



# 电子商务技术基础

## 第一章 电子商务技术概论

# 第一章 电子商务技术概论

- 第一节 电子商务基本概念
- 第二节 电子商务系统
- 第三节 电子商务系统平台
- 第四节 Java技术基础



# 第一节 电子商务的基本概念

## 1、电子商务的定义

### ■ 对电子商务的不同理解

- 1996.12. 联合国国际贸易法委员会通过的《电子商务示范法》中提到电子商务，但未给出明确定义，只是强调其特殊性——**在交易中使用数据电文作为交易信息载体。**
- 1997.11. 国际商会第一届世界电子商务会议（法国巴黎）——**对整个贸易活动实现电子化**，强调交易各方以电子交易方式而不是通过当面交换或直接面谈的方式进行的任何形式的商业交易。对**电子化技术和业务范围**做了阐明。

# 第一节 电子商务的基本概念

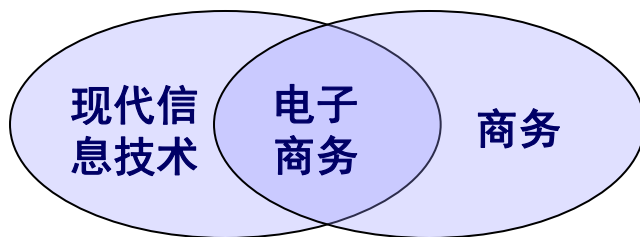
## 1、电子商务的定义

- 电子商务（Electronic Commerce, E-Commerce）  
是指交易当事人或参与人利用**现代信息技术和计算机网络**(主要是因特网)所进行的各类商业活动，包括货物贸易、服务贸易和知识产权贸易。
- 电子商务是以信息网络技术为手段，实现消费者的网上购物、商户之间的网上交易和在线电子支付以及各种商务活动、交易活动、金融活动和相关的综合服务活动的一种新型的商业运营模式。
- 电子商务的出现，给社会、经济、政治等带来的巨大影响。

# 第一节 电子商务的基本概念

## 1、电子商务的定义

- 电子商务是商务与信息技术的结合体。一方面，电子商务使用了网络为基础的现代信息技术；另一方面，一切包括契约型或非契约型、具有商务性质的事务都属于电子商务的商务范畴。
- 电子商务所覆盖范围是这两个子集形成的交集。



**电子商务是“现代信息技术”和“商务”两个子集的交集。**

# 第一节 电子商务的基本概念

## 2、电子商务的特点

- **价值创新。**电子商务把互联网作为商品销售渠道，与线下渠道形成差异化竞争策略。单就购买体验来讲，它与线下渠道互有长短，但是电子商务具有关键的低成本优势，为用户创造了更大价值。
- **覆盖头部市场和长尾市场。**电子商务能够提供标准化的热门产品；而对于个性化的小众需求，由于品类足够丰富、商品供应者众多，这些需求同样可以被满足，而其满足成本与热门商品并无显著差异。

# 第一节 电子商务的基本概念

## 2、电子商务的特点

- **以消费体验带动流程优化。**对于用户体验的重视，是电子商务成功吸引海量用户高频购买的前提。这种体验不仅体现于软件操作层面，而且贯穿于商品查找、支付、交付和售后服务的全过程。例如对于商品交付周期和价格的关注，导致供应链不断优化，直接促进了物流行业的爆发。
- **消费者地位的反转。**电子商务汇集起来的庞大用户数，使得个性化产品也能拥有不小的市场。在商品的设计与生产、自身权益保护方面，个人消费者开始拥有发言权。消费者地位的反转，推动了制造企业流程的再造，强化了渠道的服务职能，使得商业服务更加贴近本质。

# 第一节 电子商务的基本概念

## 3、电子商务与传统商务的区别

- 电子商务主要是依托互联网技术把传统的商品贸易形式迁移到线上，以降低交易成本、吸引价格敏感用户。随着电子商务日益发展，与传统商务的区别日益明显：
  - 通过新型交易技术（线上交易、网络支付等）扩大了销售范围和消费者的选择权，提升消费体验，降低了产品展示与销售的成本；
  - 带动了企业的经营过程（依托信息系统的企业流程再造）、组织机构（依托快速信息交流的组织机构扁平化）和生产工具（柔性生产、制造技术）的优化，降低了生产与运输成本；
  - 为企业搜集、理解相应用户的个性化需求提供了快速、低成本的手段（如在线反馈、客户关系管理系统、基于数据的客户需求分析系统等）。
- 电子商务以交易上网为契机，不但节省了交易成本，提高了商务效率，而且驱动、融合了整个产业链（生产、物流、渠道和支付）的改造。



# 第一节 电子商务的基本概念

## 4、电子商务的主要类型

分类标准	分类
参与主体	B2C、B2B、B2G、C2C、O2O等
网络类型	基于企业内部网的电子商务、基于企业外部网的电子商务、基于互联网的电子商务
交易活动网上完成的程度	完全电子商务、不完全电子商务
交易的地域范围	本地电子商务、区域电子商务、国家电子商务、跨境电子商务

# 第一节 电子商务的基本概念

## 5、电子商务相关技术

- 电子商务是商务与信息技术(计算机技术、网络技术、通信技术)的结合，商务的发展为各种技术的应用与发展提供了广阔的舞台，技术的发展又有力地促进了电子商务的产生和发展。
- 电子商务中应用到的一些技术主要有：网页设计技术、分布处理与中间件技术、应用系统设计与整合技术、网络安全技术、电子支付技术、物流技术、网络营销技术（如信息检索与数据挖掘技术、信息发布技术等）、移动电子商务技术、云网技术(大数据与云计算)、Web2.0应用技术等。

# 第一章 电子商务技术概论

- 第一节 电子商务基本概念
- **第二节 电子商务系统**
- 第三节 电子商务系统平台
- 第四节 Java技术基础



## 第二节 电子商务系统

### 1、电子商务系统的定义

- 从广义上讲，电子商务系统是指支持商务活动的电子技术手段的集合。
- 从狭义上讲，电子商务系统则是指在互连网和其他网络的基础上，以实现企业电子商务活动为目标，满足企业生产、销售及管理的需要，支持企业对外业务协作，为企业提供商业智能的信息系统。
- 注：电子商务系统 ≠ 电子商务网站

## 第二节 电子商务系统

### 2、电子商务与电子商务系统的区别

- 以电子技术为手段的商务活动称为**电子商务**。
- 商务活动所赖以存在的环境称为**电子商务系统**。

↕	电子商务↕	电子商务系统↕
定义↕	以电子技术为手段的商务活动↕	电子商务活动赖以存在的物质基础和活动环境↕
目标↕	完成商务活动↕	提供商务活动所需的信息沟通与交流的软硬件环境及相关的信息流程↕
功能↕	及时、准确地提供商品或商务服务↕	适时、适地提供恰当的信息（支持电子商务的运行）↕
内容↕	商品的生产、配销、运输活动；信息收集、处理、控制和传递活动↕	信息搜集、处理、控制和传递活动↕

## 第二节 电子商务系统

### 3、电子商务系统的三个特性

- 电子商务系统是一种特殊的管理信息系统。
- 电子商务系统与企业内部信息系统形成了一个整体。
- 电子商务系统的逻辑结构呈现出清晰的层次。

## 第二节 电子商务系统

### 3、电子商务系统的三个特性

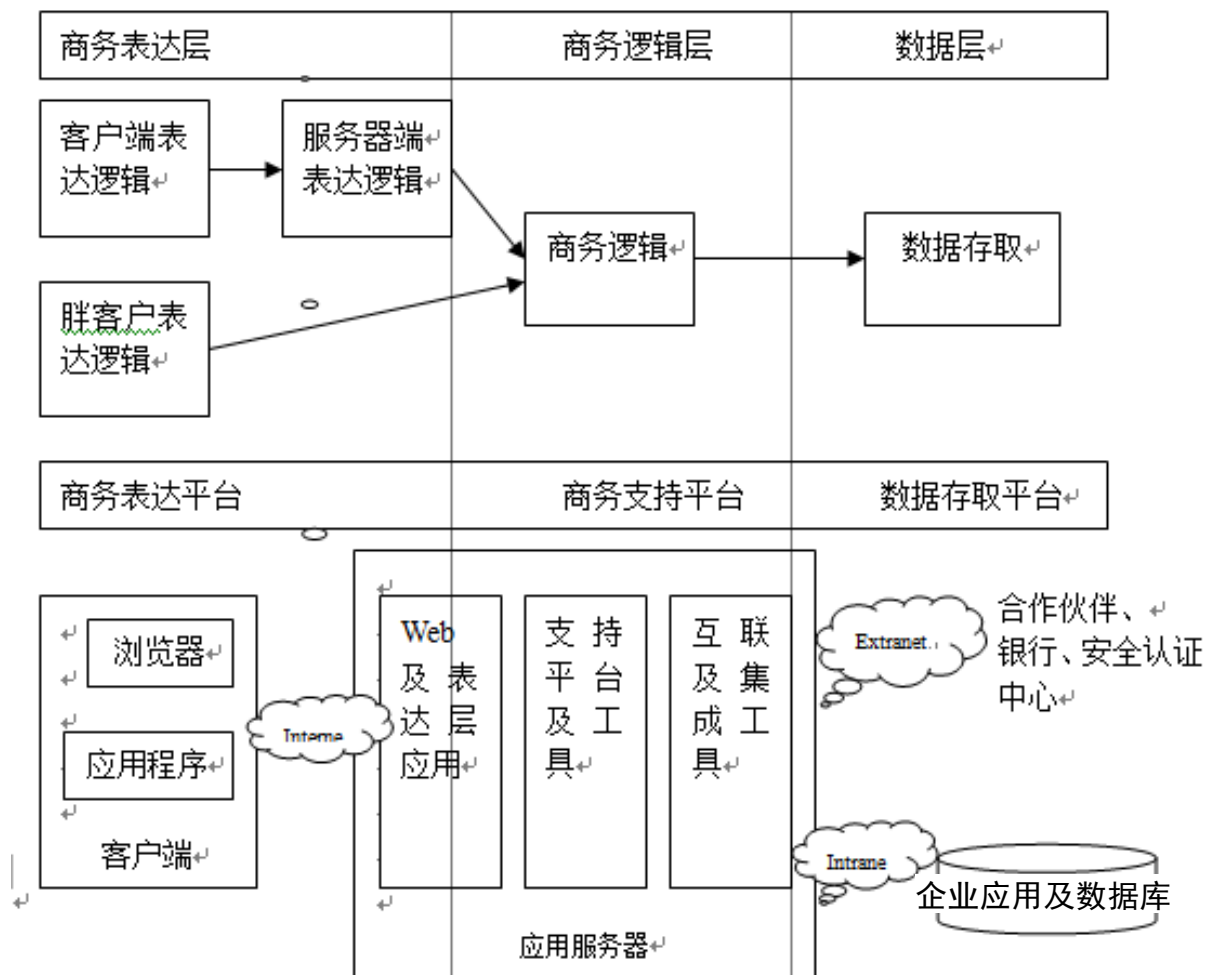
#### 电子商务与管理信息系统的区别

	系统需求		信息特点		技术特点	系统特点
MIS	管理者的 信息需求 分析	组织内部 个体功能	信息共享 纵向加工	量大， 传递少	数据库、模型等 通过局域网 进行数据共享	系统是专用、封 闭的，对外不工 开
EC	业务流程 分析	组织内外 多方交互	信息交易 横向互动	量小， 传递多	Web技术等 Internet、广域网 将不同组织的系统 连在一起	开放大系统 动态的

## 第二节 电子商务系统

### 3、电子商务系统的三个特性

#### 电子商务系统的逻辑结构



电子商务系统的逻辑结构层次清晰



## 第二节 电子商务系统

### 4、电子商务系统的体系结构

#### ■ 商务表达层

- 作用：为最终用户提供界面，帮助系统接受用户的请求。
- 特点：支持多种标准数据格式；支持多种主流数据终端；支持用户的个性化要求。

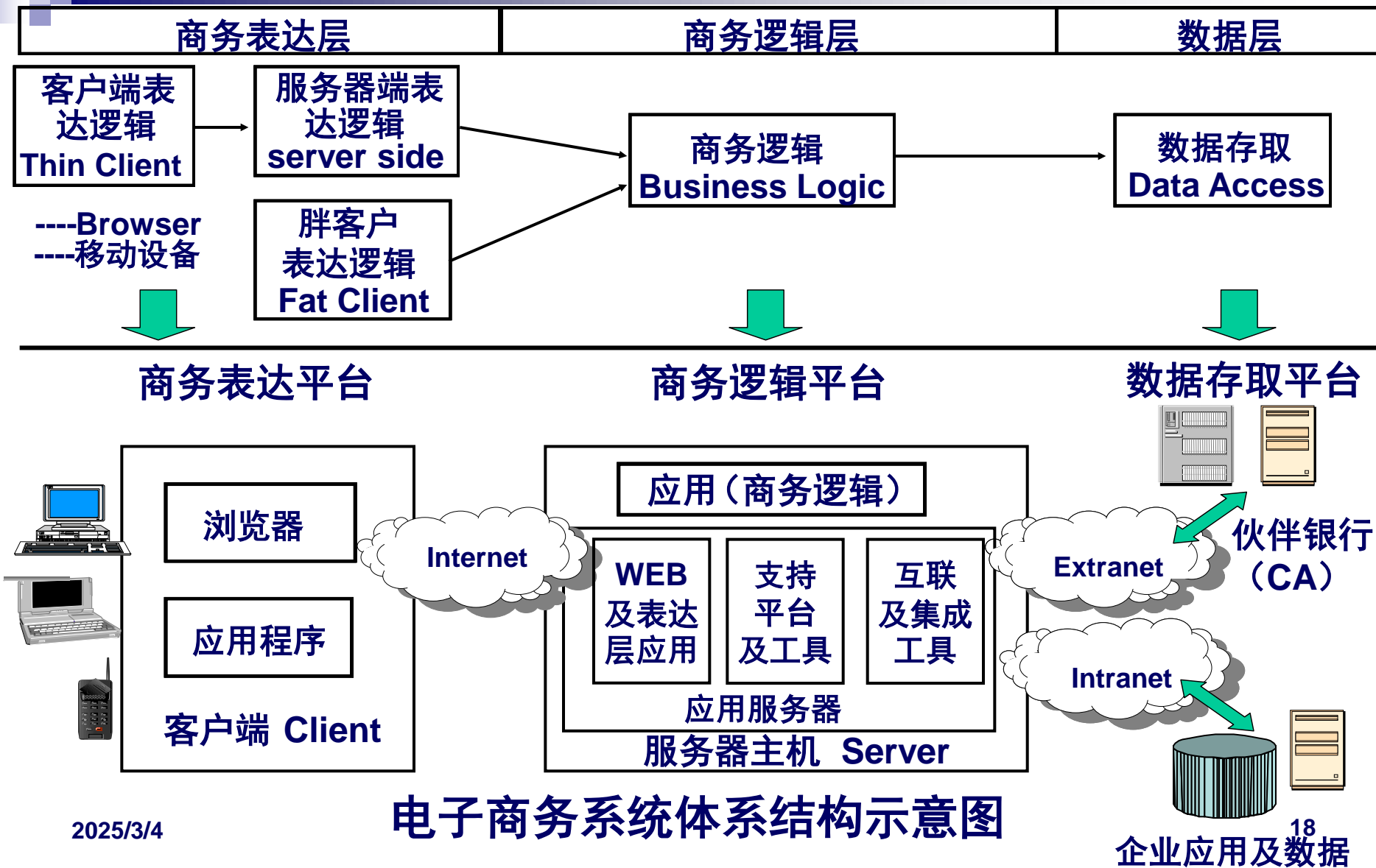
#### ■ 商务逻辑层

- 作用：描述商务处理过程和商务规则。
- 组成部分：实现企业核心商务逻辑的应用软件 and 为商务逻辑提供支持的通用软件（商务和系统平台）。

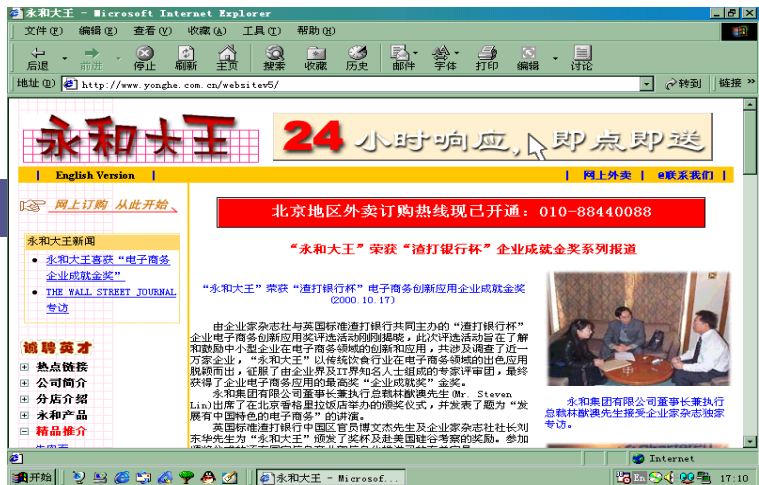
#### ■ 数据层

- 作用：为商务逻辑层的应用系统提供后端数据支持（各种来源、格式的数据）。

## 第二节 电子商务系统

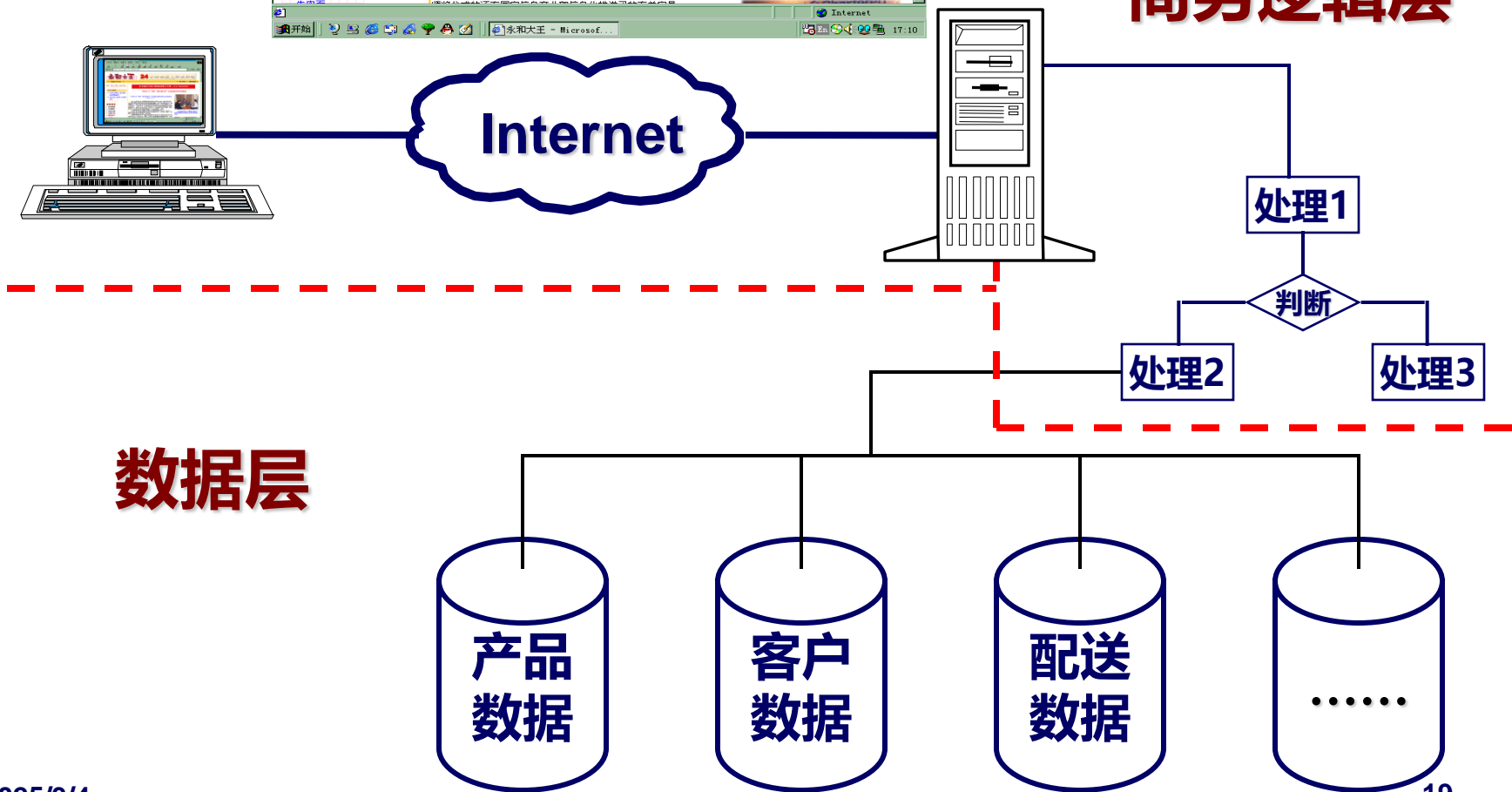


# 商务表达层



构成企业核心商务应用的核  
心商务逻辑，与具体的企业应用密切相关

# 商务逻辑层



# 数据层

## 第二节 电子商务系统

### 5、电子商务系统的功能

#### ■ 电子商务系统的整体功能

- 电子商务系统作为一个完整的大系统，不但具有相应的商务应用功能，而且还具有安全、支付及目录服务等功能。
- 考虑到与其它系统的互联，特别是与异种平台系统（如不同的网络、操作系统、数据库、应用系统等）的互联，以及为了降低系统的复杂性，电子商务系统采用了一些标准的协议和中间件等技术对功能进行了分层，因此**电子商务系统是一个具有层次功能的大系统。**

## 第二节 电子商务系统

### 5、电子商务系统的功能

#### ■ 电子商务系统的层次功能

电子商务应用 (企业宣传、网络银行、网络购物、拍卖等)					电子商务服务
电子商务服务平台					商务服务基础环境
安全 Firewall	支付 SET/SSL	认证 CA	目录服务 (LDAP等)	具体服务 CRM、SCM	
电子商务应用开发支持平台					软件及开发环境
操作系统 Windows、UNIX等	网络通信协议 TCP/IP、HTTP、WAP等		开发语言 Java、ASP等	对象组件 JavaBeans、EJB等	
计算机硬件 (主机、外围设备等)					硬件环境
网络基础设施 (电信网、有线网、无线网等)					网络环境
社会环境 (法律法规、公共政策、税收、隐私、安全标准、技术标准、人才等)					社会环境

2025/3/4

21

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

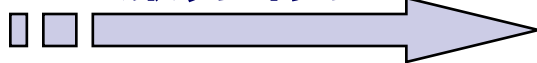
#### (1) 传统的两层客户/服务器结构（C/S结构）

用户界面  
商业和应用逻辑



客户机

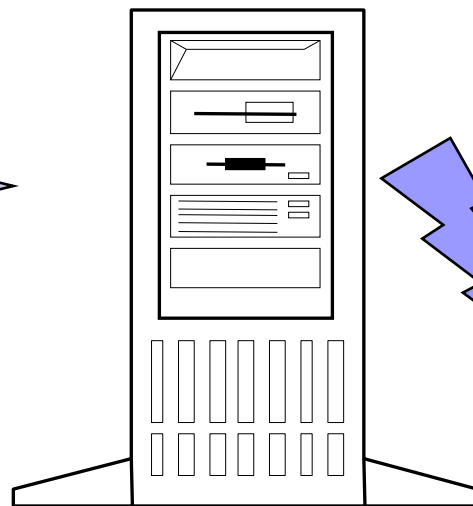
服务请求



响应



数据库



服务器

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (1) 传统的两层客户/服务器结构（C/S结构）

- 由客户机完成商务数据的表示和商务逻辑的运算功能，服务器作为存储数据的部件，充当文件服务器的作用。
- 完成商务数据表示和商务逻辑运算功能的程序，其用户界面与商务逻辑程序代码是紧密地耦合在一起的，并作为一个整体安装在客户机上。



传统的两层客户/服务器结构（C/S结构）

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (1) 传统的两层客户/服务器结构（C/S结构）

优点：

- 减轻主机系统和网络的负担
- 充分利用客户端的处理能力
- 突出了应用程序的功能和能力



## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (1) 传统的两层客户/服务器结构（C/S结构）

缺点：

- 用户界面和数据被分割
- “肥胖”的客户端要管理复杂的软件
- 无法进行有效地扩展，不能支持大量用户访问和高容量事务处理的应用
- 维护困难
- 费用增加
- 培训困难
- 网上的数据流量大，网络负载较重。

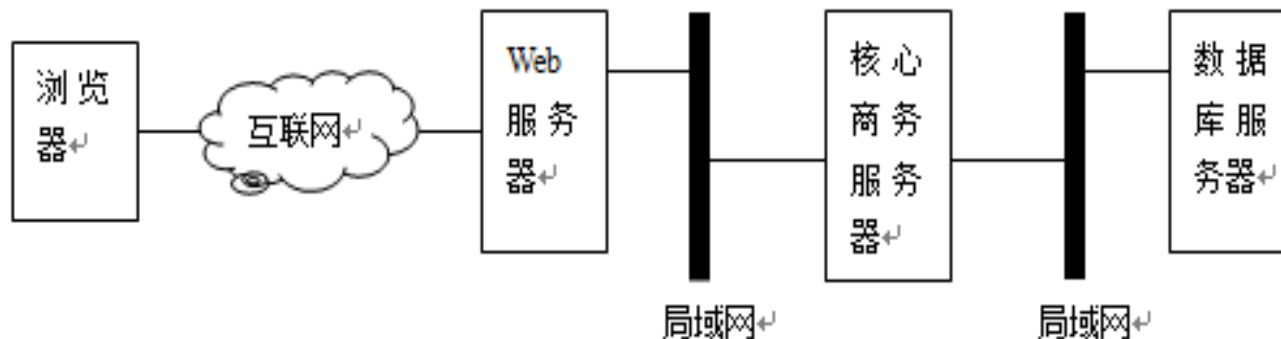
## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (2) 三层客户/服务器结构（B/S结构）

1998年，Sun公司首先提出“三层结构”：

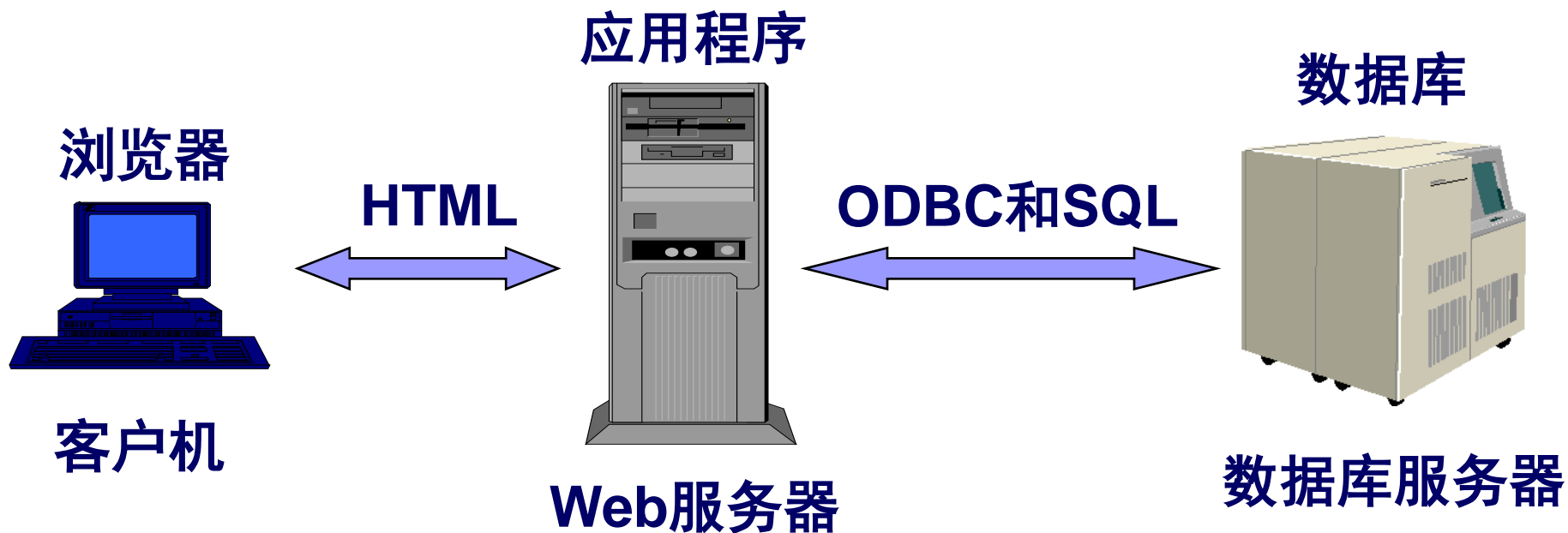
- 表达层以Web服务器为基础，负责信息的发布；
- 应用(逻辑)层负责处理核心业务逻辑；
- 数据层的基础是数据库管理系统DBMS，负责数据的组织，并向应用层提供接口。



## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (2) 三层客户/服务器结构（B/S结构）



B/W/S结构工作原理示意图

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (2) 三层客户/服务器结构（B/S结构）

##### B/W/S或B/S结构

- 客户端（第一层）：只需安装浏览器和相关支持库，负责将标准语言转化为用户界面。
- Web服务器（第二层）：接受客户端的请求，将其转化SQL语句，通过ODBC或其他手段传送至数据库服务器，将数据库服务器的处理结果用HTML格式返回客户端。
- 数据库服务器（第三层）：根据Web服务器发送的请求进行数据库操作，将结果返回Web服务器。

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (2) 三层客户/服务器结构（B/S结构）

优点：

- 在网络的流量和系统的反应速度方面优于两层结构；
- 可伸缩性（根据客户量动态调整连接）；
- 可管理性（主要管理工作集中在应用服务器）；
- 安全性（客户机无法直接访问数据库）；
- 克服了C/S模式客户端多种程序的不一致性；
- 服务器端开放和连接方案的标准化加强了企业和外部的联系；
- 动态交互式信息发布在改进服务质量的同时增加了商业机会；
- 采购数据库服务使用户只要集中于应用程序的开发而无须考虑服务器和数据库的管理。

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (2) 三层客户/服务器结构（B/S结构）

##### C/S架构和B/W/S结构的比较

	C/S模式	B/S模式
结构	分散、多层次结构	分布、网状结构
用户访问	客户端采用事件驱动方式一对多地访问服务器上资源。	客户端采用网络用户界面(Network User Interface, NUI)多对多地访问服务器上的资源,是动态交互、合作式的。
主流语言	第四代语言, 专用工具	Java、HTML类
成熟期	20世纪90年代中期	20世纪90年代末期

## 第二节 电子商务系统

	C/S模式	B/S模式
优点	客户端使用图形用户接口 (Graphic User Interface, GUI), 易于开发复杂程序。	<p>① <b>分散应用与集中管理</b>: 任何经授权且具有标准浏览器的客户均可访问网上资源, 获得网上服务;</p> <p>② <b>跨平台兼容性</b>: 网络软件、语言和应用开发接口均与硬件和操作系统无关;</p> <p>③ <b>系统易维护</b>: “瘦”客户端维护工作大大降低, 灵活性提高; 系统软件版本的升级再配置工作量大幅度下降;</p> <p>④ <b>同一客户机可连接任意服务器</b>。</p>
问题	<p>① 客户端必须安装相应软件才可获得服务;</p> <p>② 与应用平台相关, 跨平台性差;</p> <p>③ 客户端负担较重, 服务器应用需客户端程序;</p> <p>④ 只能与指定服务器相连。</p>	Web服务器应用环境弱、不能构造复杂应用程序。

## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (3) 多层客户/服务器结构（N层结构）

由1999年提出，是对B/S结构的扩展，包括：

- **Web服务器**：处理Internet客户提出的HTTP请求，调用后台网页生成服务，并将生成的页面经HTTP协议返回给客户。
- **逻辑表达及动态网页服务器**：接受Web Service提交的请求，访问后台提供的业务逻辑服务，提取业务数据，生成HTML页面，返回给Web服务器。
- **应用服务器**：执行业务逻辑并访问数据库，更新并提取业务数据，并将结果返回给表达逻辑层。



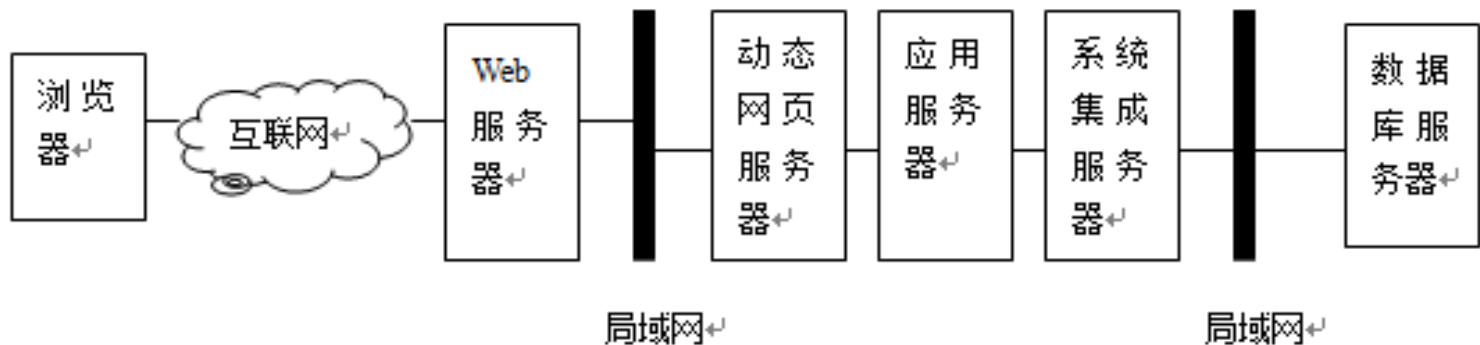
## 第二节 电子商务系统

### 6、电子商务系统的技术架构

#### (3) 多层客户/服务器结构（N层结构）

由1999年提出，是对B/S结构的扩展，包括：

- **系统集成服务器**：接收来自表达逻辑层或业务逻辑层的请求，访问同构或异构资源，并将业务数据返回服务调用者。
- **数据库服务器**



多层客户/服务器结构（N层结构）

# 第一章 电子商务技术概论

- 第一节 电子商务基本概念
- 第二节 电子商务系统
- **第三节 电子商务系统平台**
- 第四节 Java技术基础



# 第三节 电子商务系统平台

- 两大电子商务系统平台
  - Microsoft .NET
  - Sun J2EE (Java EE)

# 第三节 电子商务系统平台

## 1、Microsoft .NET框架

■ .NET = 新平台 + 标准协议 + 统一开发工具

■ 目标

- 一致的面向对象编程环境；
- 软件版本控制冲突最小化的执行环境；
- 保证代码安全执行的环境；
- 可消除脚本环境或解释环境的性能问题的代码执行环境；
- 使开发人员的经验在面对大小不同的应用程序时保持一致；
- 按工业标准生成所有通信，以确保基于.NET框架的代码可与任何其他代码集成。

# 第三节 电子商务系统平台

## 1、Microsoft .NET框架

### .NET框架两大构成部分

- .NET 框架类库（FCL, Framework Class Library）
- .NET 公共语言运行库（CLR, Common Language Runtime）- 支持多种语言，如C#, VB.NET, ASP.NET等

# 第三节 电子商务系统平台

## 2、J2EE平台简介

### 产生背景

- 企业应用系统的开发一直面临着重大挑战。
  - 一方面，企业应用系统面对的是一个异构的分布式环境，需要与已有的系统集成并能进行互操作；
  - 另一方面，作为客户、合作伙伴和企业的信息服务平台，企业应用系统还必须具备高可用性、安全性、可靠性和可伸缩性等特点。
  - 加上用户需求的复杂多变和交付时间的不断伸缩，企业应用系统的开发变得越来越困难。
- 开发商和广大的开发人员一直在努力推动和期待出现一个成熟、标准的公共企业应用平台，以简化和规范企业应用系统的开发和部署。
- 在此背景下，1995年和1997年4月Sun公司的Java One大会上Java技术和J2EE平台相继出现。

# 第三节 电子商务系统平台

## 2、J2EE平台简介

- J2EE本身是一个标准，而不是一个产品(虽然现在有很多符合J2EE标准的产品)。它为不同厂商创建产品提供了标准，并能在这些产品之间实现交互。由以下4个部分组成：
  - **J2EE规范**。该规范定义了J2EE平台的体系结构、平台角色及J2EE中每种服务和核心API的实现要求，它是J2EE应用服务器开发商的大纲。
  - **J2EE兼容性测试站点**。Sun公司提供了一个测试J2EE应用服务器是否符合J2EE规范的站点，对通过该站点测试的产品，Sun公司将发放兼容性证书。
  - **J2EE参考实现，即J2EE SDK**。它既是Sun公司自己对J2EE规范的一个非商业性实现，又是为开发基于J2EE企业级应用系统提供了一个免费的底层开发环境。
  - **J2EE实施指南，即BluePrints文档**。该文档通过实例来指导开发人员如何去开发一个基于J2EE的多层企业应用系统。

# 第三节 电子商务系统平台

## 2、J2EE平台简介

- J2EE平台定义了一个基于组件的多层企业应用系统开发框架，主要包括：**容器、组件、应用程序接口(API)、访问、协议**等五个部分，其中**容器是其核心**。
- **组件**：是一个可以部署的程序单元，它以某种方式运行在容器中。
- **容器**：是指为特定组件提供服务的一个标准化的运行环境，封装了J2EE底层的若干API，为组件提供事务处理、数据访问、安全性、持久性等服务。
- 组件和组件之间不直接访问，而是通过容器提供的协议和方法来相互调用。



# 第三节 电子商务系统平台

## 2、J2EE平台简介

- J2EE定义了4种类型的组件：**Application客户组件、Applet组件、Web组件、EJB组件。**
- Applet组件和Application客户组件运行在客户端：
  - J2EE通过Applet容器为Applet组件提供运行环境，如浏览器中的Applet插件是Applet容器。
  - 通过Application容器为Application组件提供运行环境，如客户端上的Java虚拟机就是Application容器。

Java虚拟机（Java Virtual Machine, JVM）是一个典型的容器。它屏蔽了与具体操作系统平台相关的信息，使得Java程序只需生成在Java虚拟机上运行的目标代码(字节码)，就可以在多种平台上不加修改地运行。



# 第三节 电子商务系统平台

## 2、J2EE平台简介

- J2EE定义了4种类型的组件：**Application客户组件、Applet组件、Web组件、EJB组件。**
- **Web组件和EJB组件运行在服务器端：**
  - Web组件分为JSP和Servlet两种，两者都能动态地生成Web页面，JSP是将Java代码嵌在HTML中，Servlet代码是在服务器端动态生成HTML代码。J2EE通过Web容器(主要由Web服务器来实现)为Web组件提供运行环境。
  - EJB(Enterprise JavaBeans)组件通过EJB容器提供运行环境。EJB容器是J2EE平台最重要的内容之一，实现目录服务、事务处理、持久性、安全性等底层的系统逻辑，使开发者只需关心商业逻辑。

# 第三节 电子商务系统平台

## 2、J2EE平台简介

- **J2EE服务器**：即应用服务器，为容器提供了J2EE中定义的各种服务和API。
- J2EE服务器可以支持一种或多种容器；同时，为了适应组件的应用，J2EE服务器还提供了部署、管理和运行组件的手段。

# 第三节 电子商务系统平台

## 2、J2EE平台简介

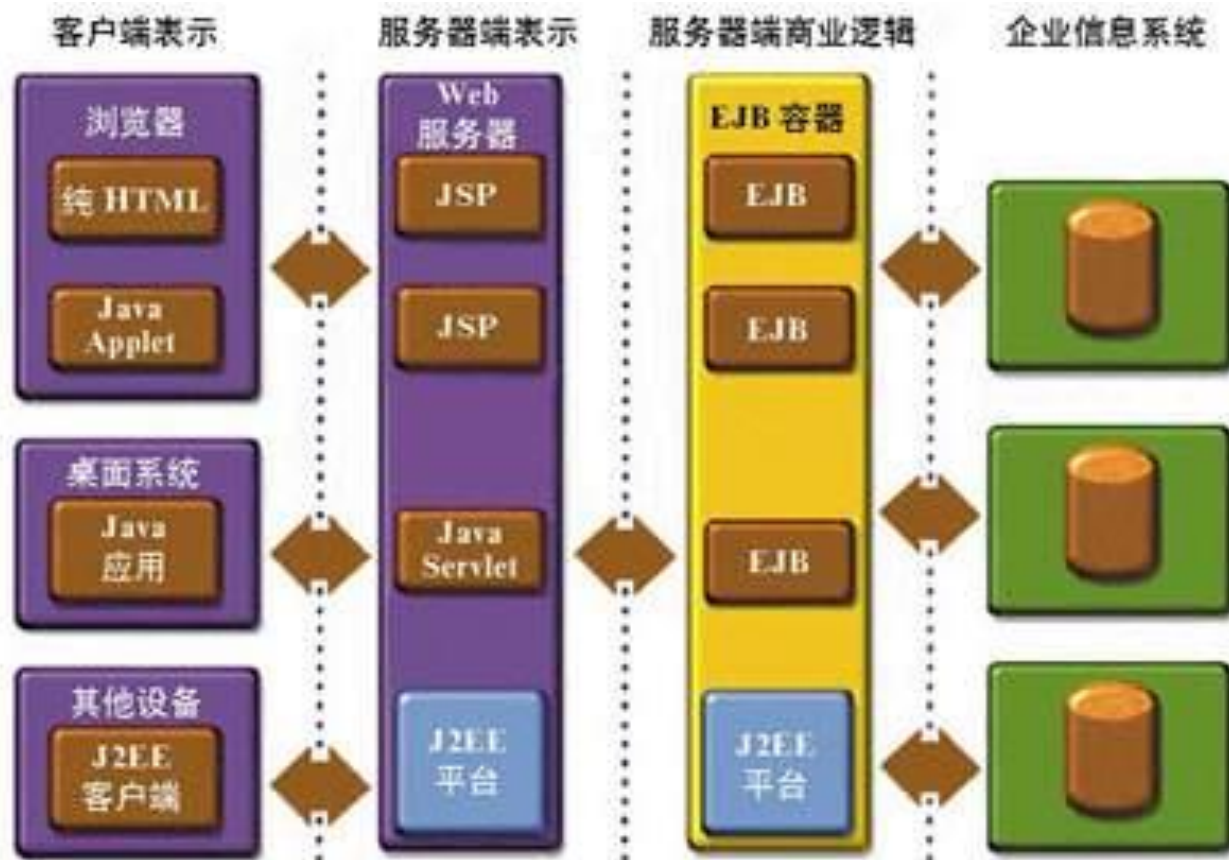
### 典型的J2EE四层结构

- 运行在客户端机器上的客户层组件；
- 运行在J2EE服务器上的Web层组件；
- 运行在J2EE服务器上的业务逻辑层组件；
- 运行在EIS（Enterprise Information System—企业信息系统）服务器上的企业信息系统层软件。

# 第三节 电子商务系统平台

## 2、J2EE平台简介

### 典型的J2EE四层结构



# 第一章 电子商务技术概论

- 第一节 电子商务基本概念
- 第二节 电子商务系统
- 第三节 电子商务系统平台
- 第四节 Java技术基础



## 第四节 Java技术基础

### 1、Java语言简介

- Java是由Sun Microsystems公司于1991开发成功的一种类似C++的高级语言，具有简单、面向对象、安全、分布式、多线程及独立于平台的特点。由于这些特点，使得它不仅是一种面向对象的高级编程语言，而且成为电子商务系统开发与运行的一种标准平台。
- Java既可以开发大型企业的电子商务应用程序，增强Web服务器的功能，也可以编写一些无线或移动设备的通信程序，这是其他高级语言很难做到的。

# 第四节 Java技术基础

## 1、Java语言简介

### ■ Java语言集成开发工具


- Borland公司的Jbuilder
- IBM公司的Eclipse
- Sun公司的J2SDK或NetBeans
- Creator公司的JCreate等。



## 第四节 Java技术基础

### 2、一个简单的Java应用程序

语句行号	程序代码
1	// Hello.java
2	// Text-printing program
3	
4	public class Hello {
5	// main method begins execution of Java application
6	public static void main( String[] args )
7	{
8	System.out.println( "Hello, My Customer!" );
9	} // end method main
10	} // end class Hello



```
C:\> 命令提示符

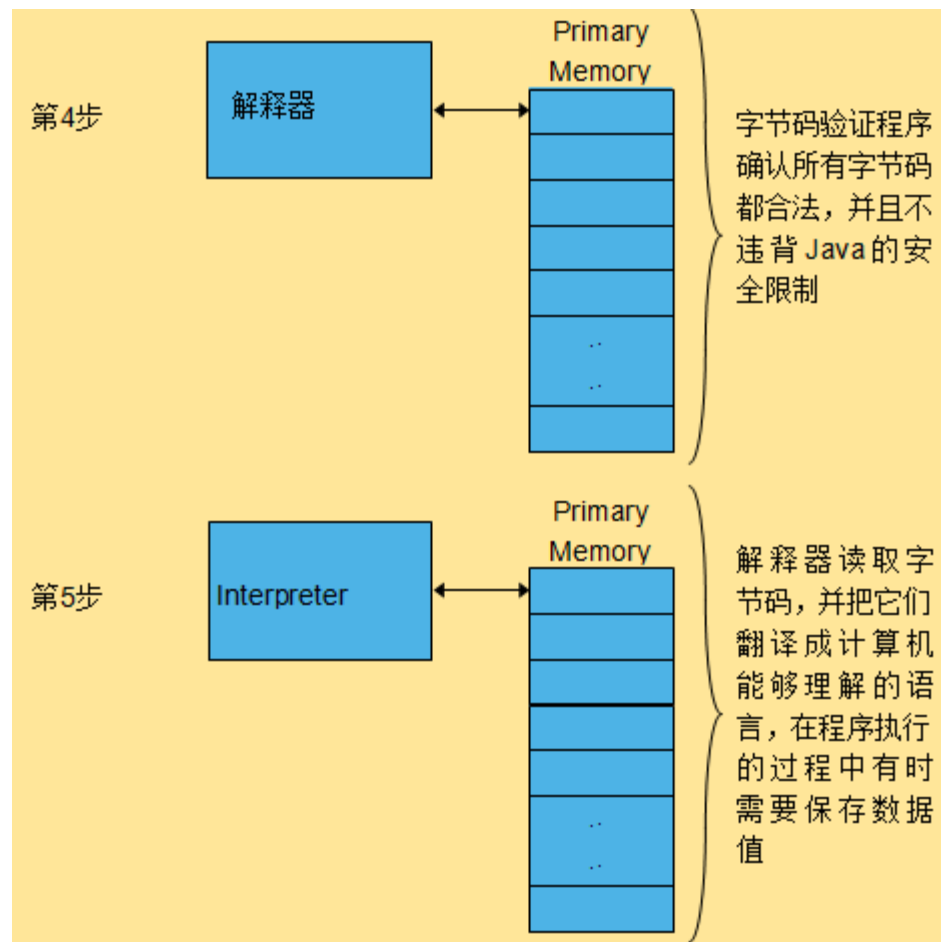
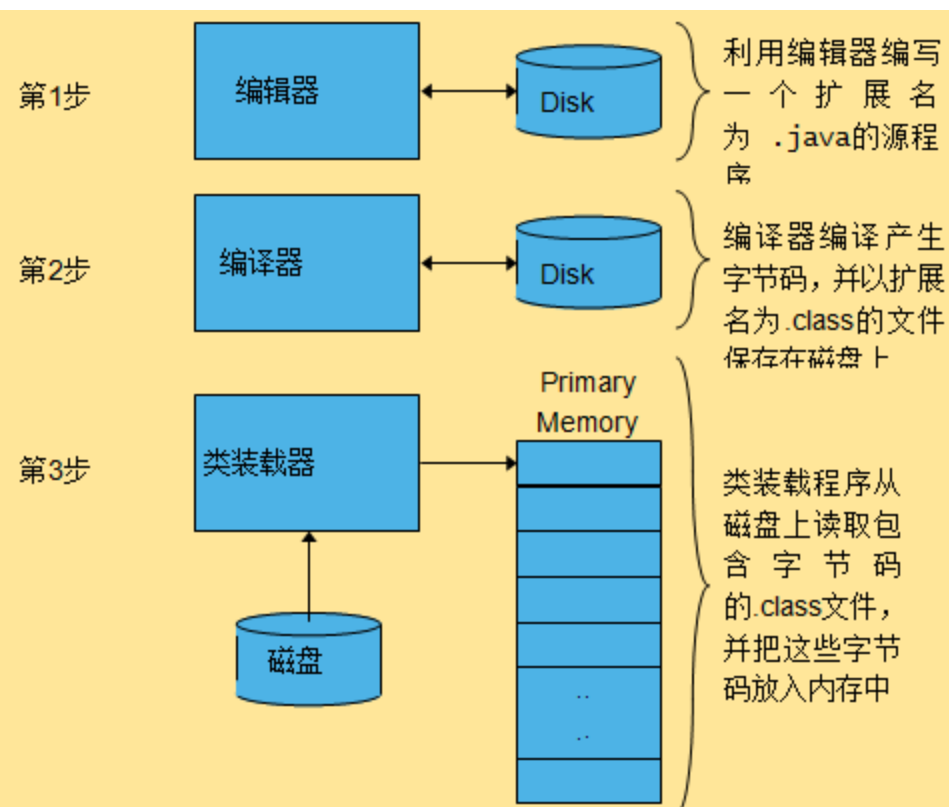
D:\>javac Hello.java

D:\>java Hello
Hello, My Customer!

D:\>
```

# 第四节 Java技术基础

## 3、Java程序的运行（5个步骤）



## 第四节 Java技术基础

### 3、Java程序的运行（5个步骤）

#### ■ 环境变量设置(通用)

□ PATH值为：c:\j2sdk1.4.1\bin\;%path%;

- 作用是指定java命令搜索路径。一般添加jdk安装目录下的bin目录，其中包含常用的可执行文件如javac、java等，设置好PATH变量后，可在任何目录下执行javac/java等工具了。

□ JAVA\_HOME值为：c:\j2sdk1.4.1;

- 一般指java安装目录。

□ CLASSPATH值为：c:\j2sdk1.4.1\lib;

- 作用是指定类搜索路径，要使用已经编写好的类，就是通过CLASSPTH来寻找，一般与import、package等关键字有关。

## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

- Java语言中，存在的数据类型有两大类：**基本的数据类型**和**类（class）**。
- 基本的数据类型有8种：6种数字类型（4个整数型，2个浮点型），1种字符类型，还有1种布尔型。
  - 整数：包括byte, short, int, long
  - 浮点型：float, double
  - 字符：char
  - 布尔：boolean

## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

#### Java中的基本数据类型

基本数据类型	所占字节数 (byte)	取值范围	表示方法示例
boolean	1	true, false	boolean x=true;
byte	1	-128~+127	byte x=13;
char	2	'\u0000'~'\uFFFF'	char x='a'; char y='\u0061'; char z='\\';
short	2	-32768~+32767	short x=230;
int	4	-2147483648~+2147483647	int x=5;
long	8	-2 <sup>63</sup> ~ +2 <sup>63</sup> -1	long x=45L;
float	4		float x=345.00f
double	8		double x=345.00

# 第四节 Java技术基础

## 4、Java程序基本结构、数据类型和运算符

### ■ 类

- 系统类、用户自定义类
- 顶级类、内部类（有名、无名）

- 在一个用户声明的类中，可定义若干变量和函数，分别称为**类的属性**和**方法**。
- 类中还可以再嵌套定义一个类，这样的类称为**内部类**。

## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

表 1-6 一个含有内部类的 Java 程序

<pre>public class MyClass1 {     int attr1;     public void method1(int x) {         attr1=x;         System.out.println("attr1="+attr1);     }     public static void main(String[] args) {         method1(10);     }     private class InnerClass {         int attr2;     } }</pre>	<p>变量attr1是类MyClass1的一个属性</p>
	<p>Method1和main是类MyClass1的两个方法</p>
	<p>InnerClass是类MyClass1的一个内部类</p>

## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

#### ■ Java中的基本运算符

- 算术运算符
- 逻辑运算符

	运算符	部分举例与说明
算术运算符	+	2+3 结果为 5
	-	2-3 结果为-1
	*	2*3 结果为 6
	/	7/5 结果为 1; 7.0/5 结果为 1.4
	%	7%5 结果为 2
逻辑运算符	&&	true && true 结果为 true ; true && false 结果为 false
		false    true 结果为 true ; false    false 结果为 false
	!	!false 结果为 true ; !true 结果为 false



## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

- **补码**：所有的整数类型（除了char类型之外）都是有符号的整数，既能表示正数，又能表示负数。Java使用补码来表示二进制数。
  - 在补码表示中，最高位为符号位，正数的符号位为0，负数为1。补码的规定如下：
    - 对正数来说，最高位为0，其余各位代表数值本身(以二进制表示)，如+42的补码为 00101010。
    - 对负数而言，把该数绝对值的补码按位取反，然后对整个数加1，即得该数的补码。如-42的补码为 11010110。
- (00101010 按位取反 11010101 +1=11010110 )

## 第四节 Java技术基础

### 4、Java程序基本结构、数据类型和运算符

#### ■ Java中的基本运算符

- 位运算符
- 关系运算符

位运算符	&	2&3 结果为 2
		2 3 结果为 3
	~	~2 结果为-3
	^	2^3 结果为 1
	<<	3<<2 结果为 12
	>> (带符号位)	-5>>2 结果为-2
	>>> (不带符号位)	((byte)-5)>>>2 结果为 62
关系运算符	==	2==3 结果为 false
	!=	2!=3 结果为 true
	>	2>3 结果为 false
	<	2<3 结果为 true
	>=	3>=3 结果为 true
	<=	2<=3 结果为 true

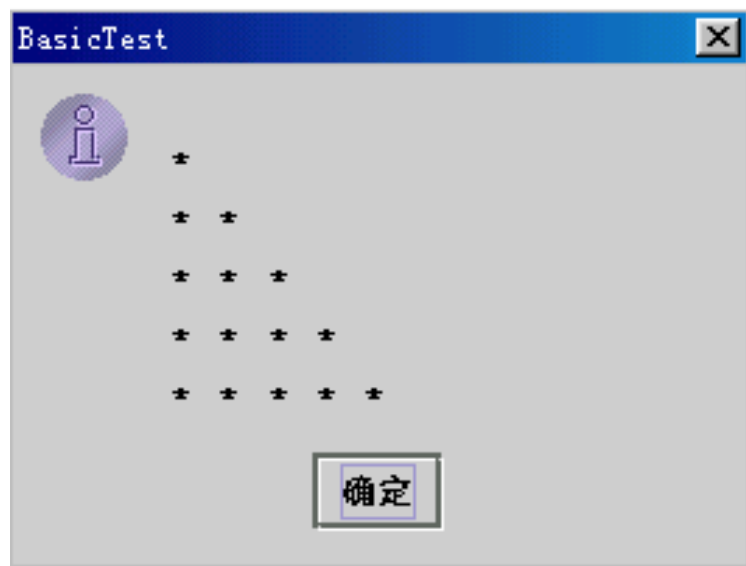
# 第四节 Java技术基础

## 5、Java表达式和基本语句

- 赋值语句
- 条件语句if
- 多重选择语句switch
- 循环语句for和while

表 1-9 一个使用了循环语句的 Java 程序

行号	程序代码
1	//filename:BasicTest.java
2	import javax.swing.JOptionPane;
3	
4	public class BasicTest {
5	public static void main(String[] args)
6	{
7	String output="";
8	nextrow:
9	for(int row=1;row<=5;row++) {
10	output+="\n";
11	for(int column=1;column<=10;column++) {
12	if(column>row)
13	continue nextrow;
14	output+="*  ";
15	} //end inner for
16	} //end outer for
17	JOptionPane.showMessageDialog(null,output,"BasicTest",JOptionPane. INFORMATION_MESSAGE);
18	System.exit(0);
19	} //end main
20	} //end class



# 第四节 Java技术基础

## 6、Java类

### ■ 类的概念

- 类(class)是一种数据实体（对象）的抽象，是特殊的数据类型，含有数据成员，分别为数据（属性）与方法（行为），数据和方法被封装在类中，外面的数据实体看不到它们的具体实现，这种程序编码原则在Java中又称为数据隐藏（封装）。
- 分为系统类和用户自定义的类。

# 第四节 Java技术基础

## 6、Java类

表 1-10 用户自定义的类 Student

行号	程序代码
1	public class Student {
2	private String No;
3	private String name;
4	private int age;
5	public Student(String No, String name, int age) {
6	this.No=No;
7	this.name=name;
8	this.age=age;
9	}
10	public void Display() {
11	System.out.println("No="+No);
12	System.out.println("name="+name);
13	System.out.println("age="+age);
14	}
15	public void Display(int x) {
16	if(x==0) System.out.println("No="+No);
17	if(x==1) System.out.println("name="+name);
18	if(x==2) System.out.println("age="+age);
19	if(x==4) System.out.println("Not correct!");
20	}
21	}

定义构造函数，  
为对象成员赋初值

方法名称相同，参数不同。  
这种现象称为方法的重载

# 第四节 Java技术基础

## 6、Java类

### ■ 类的概念

Student
No:String name:String age:int
Display():void Display(int):void

- 第1行定义了一个类Student，public为所定义的类的属性，class是定义类时所用的关键词，Student为所定义类名，第2、3、4、5、10、15行中的private或public为数据成员的属性，在此类中包含3个**属性**和3个**方法**，分别为属性No、name、age和方法Student()和Display()；
- 第10行和第15行分别定义了2个Display()方法，但它们的参数不同，这种现象在面向对象的程序设计语言中称为方法的“**重载**”。第5行上的Student()方法和类名相同，是一个特殊的方法，称为**构造函数**，构造函数没有函数返回类型说明。在一个类中可以不写出构造函数，如果不写出构造函数，则程序在编译时，Java编译器将自动为它添加一个默认的构造函数。

# 第四节 Java技术基础

## 6、Java类

### ■ 类的继承与组合

- 在Java中，一个类可以在已有类的基础上进行扩展，称为**类的继承**。例：类GraduateStudent可继承Student类，Student类称为父类或超类，GraduateStudent为子类或扩展类。
- 当一个类继承另一个类时，在父类中标识为public或protected的数据成员（属性或方法）将被子类继承，但标识为private的数据成员不能直接被继承。
- GraduateStudent类能继承父类中的Display()和Display(int)这2个方法，但由于子类中也有一个名为Display(int)的方法，它将“覆盖”父类中的Display(int)方法，因而使得实际能继承的方法只有父类中的Display()。GraduateStudent类最终将具有属性department和方法GraduateStudent()、Display()、Displayall()。

2025/3/4 □ 通过继承，类的功能得到了扩展。

# 第四节 Java技术基础

## 6、Java类

### ■ 类的继承与组合

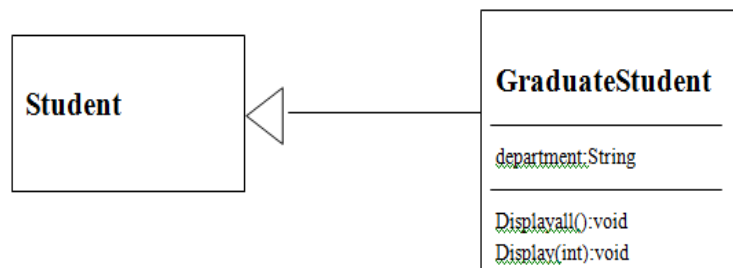


表 1-11 类 GraduateStudent 继承 Student

```
public class GraduateStudent extends Student
{
    private String department;
    public GraduateStudent(String s1, String s2, int x, String dept) {
        super(s1,s2,x);
        department=dept;}
    public void Displayall() {
        Display();
        System.out.println("department="+department);
    }
    public void Display(int x) {
        if(x==0||x==1||x==2)    super.Display();
        if(x==4) System.out.println("department="+department);
    }
}
```



## 第四节 Java技术基础

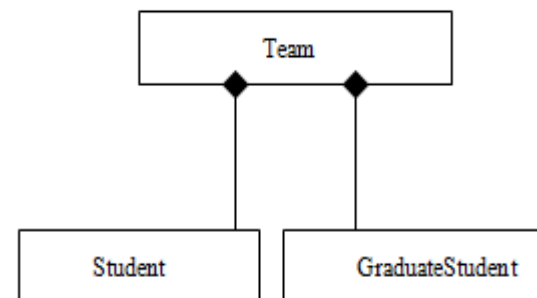
### 6、Java类

#### ■ 类的继承与组合

- 类Student和类GraduateStudent可组合成类Team

表 1-12 Student类和 GraduateStudent 类组合成 Team 类

```
public class Team {  
    public Student monitor;  
    public GraduateStudent secretary;  
    public Team(){  
monitor=new Student("21B122","zhang3",12 );  
secretary=new GraduateStudent("21B126","wang5",20,"B department");  
    }  
}
```



# 第四节 Java技术基础

## 6、Java类

### ■ 类的多态

- 在Java中，用户所定义的类都是某个类的子类，Java中最顶级的父类为Object系统类。类通过继承关系不断的向下“生长”，从而具有越来越多的“子孙”，各级子孙对象所具有的行为与属性都不完全相同，在程序中要区分这些不同层次的类较为困难。
- **类的多态性**能够减轻这方面的负担。多态性能使程序将不同层次的子孙对象看作是同一个超类的不同对象。这些对象具有超类中的相同方法与属性，在程序中能加以引用；在实际运行时，执行的是各自拥有的方法。
- 多态性使开发人员能够设计和实现易于扩展的系统。

## 第四节 Java技术基础

### 6、Java类

#### ■ 类的多态

40	public class MorphTest {
41	public static void main(String[] args)
42	{
43	GraduateStudent stud1=new
	GraduateStudent("22B210","zhang3",20,"imformation");
44	Student stud2=new Student("22B215","wang5",12);
45	Student stud1Ref=stud1;
46	stud1Ref.Display(4);    //执行的是 GraduateStudent 类中的
	Display()
47	stud2.Display(4); //执行的是 Student 类中的 Display()
48	}
49	}//end class MorphTest

# 第四节 Java技术基础

## 7、Java源程序说明

- Java源程序是由一个或多个类组成的。在Java源程序中，不管存在多少个类，有且只能有一个public类，源程序主文件名应与此public类的类名相同，扩展名只能为.java；
- 不管是顶级类还是内部类，在用javac编译后，都会产生各自独立的.class文件。顶级类对应的.class文件名与各自的类名相同，如Time1.class，TimeTest1.class；有名内部类对应的.class文件名为它所在的外部类名加上一个美元符号，再加上内部类的名字，如OuterClass\$InnerClass.class；匿名内部类对应的.class文件名为它所在的外部类名加上一个美元符号，再加一个数字序号，数字序号为匿名内部类的编号，如OuterClass\$#.class，这里#表示数字序号；
- 包实际上是一个树行目录结构，在编译时若不存在的话，则编译程序将会自动产生这样的目录结构，将.class文件放在包中，也就是将.class文件放在相应的树型目录结构中，一个Java源程序中只能有一个包语句，且只能是第一条语句；
- 将编译后产生的.class文件放入包中后，在以后的程序中如要使用这样的.class文件，可以使用import语句，import只能位于包语句的后面（若存在包语句的话）。包放的位置，则由环境变量classpath指定。

## 第四节 Java技术基础

### 8、数组

- 数组是包含多个连续存储位置的对象，每个数组(即数组单元)均用于保存**同一类型**的数据。
  - `int[] b = new int[10];` //创建一个数组
  - `int[] p = {2,3,5,7,11,13,17,19,23,29};` //声明一个数组p, 并同时对它进行初始化

## 第四节 Java技术基础

### 9、Java中不同类型数据之间的转换

#### ■ 实数转换为整数（整数转换为实数）

表 1-18 实数转为整数示例

行号	程序代码
1	<code>public class Int_Real {</code>
2	<code>    public static void main(String[] args)    {</code>
3	<code>        int x;</code>
4	<code>        x=(int) (10*Math.random());</code>
5	<code>        System.out.println("The random Integer="+x);</code>
6	<code>    }</code>
7	<code>}</code>

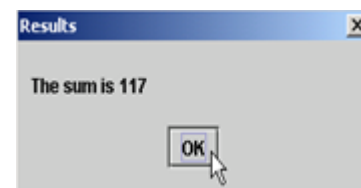
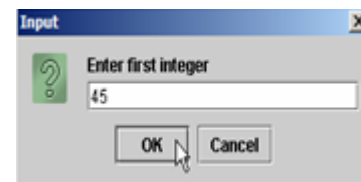
## 第四节 Java技术基础

### 9、Java中不同类型数据之间的转换

#### ■ 字符串转换为其它类型

表 1-21 两个整数相加

行号	程序代码
1	<code>import javax.swing.JOptionPane;</code>
2	
3	<code>public class Addition {</code>
4	<code>    public static void main( String args[] ) {</code>
5	<code>        String firstNumber, secondNumber;</code>
6	<code>        int number1, number2, sum;</code>
7	<code>        firstNumber = JOptionPane.showInputDialog( "Enter first integer" );</code>
8	<code>        secondNumber = JOptionPane.showInputDialog( "Enter second integer" );</code>
9	<code>        number1 = Integer.parseInt( firstNumber );</code>
10	<code>        number2 = Integer.parseInt( secondNumber );</code>
11	<code>        sum = number1 + number2;</code>
12	<code>        JOptionPane.showMessageDialog( null, "The sum is " + sum, "Results",</code> <code>JOptionPane.PLAIN_MESSAGE );</code>
13	<code>        System.exit( 0 );</code>
14	<code>    }</code>
15	<code>}</code>



# 本章小结

- 电子商务与电子商务系统的概念与区别
- 电子商务系统与管理信息系统的区别
- 电子商务系统的三个特性
- 电子商务系统的体系结构
- 电子商务系统的技术架构
- J2EE平台
- Java语言基础





**Thank You !**