

宠物诊所项目概述



本章要点

- 理解项目需求
- 理解项目设计
- 掌握开发环境搭建
- 掌握 Eclipse 的使用

1.1 宠物诊所项目简介

如今，饲养宠物的热潮正在我国的都市里悄悄兴起，随着宠物队伍的日益壮大，一个新兴的产业——宠物产业，也正如冰山浮出水面。这一产业的出现是人们生活水平改善后消费层次提高的结果。从世界范围看，宠物饲养并不是一朝一夕或某个国家、地区的个别现象，世界上许多国家的各个阶层的人们都饲养宠物。

宠物诊所近年来随着饲养宠物的热潮在国内逐步兴起，宠物诊所的信息化建设还刚刚起步，对于其管理主要还处于摸索阶段，市场也迫切需要一套规范化的管理软件去管理，从而提升宠物诊所的管理水平。

本书中使用的社区宠物诊所系统是经过简化后的宠物诊所管理系统，系统仅保留了最基本的数据管理功能，为的是让学习者将重点放在如何使用 Java Web 的知识实现业务功能，而不是去理解复杂的业务需求。

社区宠物诊所系统的初始版本必须实现以下功能：

- 能够对医生信息进行查询;
- 能够添加新的医生信息 (姓名、专业);
- 能够对客户信息进行查询;
- 能够查看客户的详细信息;
- 能够添加新的客户信息 (姓名、电话、地址);
- 能够添加新的宠物信息 (姓名、生日、照片);
- 能够删除宠物信息;
- 能够添加宠物的病历信息 (病情描述、治疗方案、问诊时间);
- 能够浏览宠物的病历信息。

除此以外, 诊所的职员在使用系统提供的上述功能之前需要进行登录。当职员不需要使用系统的上述功能时, 也可退出系统。

社区宠物诊所系统的初始版本功能全部是以诊所职员为主, 考虑到将来需要为客户提供一个在线交流的问诊平台, 系统还需要预留客户身份登录功能, 并且对页面加入访问控制权限验证。

1.2 系统分析与总体设计

1.2.1 功能需求分析

需求分析需要对每个功能进行详细的描述, 需求分析是以正确可行等标准对系统进行完整的需求说明。图 1-1 是社区宠物诊所系统的用例图。

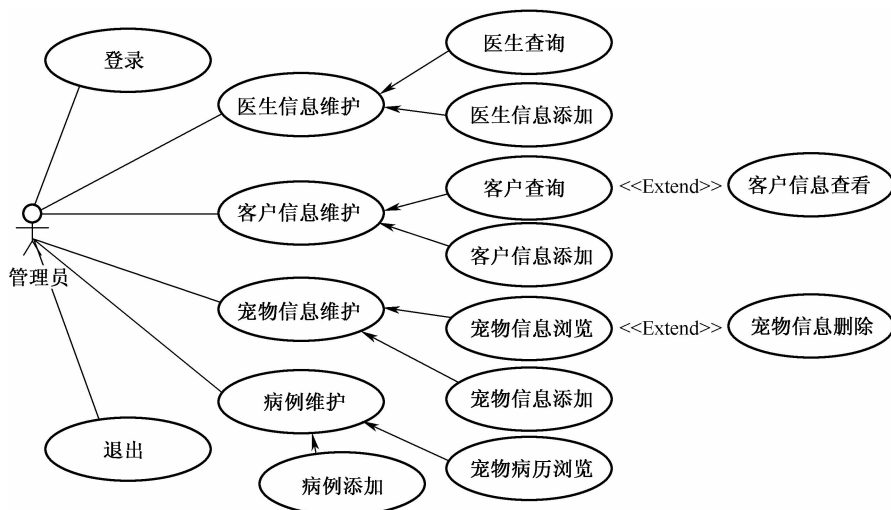


图 1-1 社区宠物诊所系统用例图

这里将系统的每个最基本的有价值的业务功能，如登录、医生信息维护等，称为用例。用例图中，使用一个椭圆表示用例，里面的文字描述了用例的名称。在图中使用一个“火柴人”表示管理员，称为用例的参与者，系统目前只有一个参与者。

为了读者学习方便，将用例图的详细说明分散到各个功能实现章节。

1.2.2 系统功能结构

社区宠物诊所系统的功能结构图如图 1-2 所示。

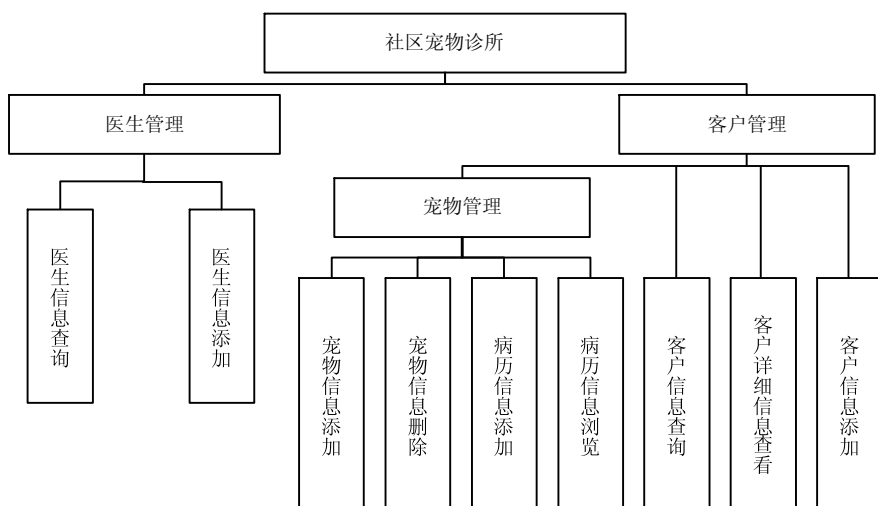


图 1-2 社区宠物诊所系统功能结构图

1.3 系统架构设计

1.3.1 实体模型设计

从需求中可以得出系统的如下关键抽象：兽医、专业特长、客户、宠物和宠物的访问等。这些实体可以设计为 `JavaBean` 类，例如宠物具有这些属性：名称、标识 ID、主人 ID、照片和出生日期等。宠物主人和宠物之间具有如下关系：一个宠物主人可以拥有多个宠物，每一个宠物属于一个主人；每一个宠物可能有到诊所多次就诊的病历；每个兽医具有多项专长，而同一专长可能有多个兽医。图 1-3 描述了系统的关键抽象，它们为系统中使用的实体类。

在设计用户类时，将管理员角色和客户角色合并成为一个实体类，通过 `role` 属性进行区分，如果 `role` 的值为 `admin` 表示管理员，如果 `role` 的值为 `customer` 表示客户。

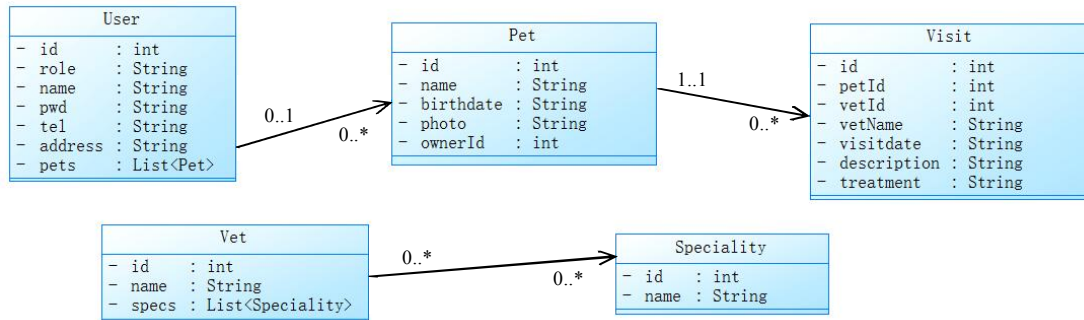


图 1-3 实体类图

通过类图可以编写系统的实体类代码，例如：

【User.java】

```

package ph.entity;

import java.util.ArrayList;
import java.util.List;

public class User {
    private int id;
    private String role;
    private String name;
    private String pwd;
    private String tel;
    private String address;
    private List<Pet> pets=new ArrayList<Pet>();
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPwd() {
        return pwd;
    }

```

```
    }
    public void setPwd(String pwd) {
        this.pwd = pwd;
    }
    public String getTel() {
        return tel;
    }
    public void setTel(String tel) {
        this.tel = tel;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public List<Pet> getPets() {
        return pets;
    }
    public void setPets(List<Pet> pets) {
        this.pets = pets;
    }
}
```

1.3.2 数据库设计

根据实体模型设计得到数据库设计如表 1-1 至表 1-6 所示。

表 1-1 用户表 (t_user)

字段	类型	允许为空	约束	其他	说明
id	int	NO	PRI	auto_increment	主键
role	varchar(8)	NO			角色
name	varchar(32)	NO			用户名
pwd	varchar(32)	NO			密码
tel	varchar(16)	YES			电话
address	varchar(255)	YES			地址

表 1-2 医生表 (t_vet)

字段	类型	允许为空	约束	其他	说明
id	int	NO	PRI	auto_increment	主键
name	varchar(32)	NO			医生名

表 1-3 专业表 (t_speciality)

字段	类型	允许为空	约束	其他	说明
id	int	NO	PRI	auto_increment	主键
name	varchar(32)	NO			专业名

表 1-4 医生专业关系表 (t_vet_speciality)

字段	类型	允许为空	约束	其他	说明
vetId	int	NO	FK		医生主键
specId	int	NO	FK		专业主键

表 1-5 宠物表 (t_pet)

字段	类型	允许为空	约束	其他	说明
id	int	NO	PRI	auto_increment	主键
name	varchar(32)	NO			宠物名
birthdate	varchar(16)	NO			生日
photo	varchar(64)	NO			照片
ownerId	int	NO	FK		主人主键

表 1-6 病历表 (t_visit)

字段	类型	允许为空	约束	其他	说明
id	int	NO	PRI	auto_increment	主键
petId	int	NO	FK		宠物主键
vetId	int	NO			医生主键
visitdate	varchar(10)	NO			日期
description	text	NO			病情描述
treatment	text	NO			治疗方案

1.3.3 业务逻辑设计

在社区宠物诊所系统中，通过 DAO 模式实现对数据访问的封装。DAO 模式是 Java 编程的一种常用模式，其主要思想是在业务处理逻辑和数据库之间增加一层数据访问代码。通过数据访问代码的连接可以实现业务处理核心代码和底层数据库之间的分离，降低耦合性。

在实际开发中，数据库的类型有很多种，比如常见的关系型数据库 SQL Server、Oracle 或者 MySQL。通过使用 DAO 模式，能够将数据库的实现细节完全封装在 DAO 代码中，业务代码在调用 DAO 的方法时完全不用关心底层数据库到底是何种类型，而只需要将注意力放在

如何实现业务逻辑，需要访问数据库时调用 DAO 即可，如图 1-4 所示。

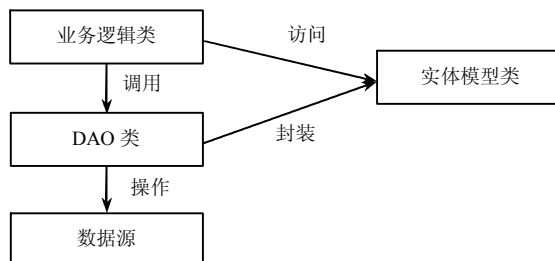


图 1-4 DAO 模式

1.4 开发环境搭建

1.4.1 安装 JDK

Java 目前被 Oracle 公司收购，下载 JDK 需要前往 <http://www.oracle.com>。在 Oracle 公司的官网首页点击 Downloads 下的 Java SE 菜单，如图 1-5 所示。

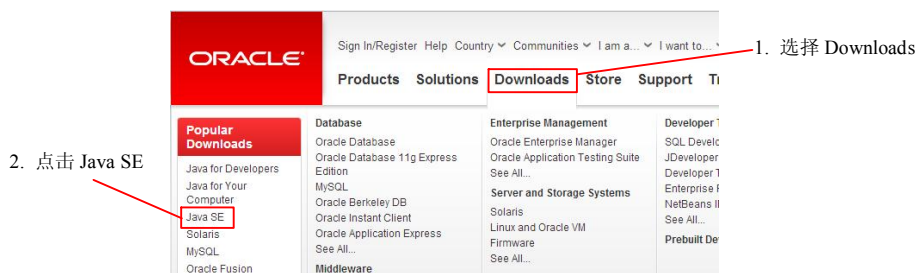


图 1-5 打开 Oracle 官网

在打开的页面中找到 JavaSe7 的下载，点击 JDK DOWNLOAD 按钮，如图 1-6 所示。



图 1-6 选择下载 JavaSe7

在打开的 JavaSe7 Development Kit 下载页面中首先选择接受协议, 然后根据个人操作系统选择合适的版本下载, 其中 x86 是基于 32 位操作系统, x64 是基于 64 位操作系统, 如图 1-7、图 1-8 所示。



图 1-7 选择接受协议

Product / File Description	File Size	Download
Linux x86	119.44 MB	jdk-7u71-linux-i586.rpm
Linux x86	136.76 MB	jdk-7u71-linux-i586.tar.gz
Linux x64	120.81 MB	jdk-7u71-linux-x64.rpm
Linux x64	135.63 MB	jdk-7u71-linux-x64.tar.gz
Mac OS X x64	185.84 MB	jdk-7u71-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.36 MB	jdk-7u71-solaris-i586.tar.Z
Solaris x86	95.48 MB	jdk-7u71-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.68 MB	jdk-7u71-solaris-x64.tar.Z
Solaris x64	16.36 MB	jdk-7u71-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	138.74 MB	jdk-7u71-solaris-sparc.tar.Z
Solaris SPARC	98.62 MB	jdk-7u71-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u71-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.35 MB	jdk-7u71-solaris-sparcv9.tar.gz
Windows x86	127.78 MB	jdk-7u71-windows-i586.exe
Windows x64	129.52 MB	jdk-7u71-windows-x64.exe

根据操作系统
选择下载

图 1-8 选择对应的操作系统版本

下载完毕后运行安装程序, 如果不需要修改安装路径则可以点击【下一步】按钮直到安装结束, 如图 1-9 所示。



图 1-9 安装 JDK

安装结束后要对环境变量进行配置，打开【系统属性】对话框，点击【环境变量】按钮，如图 1-10 所示。

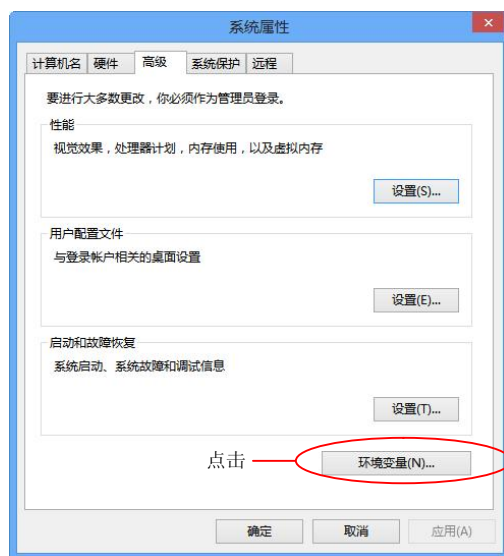


图 1-10 配置环境变量

在弹出的【环境变量】对话框中点击【新建】按钮，如图 1-11 所示。

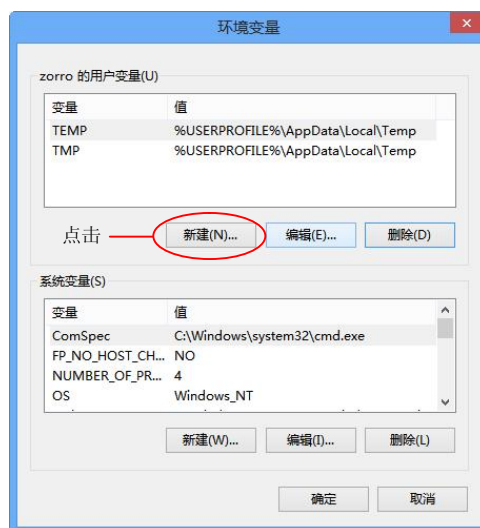


图 1-11 新建环境变量

在弹出的【新建用户变量】对话框中输入变量名 `JAVA_HOME`，变量值为实际的 JDK 安装路径，如 `C:\Program Files\Java\jdk1.7.0_71`。输入无误后点击【确定】按钮，如图 1-12 所示。

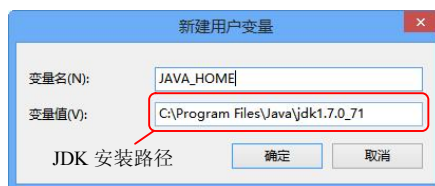


图 1-12 添加 JAVA_HOME 变量

1.4.2 安装 Tomcat

下载 Tomcat 文件，登录网址 <http://tomcat.apache.org/>，在打开的页面中点击 Download 菜单下的 Tomcat7.0 选项，如图 1-13 所示。

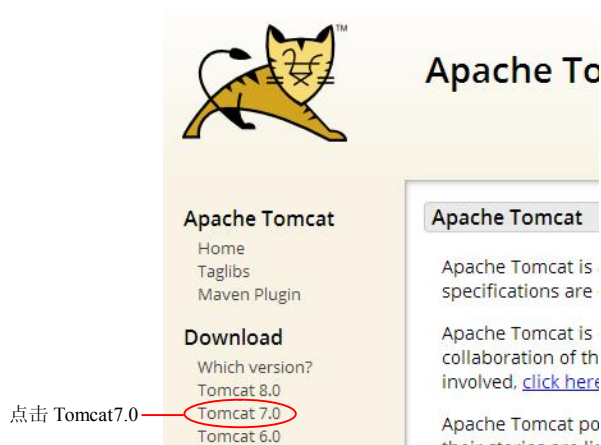


图 1-13 点击 Tomcat7.0 下载

在打开的下载页面中根据实际的操作系统选择合适的下载文件，Tomcat 下载文件分为解压版和安装版，解压版可以实现绿色安装，这里选择 32 位的解压版，如图 1-14 所示。

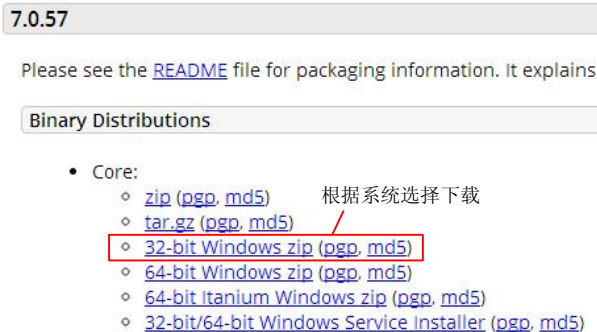


图 1-14 下载 Tomcat

下载完成后将 Tomcat 的内容解压到合适的目录下，解压完成后的结构如图 1-15 所示。

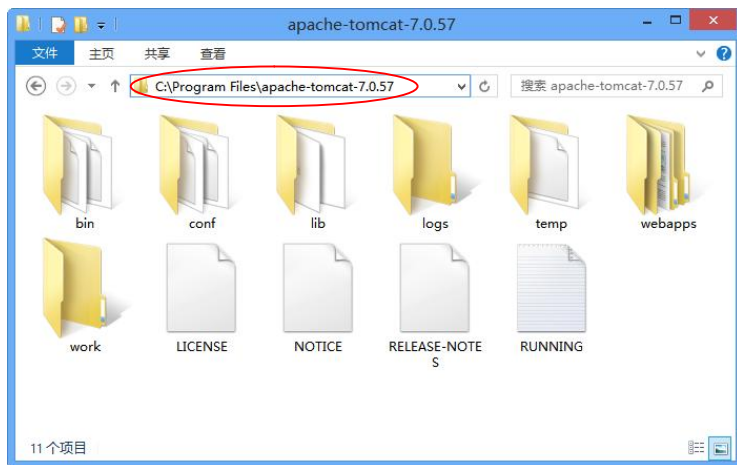


图 1-15 解压 Tomcat

如需单独运行 Tomcat 可以参考前面内容配置 TOMCAT_HOME 环境变量为 Tomcat 的解压路径，然后运行 bin 目录下的 startup.bat。

1.4.3 安装 MySQL

下载 MySQL 安装文件，登录网址 <http://www.mysql.com/downloads/mysql/>，会出现下载列表，如图 1-16 所示，根据个人操作系统选择相应的下载项，笔者下载的是 Windows(x86,32-bit)，MSI Installer。

MySQL Community Server 5.1.73

Select Version:

Select Platform:

Windows (x86, 32-bit), MSI Installer (mysql-5.1.73-win32.msi)	5.1.73	106.2M	Download
	MD5: cbb731380efc49ce6ff4aac22adb095 Signature		
Windows (x86, 64-bit), MSI Installer (mysql-5.1.73-winx64.msi)	5.1.73	99.4M	Download
	MD5: b218f3e8a893e9a6d94250512f47c88f Signature		
Windows (x86, 32-bit), MSI Installer Essentials - Recommended (mysql-essential-5.1.73-win32.msi)	5.1.73	38.8M	Download
	MD5: 6383f51f0871d11af27ef992ab244f00 Signature		

图 1-16 MySQL 下载列表

运行安装程序，弹出图 1-17 所示的安装类型选择界面，选择 Typical 安装。

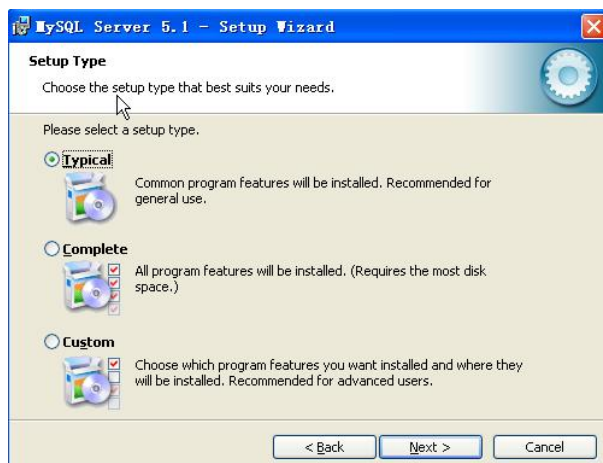


图 1-17 安装类型选择

安装结束后弹出图 1-18 所示的安装完毕界面，勾选 **Configure the MySQL Server now** 复选框后点击 **Finish** 按钮，打开 MySQL 配置界面。



图 1-18 安装完毕

配置界面采用默认设置直到出现图 1-19 所示的字符集选择界面，选择第二个选项将 MySQL 的字符集设置成 UTF-8 格式。

在弹出图 1-20 所示的密码设置界面时，需要输入数据库的访问密码并确认，这里笔者输入的是 123456。MySQL 的默认访问用户名是 root。

弹出图 1-21 所示的配置完毕界面时，如果没有出现错误信息表示配置完毕。

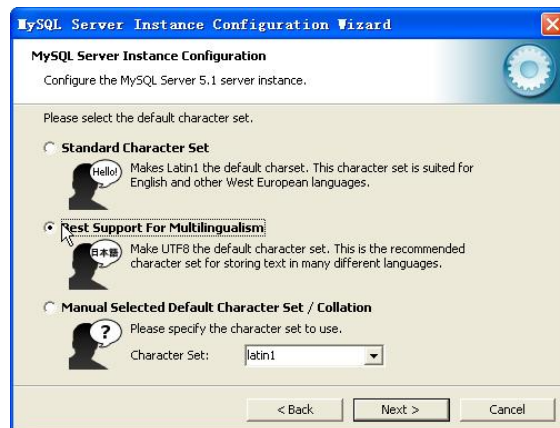


图 1-19 字符集选择



图 1-20 密码设置



图 1-21 配置完毕

1.4.4 安装 Eclipse

下载 Eclipse 文件, 登录 Eclipse 官网 <http://www.eclipse.org/>, 在首页点击右侧的 DOWNLOAD 图标, 根据操作系统, 在打开的页面中选择 Eclipse IDE for Java EE Developers 一栏后的下载链接, 如图 1-22 所示。

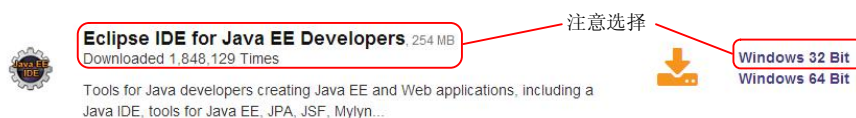


图 1-22 选择 Eclipse 下载版本

Eclipse 提供的是解压版下载, 下载完毕后将文件解压到合适的目录下, 如图 1-23 所示。

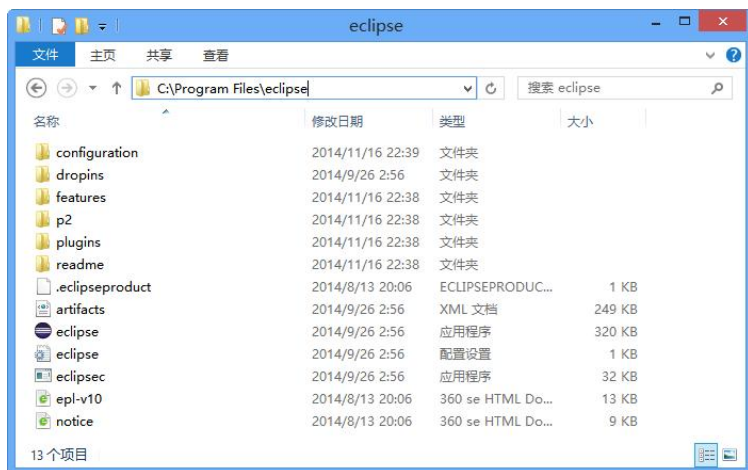


图 1-23 解压 Eclipse

执行 eclipse.exe 程序, 弹出 Workspace Launcher 对话框, 设置项目源代码的保存路径, 如图 1-24 所示。

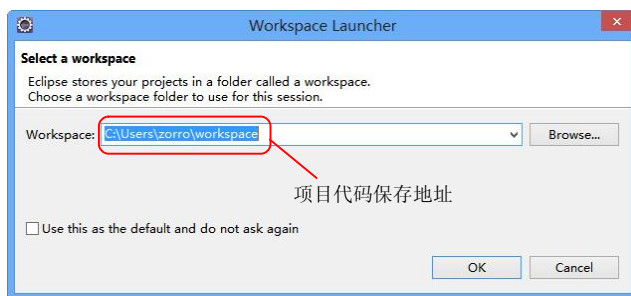


图 1-24 设置 Workspace

首次启动结束后需要设置 Eclipse 中的 Tomcat 服务器，点击 Window 菜单下的 Preferences 选项，如图 1-25 所示。

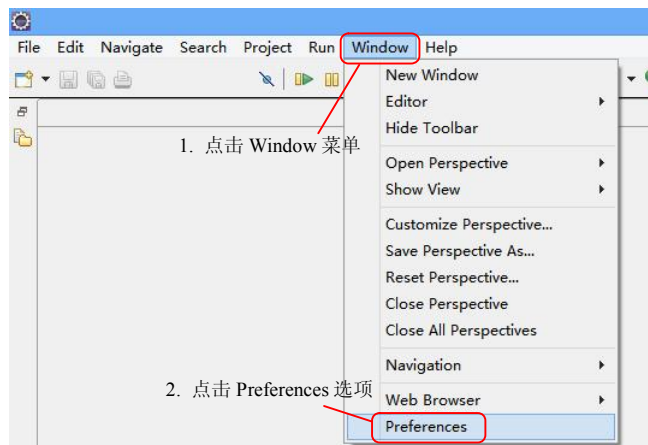


图 1-25 打开 Preferences

在打开的窗口左侧找到 Server→Runtime Environment 选项，然后点击右侧的 Add 按钮，如图 1-26 所示。

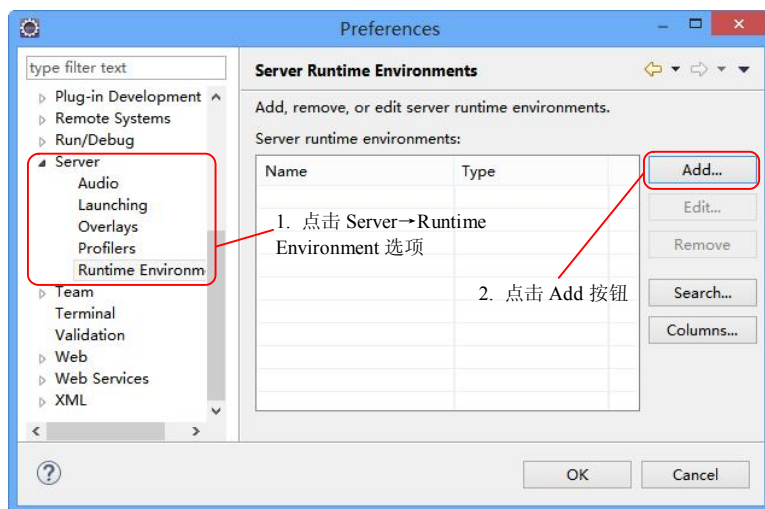


图 1-26 添加服务器设置

在打开的配置界面中选择 Apache Tomcat v7.0 选项点击 Next 按钮，如图 1-27 所示。

在配置界面中输入 Tomcat 的实际解压路径后点击 Finish 按钮，如图 1-28 所示，到此开发环境搭建完毕。

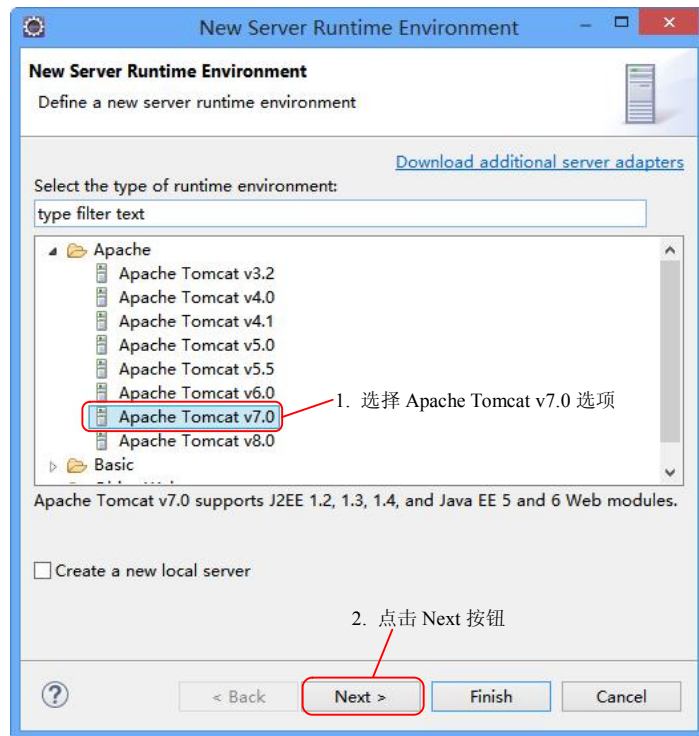


图 1-27 添加 Tomcat7 配置

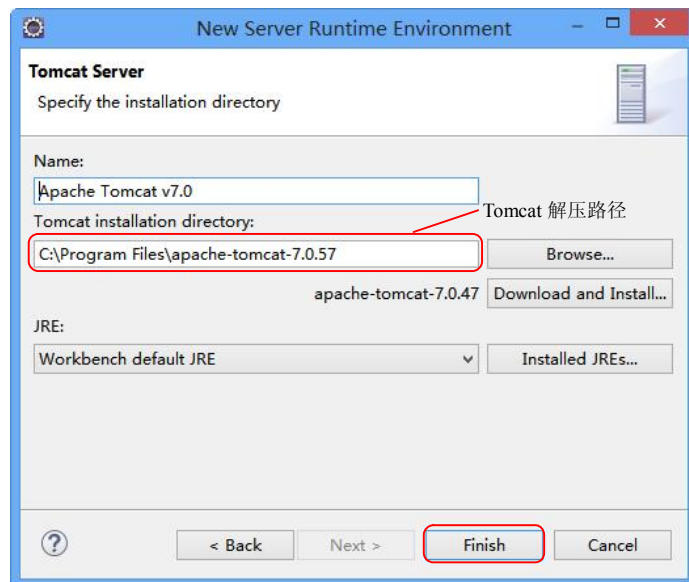


图 1-28 配置 Tomcat 路径

1.4.5 创建初始项目

使用 Eclipse 开发 Java Web 项目的步骤介绍如下。

点击 Eclipse 的【新建】按钮，在弹出的下拉列表中选择 Dynamic Web Project 选项，如图 1-29 所示。

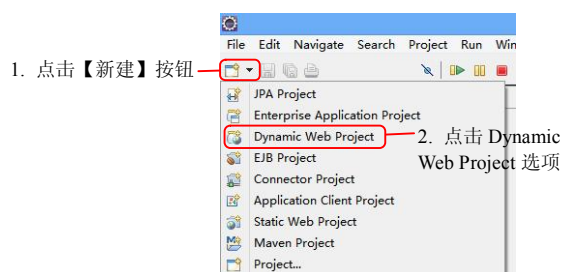


图 1-29 新建项目

在弹出的新建界面中输入项目名，选择目标服务器 Target runtime 为 Apache Tomcat v7.0 选项，因为在后面代码中使用了 Servlet3.0 的新特性注解配置和文件上传，故将动态网页技术版本 Dynamic web module version 选中为 3.0，点击 Next 按钮，如图 1-30 所示。

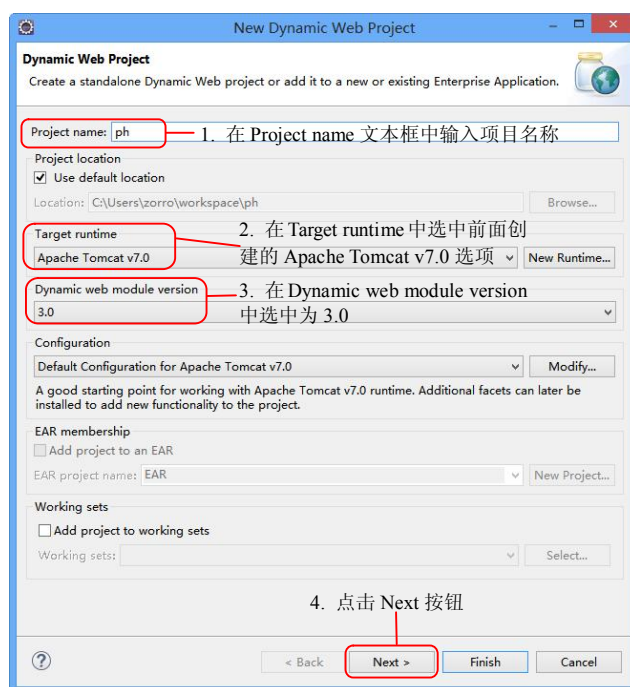


图 1-30 输入项目信息

后续内容不需要修改可以直接点击 Finish 按钮完成项目创建。

Eclipse 的项目 Java 文件放在 Java Resources 的 src 下，网页文件根目录在 WebContent 目录下，如图 1-31 所示。

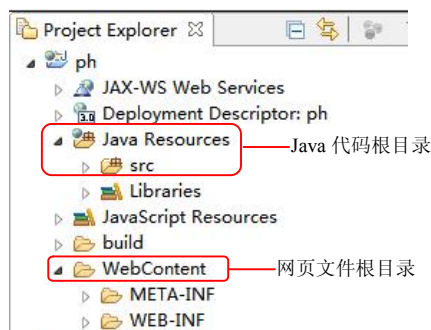


图 1-31 项目代码结构

在 WebContent 下创建 index.jsp 文件并保存，要想执行项目代码，需选中项目跟文件夹并右击，在弹出的快捷菜单中选择 Debug As 或者 Run As 选项，这两个选项的区别在于 Debug As 可以使用断点调试项目代码，点击 Debug/Run on Server 选项，如图 1-32 所示。

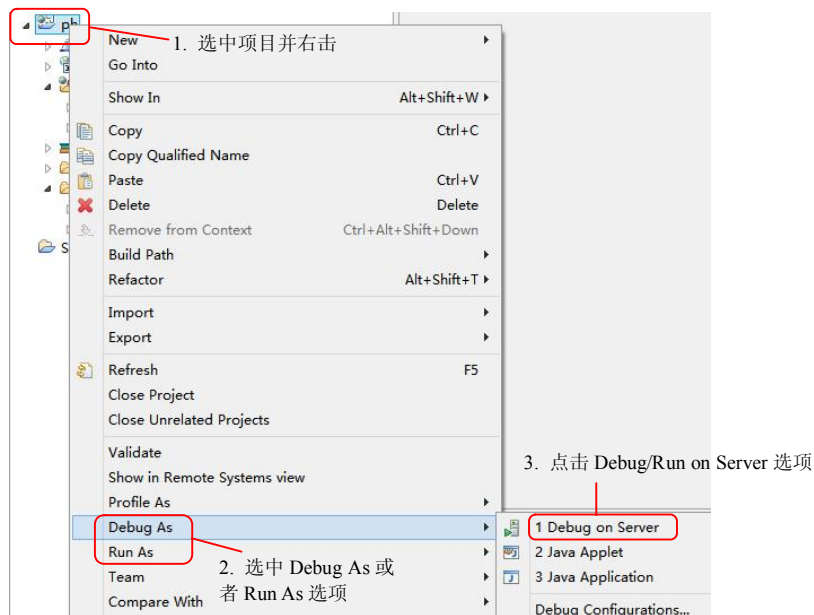


图 1-32 执行项目代码

在弹出的界面中选择 Tomcat7.0 并点击 Finish 按钮，如图 1-33 所示。如果出现如图 1-34 所示界面就表示项目执行成功。

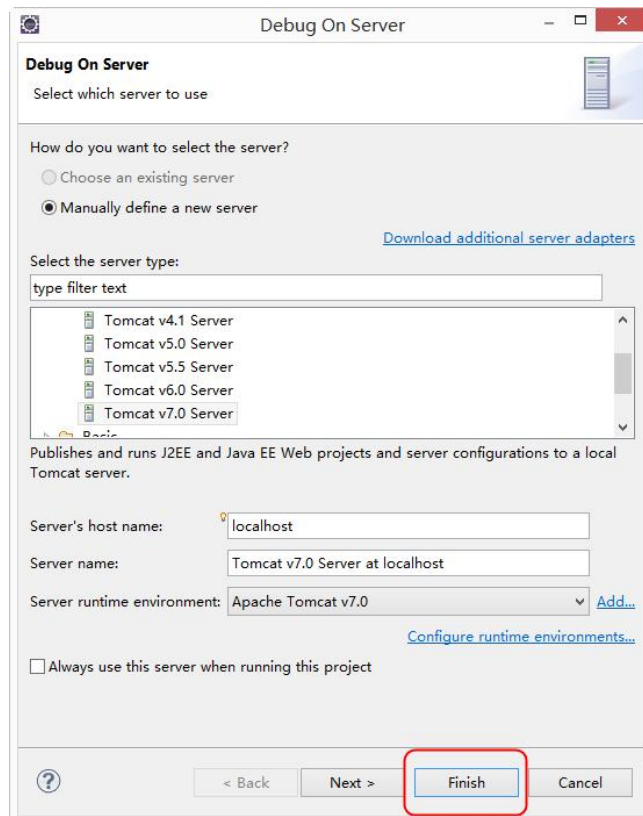


图 1-33 调试执行项目代码

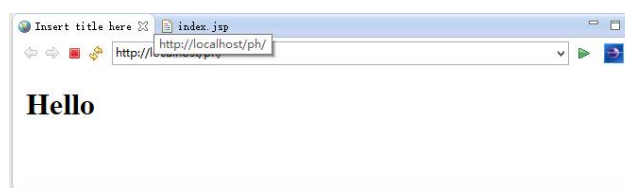


图 1-34 执行成功

任务拓展

1. 搭建社区宠物诊所系统所需要的开发环境。
2. 创建社区宠物诊所系统的数据库及表结构。
3. 创建初始项目代码。
4. 根据图 1-3 完成系统实体类代码。