对车辆购置税征收管理软件 V1.1 的核心技术架构作了一个较为详细的介绍，使得税务系统的技术人员能够依据本教材对车辆购置税征收管理软件 V1.1 有一个系统地了解，并能够对简单的功能进行扩展和优化。

本书分为技术、应用两篇。技术篇涵盖了车辆购置税征收管理软件 V1.1 的大部分技术内容，包括总体架构、Web 层、事务层、应用层以及数据层的架构设计；此外，对安全等具体的技术应用也作了详细的介绍。应用篇涵盖了车辆购置税征收管理软件 V1.1 全部业务内容，包括征收业务、减免税业务、档案管理、票证管理、一条龙服务、车价信息管理、报表分析、权限管理、系统管理、外部接口等；此外对客户端本地控件的使用也作了详细的介绍。

本书面向的读者为税务系统的技术人员，读者需要的基础技能包括：java 语言、web 开发技术、XML 基础、UML 基础、J2EE 基础、weblogic8.1 以及 oracle9i 的基础知识及相关的业务知识。

本书由国家税务总局信息中心组织编写，李伟、袁立炫、苏小全、李志、朱会彦、童咪娜、朱斌、鲁钰锋、刘翔宇、王晔、陈亚萍、王彦鹏、王殿全、李传松、章谦、王皓、李方楠、耿煜等同志具体参与了编写工作，姚琴主审。

本书在编写过程中得到了天津市国家税务局、武汉市国家税务局的大力支持，皖通科技股份有限公司积极派人参与了编写工作，在此一并表示衷心的感谢。

由于时间紧，水平有限，本书难免有错误和不足，如有疏漏之处，敬请批评指正。

车辆购置税征收管理软件核心技术指南编写组

二〇〇八年五月

目 录

第一章 引言	(1)
1. 架构	(1)
2. 框架	(2)
3. 设计模式	(3)
4. 参考资料	(4)
第二章 总体概述	(5)
1. 总体架构	(5)
2. 架构实现	(6)
第三章 WEB 层及其组件	(12)
1. 页面及表示层	(12)
2. Spring MVC 框架	(13)
3. AJAX 技术的应用	(17)
4. 主界面框架	(26)
第四章 事务层架构	(28)
第五章 中间层架构	(54)
1. 业务接口及实现	(54)
2. 业务规则引擎	(55)
第六章 数据持久层架构	(61)
1. iBatis 的 Spring 集成	(61)
2. 数据缓存	(62)
3. 存储过程及函数	(64)
第七章 安全模型	(65)
1. JAAS 简述	(65)
2. 认证提供者	(65)
3. 安全认证	(65)
3.1 Web 认证	(66)
3.2 EJB 认证	(68)
第八章 附录 A 数据库结构	(69)
1. 票证管理表	(69)
1.1 票证出入库记录 (PZ_CRK)	(70)
1.2 票证出入库单 (PZ_CRKD)	(70)
1.3 票证出入库单明细 (PZ_CRKDMX)	(71)
1.4 票证现存 (PZ_XC)	(71)

1.5	机构票证计划 (PZ_JH)	(71)
1.6	票证计划明细 (PZ_JHMX)	(72)
2.	权限管理表	(72)
2.1	可访问资源 (QX_ZY)	(73)
2.2	资源层次 (QX_ZYCC)	(73)
2.3	用户 (QX_YH)	(73)
2.4	系统管理员可授权组 (QX_KSQZ)	(73)
2.5	用户拥有的资源 (QX_YHZY)	(74)
2.6	在线用户 (QX_ZXYH)	(74)
2.7	用户窗口代码 (QX_YHCK)	(74)
2.8	审核记录 (QX_SHJL)	(75)
3.	数据交换表	(75)
3.1	MQ 传输跟踪表 (WB_MQGZ)	(75)
4.	系统管理表	(76)
4.1	系统配置段 (XT_PZD)	(76)
4.2	系统配置 (XT_PZ)	(76)
4.3	机构专用配置 (XT_ZYPZ)	(77)
4.4	政策法规 (XT_ZCFG)	(77)
4.5	政策法规关键字 (XT_ZCFG_GJZ)	(77)
4.6	被系统管理机构 (XT_NBJG)	(78)
4.7	系统业务规则 (XT_YWGZ)	(78)
4.8	套打配置 (XT_TD)	(78)
4.9	系统执行锁 (XT_ZXS)	(78)
5.	征收业务表	(79)
5.1	申报表 (ZS_SBB)	(80)
5.2	纳税申报的车型信息 (ZS_SBB_CX)	(81)
5.3	申报纳税的纳税人信息 (ZS_SBB_NSR)	(81)
5.4	申报表确定退税信息 (ZS_SBB_QDTSE)	(82)
5.5	申报表扩展 (ZS_SBB_KZ)	(82)
5.6	申报表代理人 (ZS_SBB_DLR)	(83)
5.7	申报表核发完税证明信息 (ZS_SBB_FZ)	(83)
5.8	申报表税款征收信息 (ZS_SBB_NS)	(83)
5.9	申报表确定税款信息 (ZS_SBB_QDNSE)	(84)
5.10	纳税申报的车辆信息 (ZS_SBB_CL)	(85)
5.11	新征业务车辆购置信息 (ZS_SBB_GZ)	(85)
5.12	汇总缴款书 (ZS_HZJKS)	(86)
5.13	申报表待汇总的开票信息 (ZS_SBB_KPDHZ)	(86)
6.	综合管理表	(88)
6.1	车税数据采集 (ZH_CSSJ_CJ)	(89)

6.2	车税数据统计 (ZH_CSSJ_CJTJ)	(89)
6.3	车税数据跟踪 (ZH_CSSJ_CJGZ)	(90)
6.4	车税数据明细 (ZH_CSSJ_CJMX)	(90)
6.5	异常发票分发 (ZH_YCFP_FF)	(91)
6.6	异常发票分发统计 (ZH_YCFP_FFTJ)	(91)
6.7	异常发票分发跟踪 (ZH_YCFP_FFGZ)	(92)
6.8	异常发票分发明细 (ZH_YCFP_FFMX)	(92)
6.9	车辆识别号分发 (ZH_VIN_FF)	(93)
6.10	车辆识别号分发统计 (ZH_VIN_FFTJ)	(93)
6.11	车辆识别号分发跟踪 (ZH_VIN_FFGZ)	(93)
6.12	车辆识别号分发明细 (ZH_VIN_FFMX)	(94)
6.13	厂家已生产车辆识别代码 (ZH_VIN_SC)	(94)
7.	减免税管理表	(95)
7.1	免征图册车型 (JM_TCCX)	(96)
7.2	免征图册图片 (JM_TCTP)	(96)
7.3	免税申请表 (JM_SQB)	(96)
7.4	免税申请扩展 (JM_SQB_KZ)	(97)
7.5	免税表代理人 (JM_SQB_DLR)	(97)
7.6	免税申请审批记录 (JM_SQB_SPJL)	(98)
7.7	申请免税的纳税人信息 (JM_SQB_NSR)	(98)
7.8	申请免税的车型信息 (JM_SQB_CX)	(98)
7.9	申请免税的车辆信息 (JM_SQB_CL)	(99)
7.10	免税申请车型图片 (JM_SQB_TP)	(99)
8.	基础代码表	(100)
8.1	档案清单条目 (DM_DAQDTM)	(102)
8.2	对象类型代码 (DM_DXLX)	(102)
8.3	操作类型代码 (DM_DXCZLX)	(102)
8.4	增值税税率表 (DM_ZZSSL)	(102)
8.5	登记注册类型代码 (DM_DJZCLX)	(102)
8.6	行业代码 (DM_HY)	(103)
8.7	银行种类代码 (DM_YHZL)	(103)
8.8	银行代码 (DM_YH)	(103)
8.9	预算科目代码 (DM_YSKM)	(104)
8.10	预算级次代码 (DM_YSJC)	(104)
8.11	预算分配比例代码 (DM_YSFPBL)	(104)
8.12	征收项目代码 (DM_ZSXM)	(105)
8.13	税款种类代码 (DM_SKZL)	(105)
8.14	征收项目税款种类 (DM_ZSXMSKZL)	(105)
8.15	税款属性代码 (DM_SKSX)	(105)

8.16	政策法规类别代码 (DM_ZCFGLB)	(106)
8.17	收款国库代码 (DM_SKGK)	(106)
8.18	票证非正常使用代码 (DM_PZHZCSY)	(106)
8.19	车辆来源代码 (DM_CLLY)	(106)
8.20	车辆产地代码 (DM_CD)	(106)
8.21	车辆生产厂商代码 (DM_SCCJ)	(107)
8.22	税务机关代码 (DM_SWJG)	(107)
8.23	税务人员代码 (DM_SWRY)	(107)
8.24	票证大类代码 (DM_PZDL)	(108)
8.25	票证种类代码 (DM_PZZL)	(108)
8.26	减免税条件代码 (DM_JMTJ)	(108)
8.27	减免文件代码 (DM_JMWJ)	(109)
8.28	车辆类别大类代码 (DM_CLDL)	(109)
8.29	车辆类别代码 (DM_CLZL)	(109)
8.30	燃料种类代码 (DM_RLZL)	(109)
8.31	车辆颜色代码 (DM_CLYS)	(110)
8.32	所有人证件类型代码 (DM_ZJZL)	(110)
8.33	免征车型大类代码 (DM_MZDL)	(110)
8.34	免征车型中类代码 (DM_MZZL)	(110)
8.35	税款支付方式代码 (DM_ZFFS)	(111)
9.	车价信息管理表	(111)
9.1	计税价格审核表 (CJ_SH)	(113)
9.2	暂行车辆型号 (CJ_ZXCX)	(113)
9.3	当前暂行最低计税价格 (CJ_DQZX)	(114)
9.4	当前备案车价信息 (CJ_DQBA)	(114)
9.5	暂行最低计税价格 (CJ_LSZX)	(114)
9.6	备案车价信息 (CJ_LSBA)	(115)
9.7	车价信息采集 (CJ_CJ)	(115)
9.8	车价信息采集明细 (CJ_CJMX)	(116)
9.9	新增厂家采集 (CJ_XCJCJ)	(116)
9.10	车价信息采集跟踪 (CJ_CJGZ)	(117)
9.11	机关车价采集范围 (CJ_JGCJFW)	(117)
9.12	车辆最低计税价格版本 (CJ_BB)	(117)
9.13	车辆最低计税价格信息 (CJ_LS)	(118)
9.14	车价信息新增车型 (CJ_XCX)	(118)
9.15	车价信息新厂家 (CJ_XCJ)	(119)
9.16	历史厂家备案 (CJ_LSCJBA)	(119)
9.17	车辆型号 (CJ_CX)	(119)
9.18	当前车辆最低计税价格信息 (CJ_DQ)	(120)

9.19 机构常用车型 (CJ_CYCX)	(120)
10. 报表管理表	(121)
10.1 报表 (BB_BB)	(122)
10.2 报表参数 (BB_CS)	(122)
10.3 报表显示格式 (BB_XSGS)	(122)
10.4 报表构造格式 (BB_GZGS)	(123)
10.5 报表 Excel 模版 (BB_EXCEL)	(123)
10.6 系统内表 (BB_SJB)	(123)
10.7 收入日报 (BB_SRRB)	(123)
10.8 收入月报 (BB_SRYB)	(124)
10.9 票证日报 (BB_PZRB)	(124)
10.10 票证月报 (BB_PZYB)	(125)
10.11 车辆价格分布月报 (BB_CLJGFBYB)	(125)
10.12 业务量月报 (BB_YWLYB)	(126)
10.13 报表临时缓存表 (BB_LSHC)	(126)
11. 档案管理表	(128)
11.1 车辆档案 (GD_CLDA)	(130)
11.2 档案借阅记录 (GD_DAJYJL)	(131)
11.3 归档纳税申报表 (GD_NSSB)	(131)
11.4 车辆购置信息 (GD_NSSB_GZ)	(133)
11.5 变更前信息 (GD_NSSB_BGQ)	(133)
11.6 申报退税信息 (GD_NSSB_TS)	(134)
11.7 申报免税信息 (GD_NSSB_MS)	(134)
11.8 档案申报表扩展 (GD_NSSB_KZ)	(135)
11.9 待转入车辆档案 (GD_DZRCLDA)	(135)
11.10 档案编号规则 (GD_BHGZ)	(136)
11.11 申报表开票信息 (GD_NSSB_KP)	(137)
第九章 附录 B 基础数据字典	(138)
1. 申报类型 (SBB_LX)	(138)
2. 申报子类型 (SBB_ZLX)	(138)
3. 业务状态 (ZS_SBB_ZT)	(139)
4. 汇总控制状态 (HZKZ)	(139)
5. 档案建档类型 (DA_JDLX)	(140)
6. 档案状态 (DA_ZT)	(140)
7. 档案位置 (DAWZ)	(140)
8. 免税申请状态 (ZT)	(140)
9. 免税申请类型 (SQB_LX)	(141)
10. 票证出入库类型 (CRKLX_DM)	(141)
11. 一条龙数据状态 (ZT)	(141)

12. 车税数据操作类型（CZLB）	(142)
13. 车辆识别号、异常发票操作类型（CZLB）	(142)
14. 错误类型（CWLX）	(142)
15. 统计类型（TJLX）	(143)
16. 计税价格审核状态（ZT）	(143)
17. 车价信息采集状态（ZT）	(143)
18. 车价信息采集操作类别（CZLB）	(144)
19. 车价采集信息明细有效标致类型（YXBZ）	(144)

第一章 引言

1. 架构

软件体系结构通常被称为架构，ANSI/IEEE 610.12-1990 软件工程标准词汇对于体系结构定义是：“体系结构是以构件、构件之间的关系、构件与环境之间的关系为内容的某一系统的基本组织结构以及指导上述内容设计与演化的原理（principle）”。可以简单理解为：体系结构 = { 构件（component），连接件（connector），约束（constrain）}。其中构件可以是一组代码，如程序的模块；也可以是一个独立的程序，如数据库服务器。连接件可以是过程调用、管道、远程过程调用（RPC）等，用于表示构件之间的相互作用。约束一般为对象连接时的规则，或指明构件连接的形式和条件，例如，上层构件可要求下层构件的服务，反之不行；两对象不得递归地发送消息；什么条件下此种连接无效等。

一般而言，架构有两个要素：首先，它是一个软件系统从整体到部分的最高层次的划分，一个软件系统中的元件首先是逻辑元件，这些逻辑元件如何放到硬件上，以及这些元件如何为整个系统的可扩展性、可靠性、强壮性、灵活性、性能等作出贡献，是非常重要的信息；其次，建造一个系统所作出的最高层次的、以后难以更改的、商业的和技术的决定，也就是说进行软件设计需要作出的决定中，必然会包括逻辑结构、物理结构，以及它们如何影响到系统的所有非功能性特征，这些决定中会有很多是一旦作出，就很难更改。

在开发一个系统之前会有很多的重要决定需要事先作出，而一旦系统开始进行详细设计甚至开发，这些决定就很难更改甚至无法更改。显然，这样的决定必定是有关系统设计成败的最重要决定，必须经过非常慎重的研究和考察。引入架构就是要从更高的层面去考虑问题，把注意力集中在“不变”的因素上，使得最终的软件系统可以实现如下目标：

- 可靠性（Reliable）——软件系统对于用户的商业经营和管理来说极为重要，因此软件系统必须非常可靠。
- 安全性（Secure）——软件系统所承担的交易商业价值极高，系统的安全性非常重要。
- 可定制化（Customizable）——同样的一套软件，可以根据客户群的不同和市场需求的变化进行调整。
- 可扩展性（Extensible）——软件必须能够在用户的使用率、用户的数目增加很

快的情况下，保持合理的性能。只有这样，才能适应用户的市场扩展的可能性；在新技术出现的时候，一个软件系统应当允许导入新技术，从而对现有系统进行功能和性能的扩展。

- 可维护性（Maintainable）——软件系统的维护包括两方面：一是排除现有的错误；二是将新的软件需求反映到现有系统中去。一个易于维护的系统可以有效地降低技术支持的花费。
- 客户体验（Customer Experience）——软件系统必须易于使用。
- 市场时机（Time to Market）——软件用户要面临同业竞争，软件提供商也要面临同业竞争。以最快的速度争夺市场先机非常重要。

2. 框架

框架，即 framework。框架是在给定问题领域范围内建立起可重用解决设计问题而互相间协作的一组类的集合（它典型包括了具备缺省行为的大量对象），然而框架不仅仅是行为的集合，还包括了支配着把这些行为组合在一起的方法的一组规则，或称为协议。框架包含了一组相互关联类一起工作的方法，框架在一个给定领域范围中，解决了大量具体问题的通用的设计。框架其实就是某种应用的半成品，就是一组组件，供开发人员选用完成自己的系统。简单说就是使用别人搭好的舞台，自己来做表演。

框架一般处在低层应用平台（如 J2EE）和高层业务逻辑之间的中间层。框架一般都是遵循好莱坞原则设计的，否则就不叫框架。所谓好莱坞原则，说的是 You don't call us-we will call you. 意思就是在一个框架下的代码，都是被动地被框架调用，而不是相反（如图 1-1 所示）。

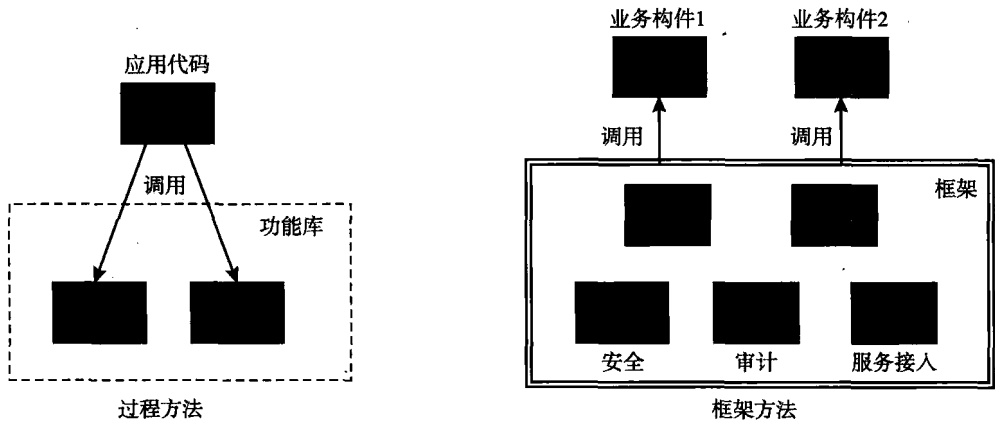


图 1-1 好莱坞原则

通过这种方式，大量重复的代码就可以隐藏在框架里面，需要特别设计的代码以预定接口的方式交给开发人员，写好后由框架调用。譬如 JSP 就是一个框架。JSP 脚本会被 JSP 引擎编译成 servlet 的一部分。这种 servlet 都带有大量重复的代码，是 JSP 程序员不需要考虑的，由框架负责。框架的主要优点有：减少编程的代码、增加代码的可

靠性和健壮性、更能保证一致性和模块化、提供了通用领域的问题（如用户接口，图形界面或网络操作等等）服务。

因为软件系统发展到今天已经很复杂了，特别是服务器端软件，涉及到的知识、内容，问题太多。在某些方面使用现有成熟的框架，就相当于让别人帮助完成一些基础工作，开发人员只需要集中精力完成系统的业务逻辑设计。而且框架一般是成熟、稳健的，并且可以处理系统很多细节问题，比如，事物处理，安全性，数据流控制等问题。还有框架一般都经过很多人使用，所以结构很好，扩展性也很好，而且它是不断升级的，开发人员可以直接享受不断升级代码带来的好处。

3. 设计模式

在说明设计模式之前，需要先了解一下什么是“模式”。

模式（Pattern）的概念最早由建筑大师 Christopher Alexander 于 20 世纪 70 年代提出，应用于建筑领域，80 年代中期由 Ward Cunningham 和 Kent Beck 将其思想引入到软件领域，Christopher Alexander 将模式分为三个部分：首先是周境（Context，也可以称上下文），指模式在何种状况下发生作用；其二是动机（System of Forces），意指问题或预期的目标；其三是解决方案（Solution），指平衡各动机或解决所阐述问题的一个构造或配置（Configuration）。他提出，模式是表示周境、动机、解决方案三个方面关系的一个规则，每个模式描述了一个在某种周境下不断重复发生的问题，以及该问题解决方案的核心所在，模式既是一个事物（thing）又是一个过程（process），不仅描述该事物本身，而且提出了通过怎样的过程来产生该事物。这一定义已被软件界广为接受。

在不同的层面上，模式提供不同层面的指导。根据处理问题的粒度不同，从高到低，模式分为 3 个层次：架构模式（Architectural Pattern）、设计模式（Design Pattern）、实现模式（Implementation Pattern）。架构模式是模式中的最高层次，描述软件系统里的基本的结构组织或纲要，通常提供一组事先定义好的子系统，指定它们的责任，并给出把它们组织在一起的法则和指南。比如，用户和文件系统安全策略模型，N-层结构，组件对象服务等，大家都熟知的 MVC 结构也属于架构模式的层次。一个架构模式常常可以分解成很多个设计模式的联合使用。设计模式是模式中的第二层次，用来处理程序设计中反复出现的问题。例如，[GOF95] [2] 总结的 23 个基本设计模式——Factory Pattern，Observer Pattern 等等。实现模式是最低也是最具体的层次，处理具体到编程语言的问题。比如，类名，变量名，函数名的命名规则；异常处理的规则等等。

设计模式的应用对软件开发产生了重大的作用。设计模式是人们在长期的设计软件、管理组织软件开发等实践中大量经验的提炼和抽象，是复用软件设计方法、过程管理经验的有力工具。设计模式类似于拳击中的组合拳，它提供了一系列软件开发中的思维套路。如，通过设计模式的使用，有利于在复杂的系统中产生简洁、精巧的设计。设计模式为设计开发人员提供了一套简洁通用的设计、管理、组织方面的词汇，同时设计模式也为设计开发人员提供了一个描述抽象事物的规范标准，可大大促进软件开发过程

中人与人之间的交流，而软件开发中的交流是至关重要的，“软件项目失败的原因最终都可追溯到信息没有及时准确地传递到应该接收它的人”。

从前面的说明中可以看出软件的大尺度结构就是架构。一个软件不管好坏，都会有一个架构。软件架构中可以利用框架，也可以不利用框架。譬如前面提到 JSP 是一种框架，而其他的系统可以利用 JSP 框架，形成自己的架构。从另一个角度来看，如果用房屋作比喻，架构就是忽略掉细节的抽象建筑结构，在图纸可以看到，存在于人脑之中，不体现为房屋的某一个物理部分。框架是房屋的骨架，房屋的骨架是物理存在的。

架构和设计模式应该是一个属于相互涵盖的过程，但是总体来说 Architecture 更加关注的是所谓的 High-Level Design，而设计模式关注的重点在于通过经验提取的“准则或指导方案”在设计中的应用，因此在不同层面考虑问题的时候就形成了不同问题域上的 Pattern。设计模式的目标是，把共同问题中的不变部分和变化部分分离出来。不变的部分，就构成了设计模式，因此，设计模式是一个经验提取的“准则”，并且在一次一次的实践中得到验证，在不同的层次有不同的模式，小到语言实现（如 Singleton）大到架构。

对于熟悉架构设计的系统架构师而言，似乎可以这样来解释架构和模式之间的关系：架构是 High-Level Design，着眼于不同业务中共性的解决方案；而模式是 General Principle（通用原理）。

4. 参考资料

- 《软件体系结构（影印版）》，科学出版社，2004 年 1 月 1 日出版。Mary Shaw、David Garlan 合著，原文书名《Software Architecture: Perspectives on an Emerging Discipline》。

- GoF95，《设计模式——可复用面向对象软件的基础》，Erich Gamma、Richard Helm 等著，英文版本《Design Patterns: Elements of Reusable Object-Oriented Software》，这是设计模式领域的经典之作，它结合设计实例从面向对象的设计中精选出 23 个设计模式，总结了面向对象设计中最有价值的经验，并且用简洁可复用的形式表达出来。

- [Buschmann96] Buschmann, Frank, 《Pattern-Oriented Software Architecture》，John Wiley & Sons Ltd, 1996。中文版《面向模式的软件体系结构》，机械工业出版社，2003 年 1 月出版。

第二章 总体概述

1. 总体架构

按照国家税务总局金税三期的技术要求，车购税管理系统应用软件结构被划分为：客户层、交互层、应用层、业务领域层、资源访问层、资源层（EIS），各层遵循如下原则：

- 不得跨层调用，每一层都只与直接相邻的层进行通信。
- 上面各层都建立在下层的基础上，隐藏下层的信息并为上层提供服务。
- 各层要封装自己的实现，向前一层提供访问接口。
- 各层支持分布式的部署，即可部署于不同的容器实例中。

各层的基本定位：

- 客户层：系统最终用户的使用界面和设备。包括基于浏览器的瘦客户端和基于 GUI 的胖客户端应用。

- 交互层：用户和系统之间的交互管理，提供用户层的展现逻辑和对应用层的访问接口。也包括单点登录、会话管理、用户输入的逻辑校验等功能。

- 应用层：业务逻辑的接口，实现业务流程的控制，是业务领域层的服务接口。

- 业务领域层：根据业务需求进行抽象的业务对象模型，包括业务规则和逻辑处理的实现。

- 资源访问层：对系统的各种资源和外部系统统一的访问逻辑的实现。

- 资源层（EIS）：各种信息系统资源，例如：RDBMS、文件系统、原有系统、信息服务、邮件服务、交易服务中间件等。

车购税管理系统是基于中间件/应用服务器的，面向关键任务的，基于税务专网的分布式应用系统。采用三层式（N 层）体系结构，车购税征收机关用户可以通过税务专网进行业务处理。

车购税管理系统是基于中间件/应用服务器的，面向关键任务的，广域网络的分布式应用系统（如图 2-1 所示）。采用三层式（N 层）体系结构，互联网用户可以通过 Internet IP 地址或网站域名访问 WEB 服务器。

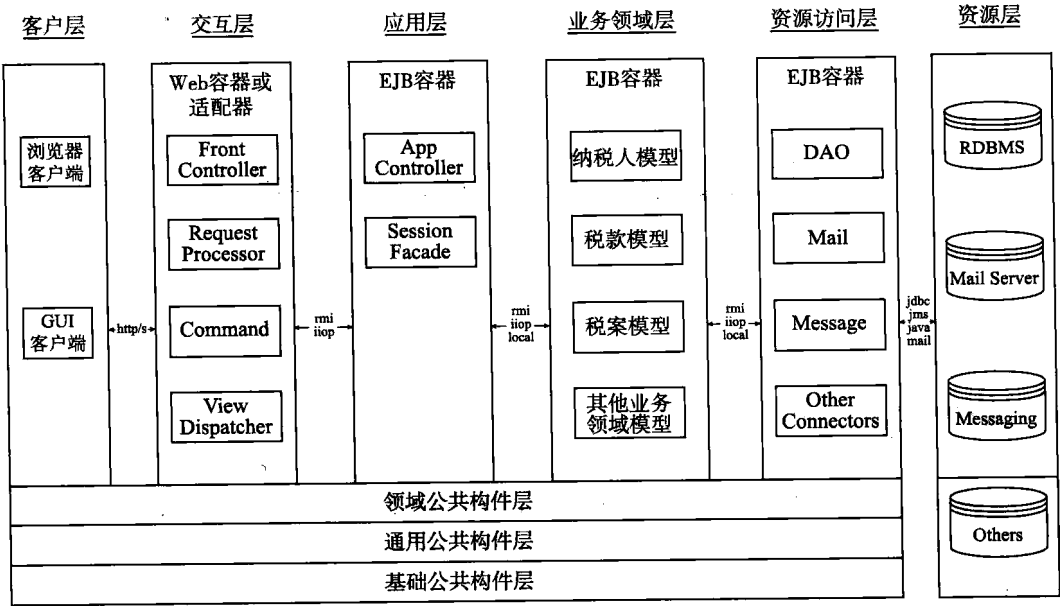


图 2-1 系统体系架构图

2. 架构实现

架构的实现上系统使用开源框架 Spring 作为主框架，整个应用系统使用两个框架上下文，分别由表示层的 Web 应用模块和 EJB 层的业务应用程序模块加载。

Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Spring 框架

Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式，如图 2-2 所示。

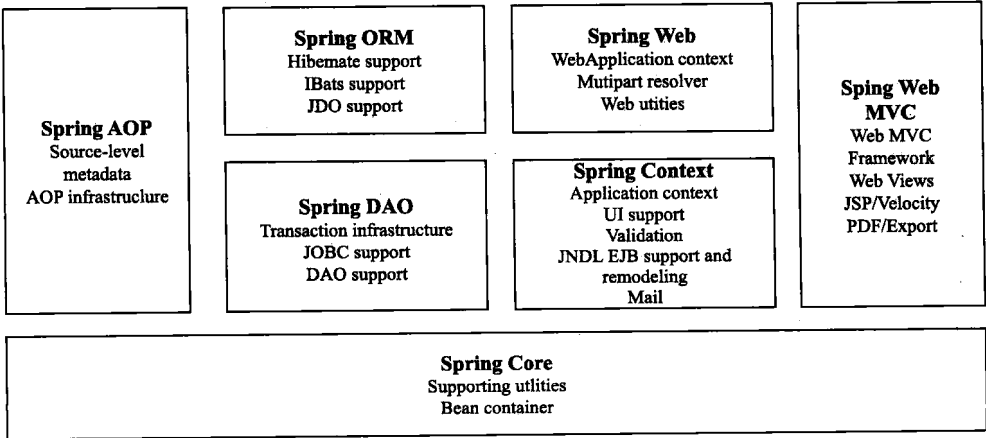


图 2-2 spring 框架模块结构