**认识J2SE**

**一、实验简介**

在本章学习开始前，你应该具备一些[Java的基础知识](https://www.shiyanlou.com/courses/18)。我们将在本章来认识J2SE，并复习一下前面学过的面向对象的相关知识。

**注：所有的蓝色文字都是带超链接的，这些链接是本课程为你提供的扩展知识点，一定不要错过。**

本章知识点

* J2SE的定义和架构
* 面向对象

**二、什么是J2SE**

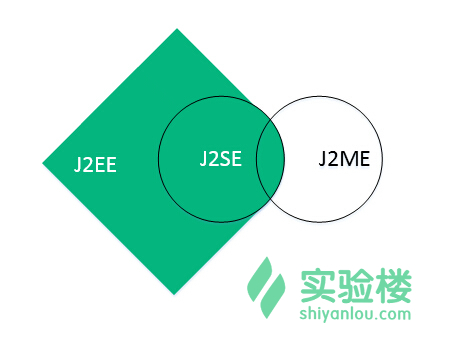
**1. J2SE的定义**

J2SE，全称为Java 2 Standard Edition。Java 2平台包括：标准版（J2SE）、企业版（J2EE）和微缩版（J2ME）三个版本。J2SE即Java 2的标准版，主要用于桌面应用软件的开发。

下面这段话是ORACLE对于Java SE的官方描述：

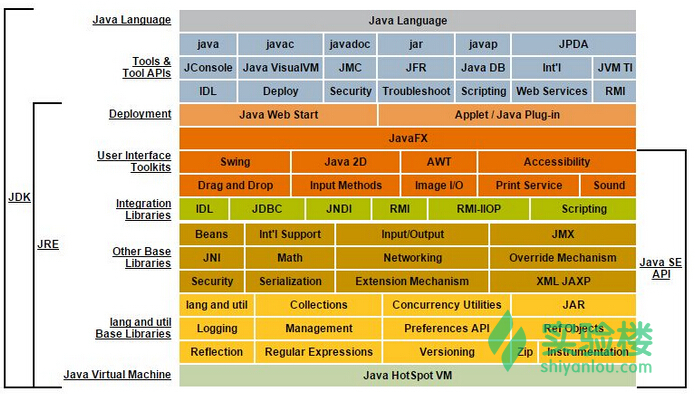
Java Platform, Standard Edition (Java SE) 可以让您在桌面和服务器以及目前要求较高的嵌入式环境中开发和部署 Java 应用程序。 Java 提供了当今应用程序所需要的丰富的用户界面、良好的性能、多功能性、可移植性和安全性。

J2SE与J2EE、J2ME之间的关系可以通过下图来表示：



**2. J2SE的架构**

J2SE的架构如下图所示，它主要包含了UI、集成库、语言和工具基础库、其他基础库、Java虚拟机等组件。



综上所述，将J2SE压缩一点再加上一些[CLDC](http://baike.baidu.com/link?url=YyhiiGfjgAR-CzCLq5nHUKwDvAKRCTC-JDmf_uAcYyqJerKQaB36sLlFNzoO9JtrutrOH1UnbAZjs7lrcmJKRK" \t "_blank)等方面的特性就是J2ME；将其扩充一点再增加一些EJB等企业应用方面的特性就是J2EE。因此J2SE是J2EE的基础，建议从事Java的开发人员从J2SE开始学习。

**三、面向对象——巩固与提高**

在前面Java基础语法的学习中，你应该接触了一些面向对象的基础知识。面向对象的概念在Java的开发体系中无处不在，在今后的开发过程中，你也应该以面向对象的思想来看待和解决问题。

既然是面向对象，就会始终谈论到类和对象的概念，以及它们之间的关系：

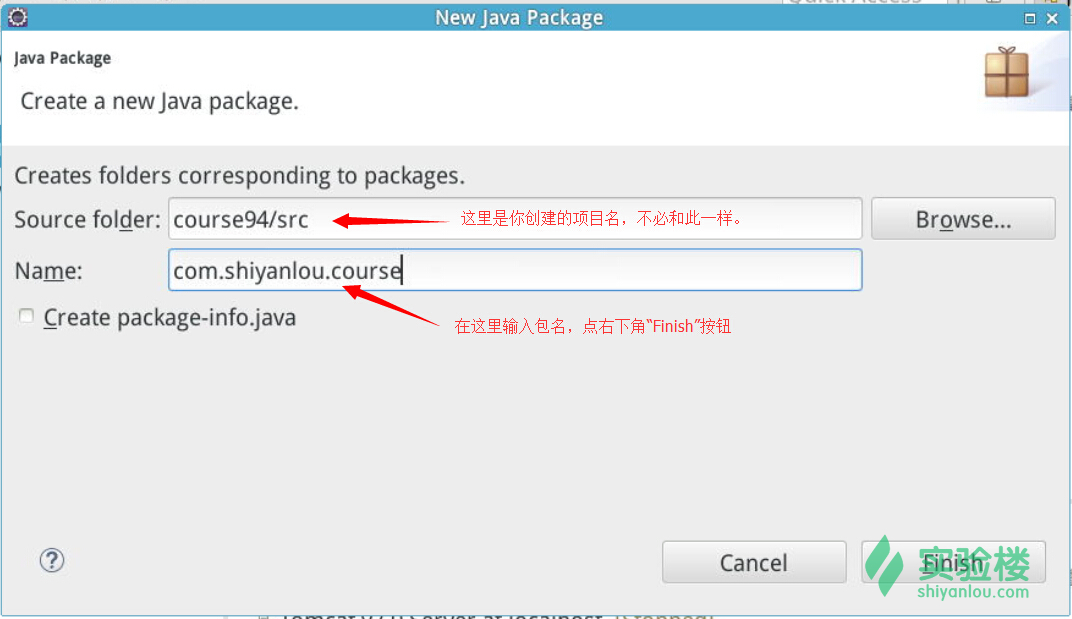
类是现实世界或思维世界中的实体在计算机中的反映，它将数据以及这些数据上的操作封装在一起。对象是具有类类型的变量。

类是对象的抽象，而对象是类的具体实例。类是抽象的，不占用内存，而对象是具体的，占用存储空间。类是用于创建对象的蓝图，它是一个定义包括在特定类型的对象中的方法和变量的软件模板。

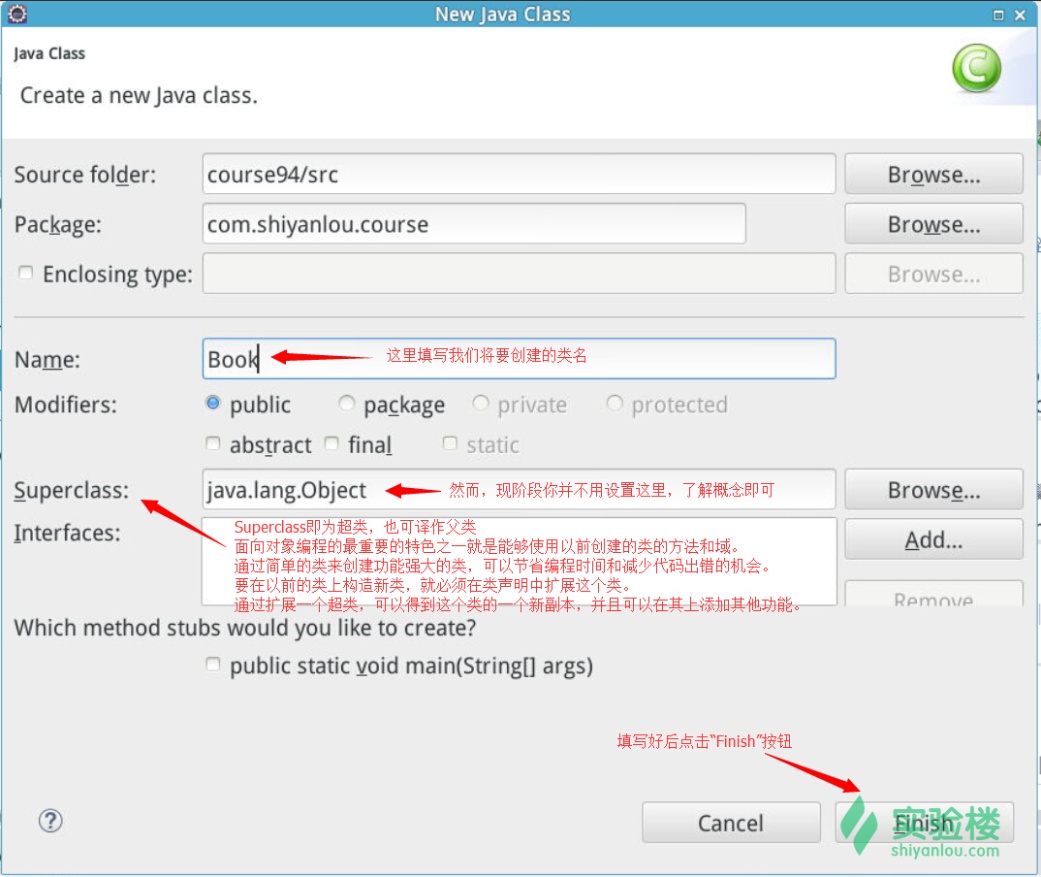
下面，我们通过一些例子来巩固面向对象的相关知识。

**1. 自定义图书类**

首先在Eclipse中创建一个项目HelloJ2SE，新建一个名为com.shiyanlou.course的包。



然后在这个包里面添加一个Book类。



对于一个图书对象，一般有书名、作者和编码等属性。我们利用这些常见属性来自定义这个图书类，创建的Book.java文件中主要代码如下：

**特别说明：在实验楼的实验环境中，暂不支持输入中文，因此你不必将代码片段中的注释也输入开发环境。该说明同样适用于本课程后续的章节。**

package com.shiyanlou.course;

//此为包名，如果你在创建包时已自动生成了该行，请忽略

public class Book {

private String name; // 定义书名

private String author; // 定义作者

private String ISBN; // 定义编码

//Tips: ISBN是国际标准书号，每本书背面的条形码即为此物

public Book(String name, String author, String ISBN) {

// 利用构造方法初始化域

this.name = name;

this.author = author;

this.ISBN = ISBN;

//Q:你清楚this的用法吗？

}

public String getName() { // 获得书名

return name;

}

public String getAuthor() { // 获得作者

return author;

}

public String getISBN() { // 获得编码

return ISBN;

}

}

为了测试我们刚刚自定义的图书类，我们在com.shiyanlou.course这个包中再创建一个名为Test的类文件，并在main()方法中创建一个Book对象。最后，我们在控制台输出这个Book对象的属性。

创建的Test.java文件中，主要的代码如下：

package com.shiyanlou.course;

public class Test {

public static void main(String[] args) {

Book book = new Book("This is my 1st book", "Peter", "1234567890");

// 创建Book对象并设定其各个属性

System.out.println("Book Name: " + book.getName());

// 输出这本书的名字

System.out.println("Book Author: " + book.getAuthor());

// 输出这本书的作者

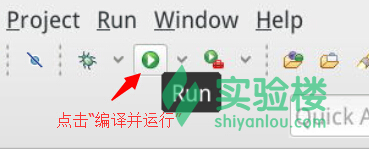
System.out.println("ISBN: " + book.getISBN());

// 输出这本书的编码

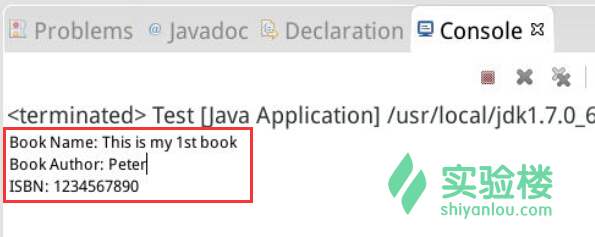
}

}

点击编译并运行。



如果在控制台中可以看到下图这样的信息，那么祝贺你，一个自定义的图书类就成功完成了。



当然，你也可以为Book对象添加更多的属性，并为每个属性编写get方法。

创建工程🡪创建包名🡪创建类:私有属性（成员变量）添加getter和setter

**2. 动态化实例类**

在面向对象的编程中，把用类创建对象的过程称为实例化。通常是使用有参数或无参数的构造方法来创建对象。其格式如下：

//有参数的情形

类名 对象名 = new 类名(参数1，参数2...参数n);

//例如下面这个例子

Person peter = new Person("Peter","boy");

//无参数的情形

类名 对象名 = new 类名();

//例如下面这个例子

Dog dog\_1 = new Dog();

你可以仿照自定义图书类的例子，通过自己创建相关的Person类和Dog类以及测试的方法来验证上述过程。

但是在Java中，类的实例化方法一共有四种途径：

1. 使用new操作符
2. 调用Class对象的newInstance()方法
3. 调用clone()方法，对现有实例的拷贝
4. 通过ObjectInputStream的readObject()方法反序列化类

我们在自定义图书类这个例子中，编写了有参数的构造方法public Book(String name, String author, String ISBN)。但如果类中没有定义构造方法，编译器便会自动添加一个无参数的构造方法。使用构造方法创建对象虽然常用，但并不灵活。因此，我们再来了解一下动态化实例类，也就是调用Class对象的newInstance()方法，通过反射创建对象。

了解一下反射的概念：

JAVA反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

那么我们就开始来学习如何动态化实例类：

请在在Eclipse中创建一个项目DynamicClass，并在该项目中创建com.shiyanlou.course包。在该包中创建Test类，并编写main()方法。

在main()方法中创建一个File对象（你只需要知道File对象也是一个对象即可，[进一步了解File对象](http://docs.oracle.com/javase/7/docs/api/java/io/File.html" \t "_blank)）。

最后，使用该对象在桌面创建一个文本文件。

主要的代码如下：

package com.shiyanlou.course;

import java.io.File;

import java.lang.reflect.Constructor;

//需要引用上述两个包

public class Test {

public static void main(String[] args) {

try {

Constructor<File> constructor = File.class.getDeclaredConstructor(String.class);

//获得File类的Constructor对象

System.out.println("Create File Object with reflection.");

//使用反射创建File对象

File file = constructor.newInstance("/home/shiyanlou/Desktop/MyFile.txt");

System.out.println("Use File Object to create MyFile.txt on desktop.");

//指定了创建的路径为桌面，名称为“MyFile.txt”

file.createNewFile(); //创建新的文件

System.out.println("File is created ?" + file.exists());

//验证文件是否创建成功

} catch (Exception e) {

e.printStackTrace();

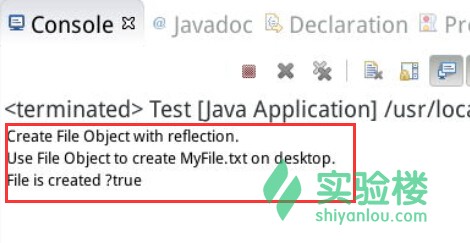
}

}

}

你可能会问为什么会用到try、catch以及Exception之类的奇怪的东西，这是由于上述代码会抛出大量的异常。我们通常不推荐这样写。不用担心，我们将在**错误处理**一章为你详细介绍他们的原理和用法。

点击编译并运行，如果在控制台可以看到下列信息则表示你成功了。



同时，你也应该可以在桌面上发现自己创建的这个MyFile.txt文件。



**四、实验总结**

本章我们学习了J2SE的基本概念和架构，复习了面向对象中类和对象、动态实例化类的相关知识。建议大家再积累一些数据结构相关知识，以便更好地投入到后续的学习中。

J2SE的整个体系还是很庞大的，本课程会力所能及地涵盖其核心的知识点，如果你想更加深入地学习Java知识，我们还推荐你去阅读**《Java 2核心技术》**等书籍。

**五、作业**

好了，我们今天的认识J2SE课程就讲完了，你对Java这个大家庭又了解了多少？

不妨通过解决下面的问题来巩固今日所学吧。

一个学校里面有老师和学生，他们具有一些相同的属性，例如姓名、生日、性别等。但老师还有一个特别的工龄属性，请尝试利用学生类来编写老师类，利用继承技术让老师类使用学生类中定义的属性和方法，并且为老师类创建新的工龄属性。

为什么要这样做？因为[继承技术](http://baike.baidu.com/link?url=nCZxTZfD01QdeodlmhGU74gt7lQlOPuqmEcD-4A4zrT5as7vovuVL69EKnpEF7_wyRrbrDWTgTFp29-WH4_DGK)可以让你少写很多代码啊…