

软件构造 实验指导书

太原科技大学
计算机学院

实验 1 软件构造的前期准备工作

一. 实验目的

通过实际案例使学生理解软件构造的前期准备工作的重要性，以及如何验证前期准备工作，理解迭代式和序列式软件开发的使用场景。

二. 实验内容

1. 打开《电厂职工培训管理系统》的原型系统，理解该软件的功能需求。

2. 根据该系统的功能需求，撰写一个简要的需求规格说明书，并参照教材中的需求核对表，检查需求准备工作是否完成。

3. 使用 Power Designer 工具菜单中的逆向工程，创建本系统的物理模型，结合软件的需求，对表以及其中的字段加上中文注释，并建立表之间的关系。通过此项工作加深对需求的理解。

4. 将创建的好的 PDM 物理模型，导入到 MySQL 数据库中，为下一步开发做好准备。

三. 实验过程及结果

1、以附件形式打印本系统的需求规格说明书

2、以附件形式打印系统的数据库模型

3、提出你对给出的数据模型的意见、建议及相应的理由。

四. 实验遇到问题及体会

实验 2 构造中的设计-设计可以工作的类

一. 实验目的

通过实验 1 对《电厂职工培训管理系统》的需求进行梳理，本实验结合系统实际需求，运用软件构造的多种启发式方法，如面向对象方法、信息隐藏、形成一致的抽象以及封装，并使用迭代、分而治之以及自上而下等多种设计实践，设计《电厂职工培训管理系统》的类结构。

通过本次实验使学生充分和理解软件技术的首要使用是管理复杂度，软件设计的特点，好的设计都是经过反复迭代，多次演化逐渐形成的。

二. 实验内容

1. 给出构造本系统的所使用的规范，包括类命名、方法命名及属性命名等方面。
2. 给出系统类结构图。
3. 根据 20/80 原则，对主要类的结构进行描述，包括每个类对外提供的接口（public method）及类之间的关系等方面。
4. 根据良好类设计的原则（良好的接口、一致抽象层次、封装等），检查设计结果是否符合要求。

三. 实验过程及结果

- 1、说明类设计的结果，以及选择此组织结构的理由，并以附件形式打印本系统的类结构图

四. 实验遇到问题及体会

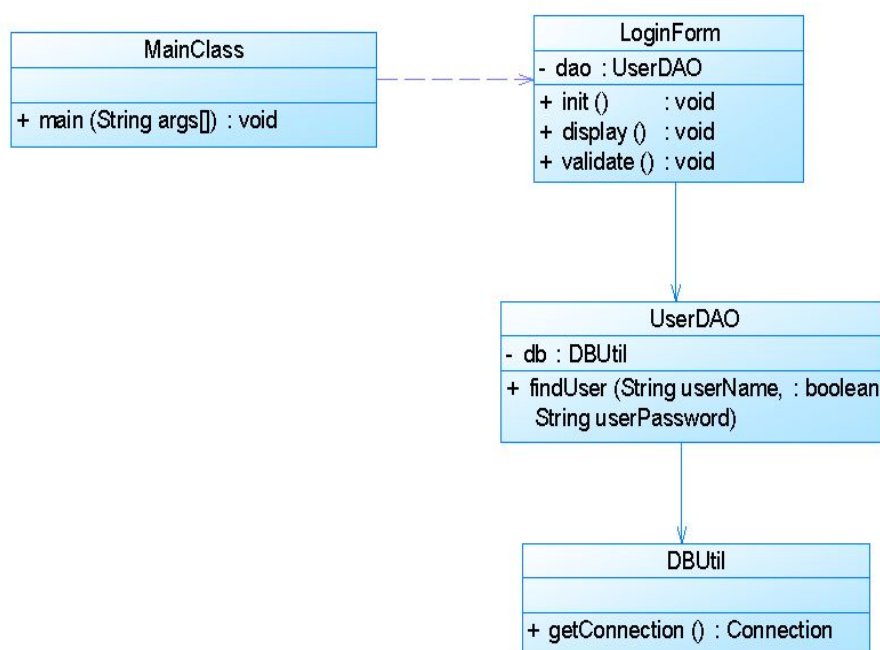
你的类设计结果是一气呵成的吗？对什么地方进行了修改，理由如何？

实验 3 系统结构设计及用户登录实现

一. 实验目的

1. 通过本次实验使学生理解掌握常规软件的层次化设计方法，通常将软件分解为视图层、业务层、持久层。其中视图层负责数据的输入和输出；业务层负责业务逻辑的实现；持久层负责数据库的增删改查等访问操作；

2. 实现用户登录功能，可按照如下类图实现该功能。LoginForm 类负责登录界面的初始化、显示、数据性验证功能的实现；UserDAO（用户数据访问对象）负责实现判断用户的合法性检验；DBUtil 类是一个工具类，负责 JDBC 连接的创建、关闭以及查询的执行等数据库操作。



一. 实验内容

1. 在 eclipse 环境下安装 windowbuilder 插件。该插件可以可视化和所见即所得方式用 swing 组件开发软件。具体安装过程参考[附件一、附件二](#)

一、附件二

2. 用安装好的插件开发本系统的登录界面（参考原型），并实现相应功能。

二. 实验过程及结果

1. 以图片和适当文字说明形式，写出插件安装过程。
2. 打印登录功能源代码，需要注意代码风格（注释、命名等方面）。

三. 实验遇到问题及体会

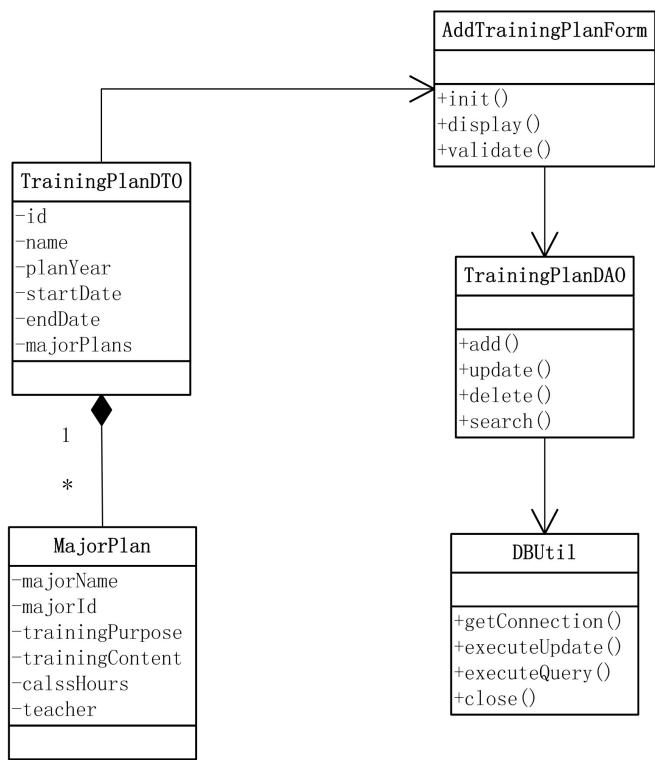
实验 4 添加培训计划

一. 实验目的

实现培训计划的增加功能，通过该功能的实现使学生理解类设计、特别是方法设计、变量命名等软件构造的理论知识。

二. 实验内容

遵循 DAO 设计模式（[参照附件三](#)），并按照如下图方式实现培训计划的添加功能。



AddTrainingPlanForm 类：表示培训计划的添加界面		
1	init	实现界面的初始化
2	display	负责界面的显示
3	Validate	验证用户输入的培训计划数据的合法性

TrainingPlanDTO 类：数据传输对象，表示一个培训计划实体		
-------------------------------------	--	--

1	id	编号
2	name	培训计划名称
3	planYear	年度
	startDate	开始时间
	endDate	结束时间
	majorPlane	各专业培训计划内容

MajorPlan 类：数据传输对象，表示一个专业的培训计划		
1	majorName	专业名称
2	majorId	专业编号
3	trainingPurpose	目的
	trainingContent	内容
	classHours	课时
	Teacher	培训教师

TrainingPlanDAO 类：数据访问对象，用于执行对培训计划的增删改查的操作		
1	add	添加
2	update	修改
3	delete	删除
4	Search	查询

DBUtil 类：数据库工具类，实现数据连接的创建、关闭、SQL 语句执行操作		
1	getConnection	创建连接
2	executeUpdate	执行 insert、update 语句
3	executeQuery	执行 select 语句
4	close	关闭连接

三. 实验过程及结果

1、按照类图以及类的说明实现功能，需要注意代码风格（注释、命名等方面）。

2、打印该功能的源代码，以及测试结果；

四. 实验遇到问题及体会（必须要写）

1、在本次实验中你应用到哪些软件构造的知识，并举例说明。

2、在实现过程遇到了什么问题，以及如何解决的？

3、讲讲本次实验的收获。

附件：

一、WindowBuilder 离线安装过程

WindowBuilder 是可视化 Java GUI 编程的 eclipse 插件。有了它的帮助，我们可以通过拖拽来编辑 Java 程序界面。安装过程如下：

1. 下载 WindowBuilder

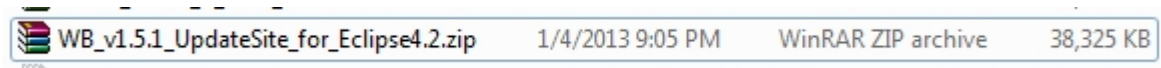
<http://eclipse.org/windowbuilder/download.php>

在上面网站选择与自己 eclipse 版本相符的 WindowBuilder. 如下图所示：

Update Sites

Eclipse Version	Release Version		Integration Version	
	Update Site	Zipped Update Site	Update Site	Zipped Update Site
4.3 (Kepler)			link	link (MD5 Hash)
4.2 (Juno)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.8 (Juno)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.7 (Indigo)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.6 (Helios)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.5 (Galileo)	link	link (MD5 Hash)	link	link (MD5 Hash)
3.4 (Ganymede)	link	link (MD5 Hash)		

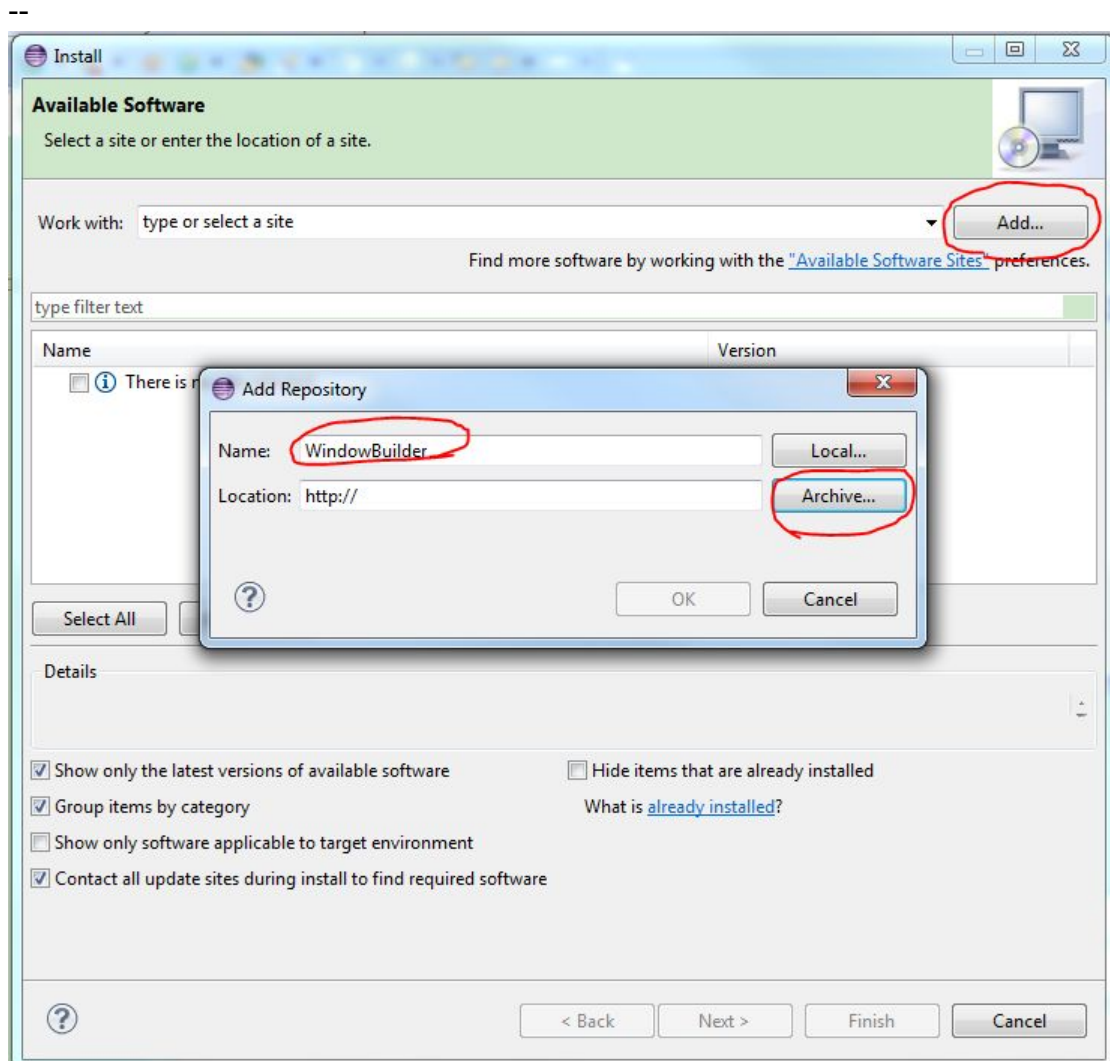
我安装的是 eclipse4.2 所以下载红笔所示的版本。下载后的.zip 文件如下图：



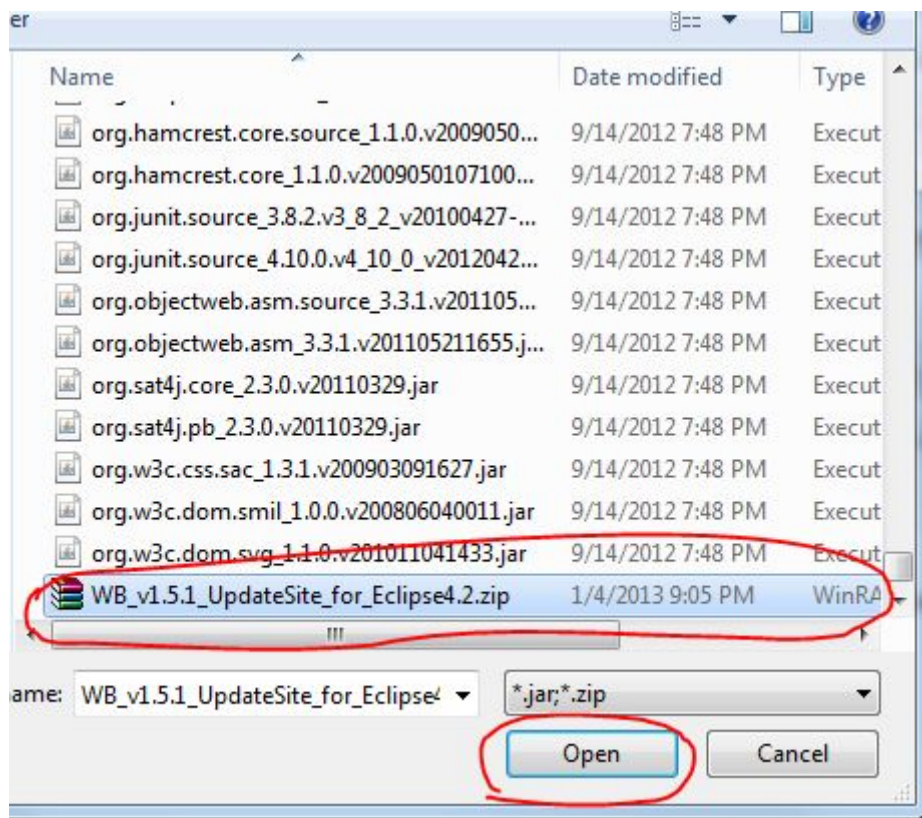
2. 安装 WindowBuilder

1) 把上图所示的.zip 文件拷贝到 eclipse 安装目录下 plugins 文件夹里。注意：是把整个 zip 文件，不需要解压。

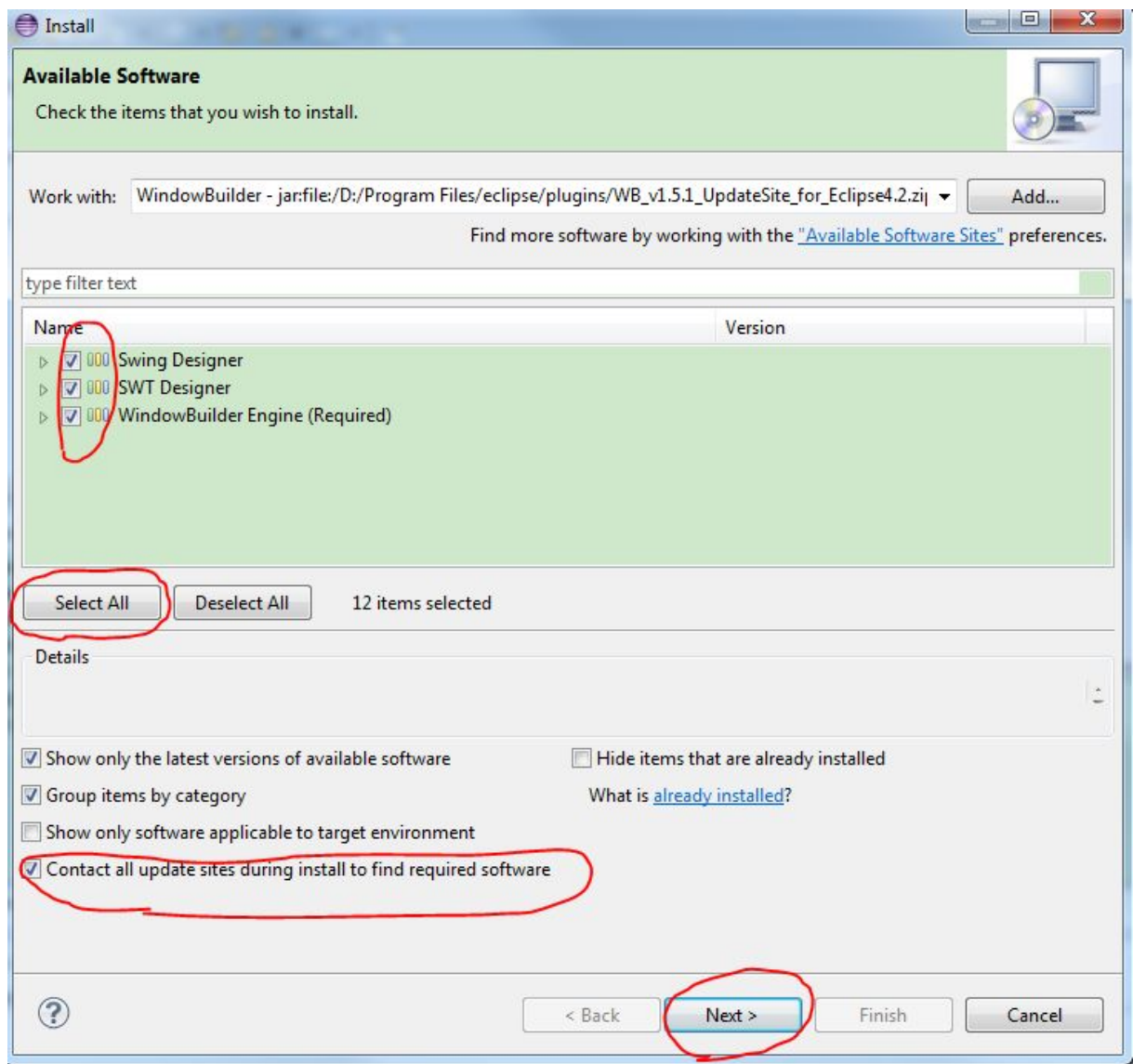
2) 打开 eclipse -- Help -- Install New Software... -- Add... -- Archive...



在弹出的对话框中选择之前放进 plugins 文件夹的 zip 文件，Open

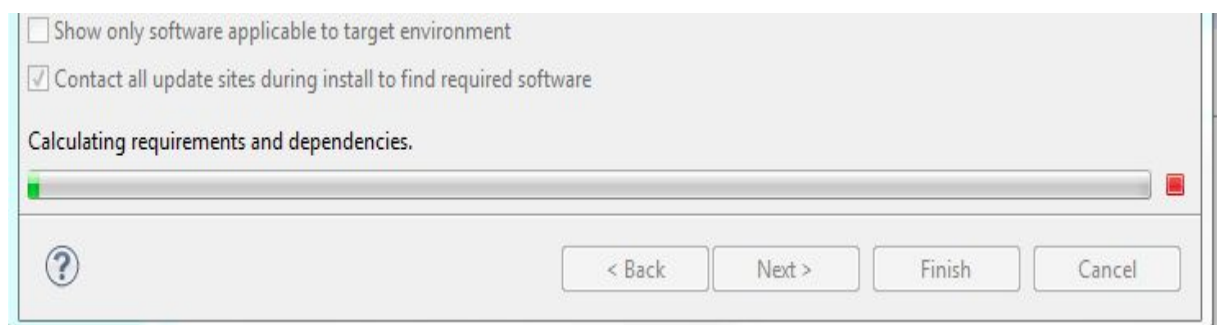


然后点 OK，会看到



选择 Select All, 然后点击 Next.

这时会看见下面的画面, 需要一些耐心来等待 完成。



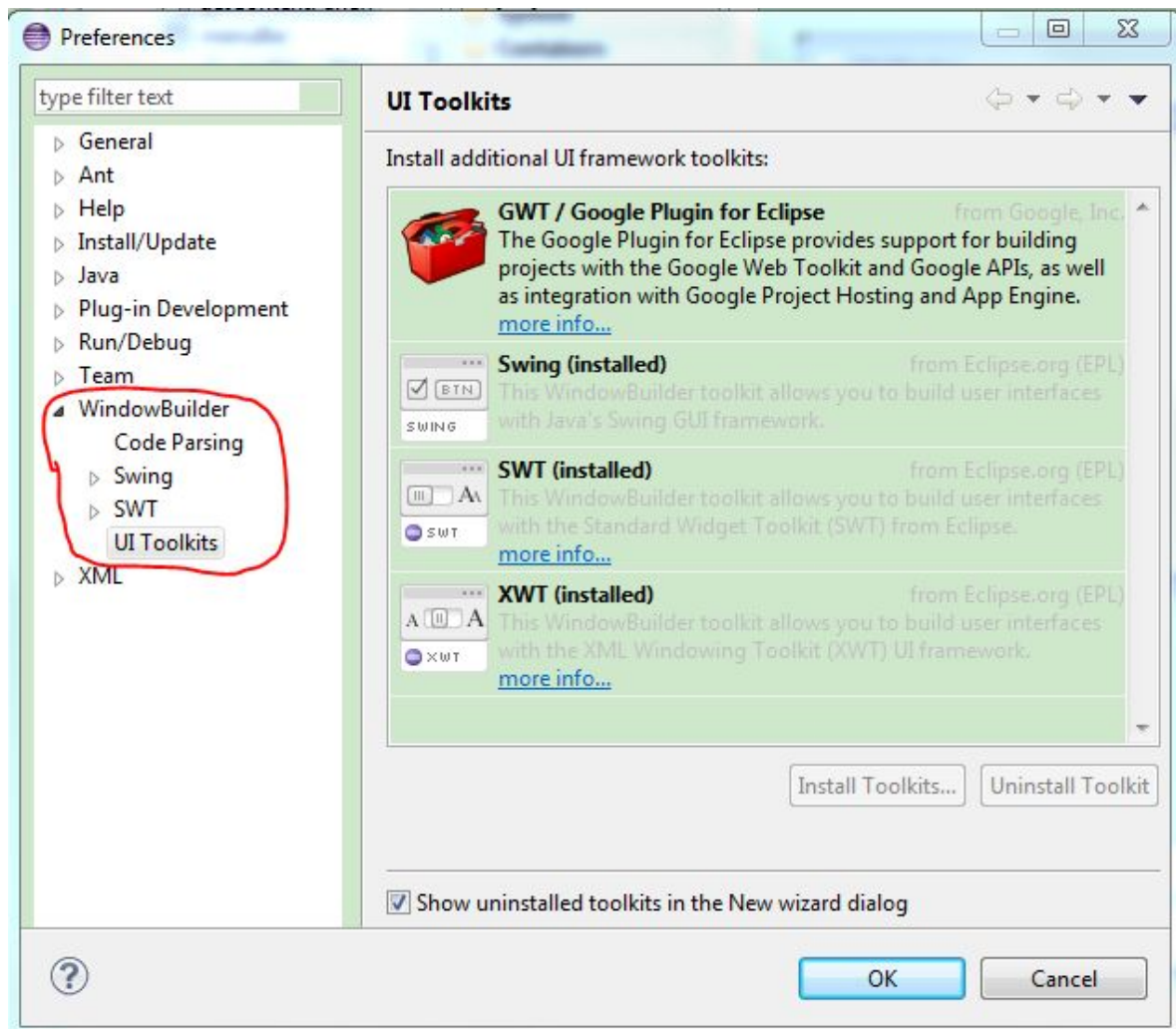
等下载完所需文件之后, 会看到这样的情况

Name	Version	Id
<input checked="" type="checkbox"/> Swing Designer	1.5.1.r42x20120916...	org.eclipse.wb.swing.feature.feature.group
<input checked="" type="checkbox"/> Swing Designer Documentation	1.5.1.r42x20120916...	org.eclipse.wb.swing.doc.user.feature.featu..
<input checked="" type="checkbox"/> SWT Designer	1.5.1.r42x20120916...	org.eclipse.wb.rcp.feature.feature.group
<input checked="" type="checkbox"/> SWT Designer Core	1.5.1.r42x20120916...	org.eclipse.wb.swt.feature.feature.group
<input checked="" type="checkbox"/> SWT Designer Documentation	1.5.1.r42x20120916...	org.eclipse.wb.rcp.doc.user.feature.featu..
<input checked="" type="checkbox"/> SWT Designer SWT_AWT Support	1.5.1.r42x20120916...	org.eclipse.wb.rcp.SWT_AWT_support.featu..
<input checked="" type="checkbox"/> SWT Designer XWT Support (requires Eclipse WTP/WST)	1.5.1.r42x20120916...	org.eclipse.wb.xwt.feature.feature.group
<input checked="" type="checkbox"/> WindowBuilder Core	1.5.1.r42x20120916...	org.eclipse.wb.core.feature.feature.group
<input checked="" type="checkbox"/> WindowBuilder Core Documentation	1.5.1.r42x20120916...	org.eclipse.wb.doc.user.feature.feature.gro..
<input checked="" type="checkbox"/> WindowBuilder Core UI	1.5.1.r42x20120916...	org.eclipse.wb.core.ui.feature.feature.gro..
<input checked="" type="checkbox"/> WindowBuilder GroupLayout Support	1.5.1.r42x20120916...	org.eclipse.wb.layout.group.feature.featu..
<input checked="" type="checkbox"/> WindowBuilder XML Core (requires Eclipse WTP/WST)	1.5.1.r42x20120916...	org.eclipse.wb.core.xml.feature.feature.gro..

Select All Deselect All

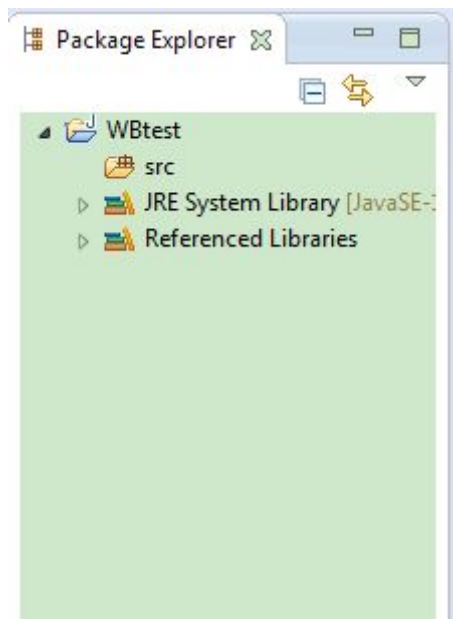
之后一路点下去就好了，会有一个安装过程，安装完后弹出一个对话框提示重新启动 eclipse。

重新启动 eclipse 后，在 Window -- Preferences 看到下面情况，说明安装成功。

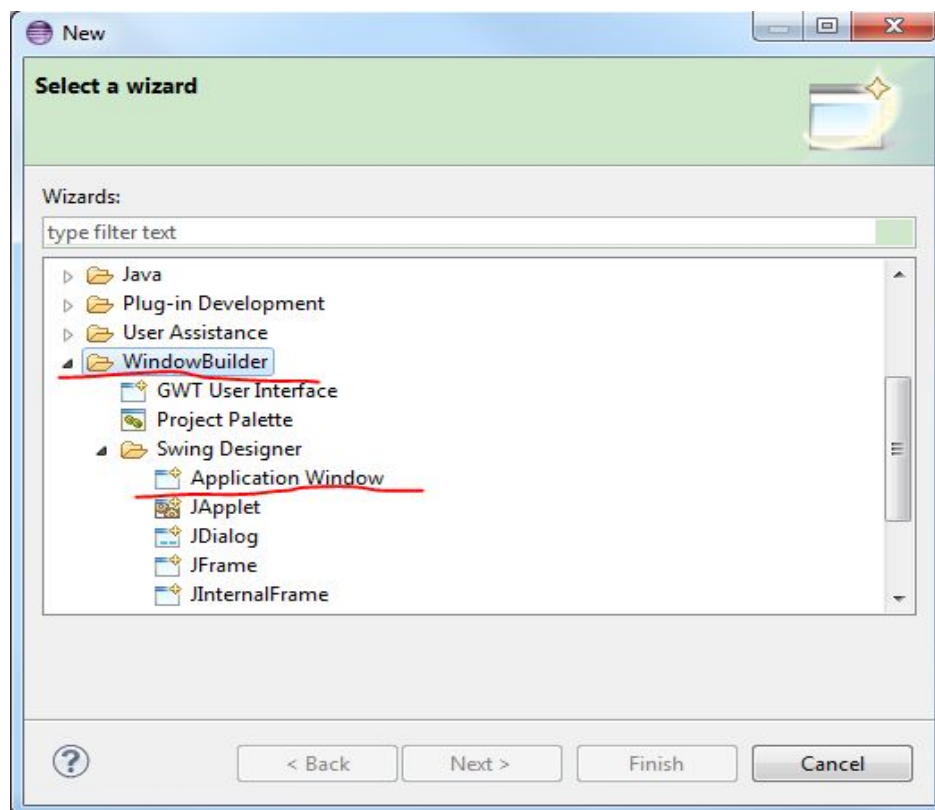


二、WindowBuilder 使用

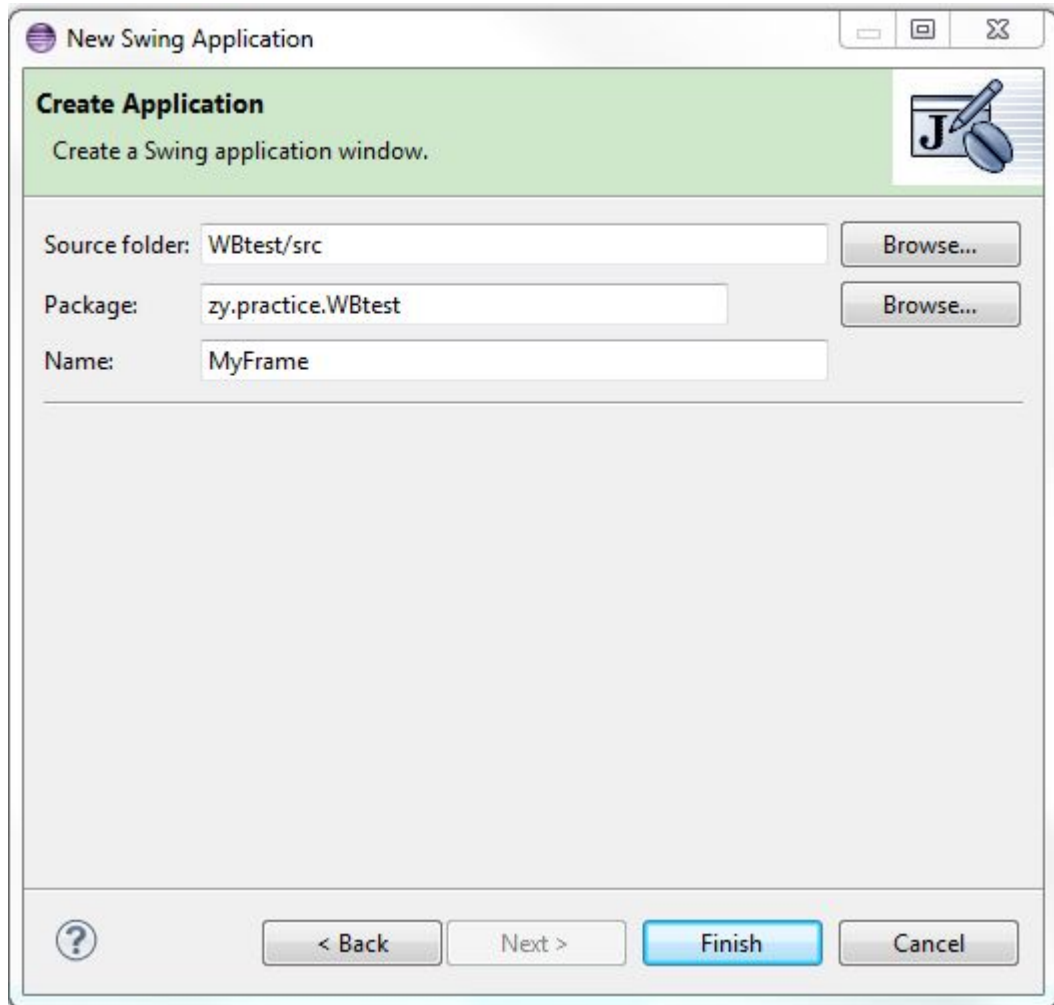
1. 新建一个 Java project. 取名为 WBtest, 如图所示:



2. 在 src 上右键 -- New -- Other... -- WindowBuilder -- Swing Designer -- Application Window



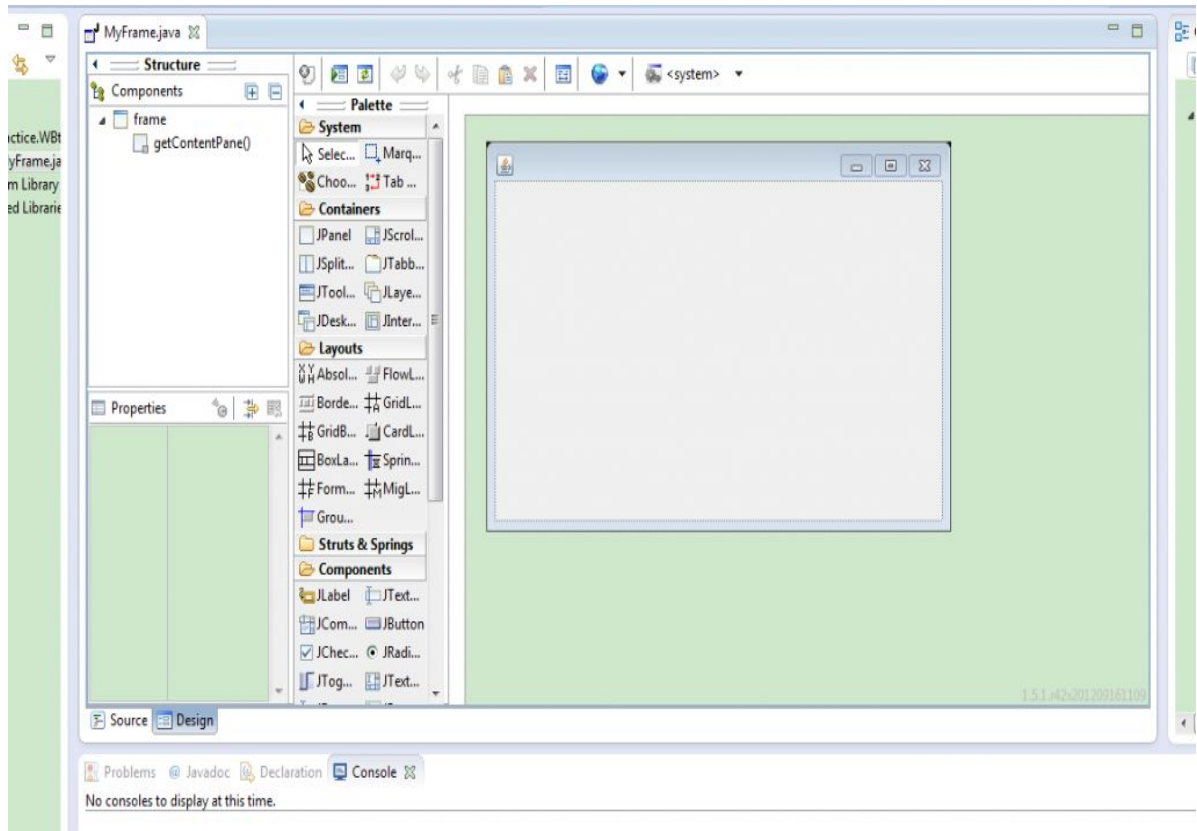
Next -- 输入 Package 和 Name -- Finish



会看到下面画面：



注意红线圈出来的位置，点击 Design



可视化编程界面就出现了，这以后就可以根据自己的需要选择设计界面了。

三、DAO 设计模式介绍

1、DAO 简介：

DAO 设计模式是属于 J2EE 数据层的操作，使用 DAO 设计模式可以简化大量代码，增强程序的可移植性。

2、DAO 各部分详解：

DAO 设计模式包括以下 5 个主要部分：

（1）、数据库连接类；

数据库连接类的主要功能是连接数据库并获得连接对象，以及关闭数据库。通过数据库连接类可以大大的简化开发，在需要进行数据库连接时，只需常见该类的实例，并调用其中的方法就可以获得数据库连接对象和关闭数据库，不必再进行重复操作。例：

```
Package com.javaweb.dao1

Import java.sql.*;

Public class DataBaseConnection{

    Private final String DBDRIVER = “com.mysql.jdbc.Driver” ;

    Private final String DBURL = “jdbc:mysql://localhost:3306/javaweb” ;

    Private final String DBUSER = “root” ” ;

    Private final String DBPASSWORD = “585762” ;

    Private Connection conn = null;

    Public DataBaseConnection(){

        Try{

            Class.forName(DBDRIVER);

            This.conn =

DriverManager.getConnection(DBURL,DBUSER,DBPASSWORD);

        }catch(Exception e){
```

```

        System.out.println( “加载数据库驱动失败！ ” );

    }

}

Public Connection getConnection(){

    Return conn;

}

Public void close(){

    Try{

        Conn.close();

    }catch(){

        System.out.println( “数据库连接关闭失败！ ” );

    }

}

}

```

（2）、VO 类：

VO 类是一个包含属性和表中字段完全对应的类，并在该类中提供 **setter** 和 **getter** 方法来设置并获取该类中的属性。例：

```

Package com.javaweb.dao1

Public classs User{

    Private int userid;

    Private String username;

    Private String password;

    Public int getUserid(){

        Return userid;

    }

    Public void setUserid(int userid){

        Return userid;

    }

}

```

```

    }

    Public String getUsername(){

        Return username;

    }

    Public void setUsername(String username){

        Return username;

    }

    Public String getUserpassword(){

        Return userpassword;

    }

    Public void setUserid(String userpassword){

        Return userpassword;

    }

}

```

(3)、DAO 接口：

DAO 接口中定义了所有的用户操作，如添加记录、删除记录及查询记录等。不过因为是借口，所以仅仅是定义，需要子类实现。例：

```

Package com.javaweb.dao1;

Import java.util.*;

Public class UserDao{

    public void insert(User user)throw Exception;

    Public void update(User user)throw Exception;

    Public void delete(int userid)throw Exception;

    Public User queryById(int userid)throw Exception;

    Public List queryAll()throw Exception;

}

```

(4)、DAO 实现类：

DAO 实现类实现了 DAO 接口，并实现了接口中定义的所有方法。

(5)、DAO 工厂类：

在没有 DAO 工厂类的情况下, 必须通过创建 DAO 实现类的实例才能完成数据库操作。这时就必须知道具体的子类, 对于后期的修改非常不方便。

使用 DAO 工厂类, 可以比较方便地对代码进行管理, 而且可以很好地解决后期修改的问题, 通过该 DAO 工厂类的一个静态方法来获取 DAO 实现类实例。这时如果要替换 DAO 实现类, 只需要修改该 Dao 工厂类中的方法代码, 而不必邀修改所有的操作数据库代码。例:

```
Package com.javaweb.daol;  
  
Public class DAOFactory{  
  
    Public static UserDao getUserDAOInstance() {  
  
        Return new UserDaoImpl();  
  
    }  
  
}
```

3、使用 DAO 完成数据库操作:

- (1)、添加记录:
- (2)、更新记录:
- (3)、删除记录:
- (4)、查询记录: