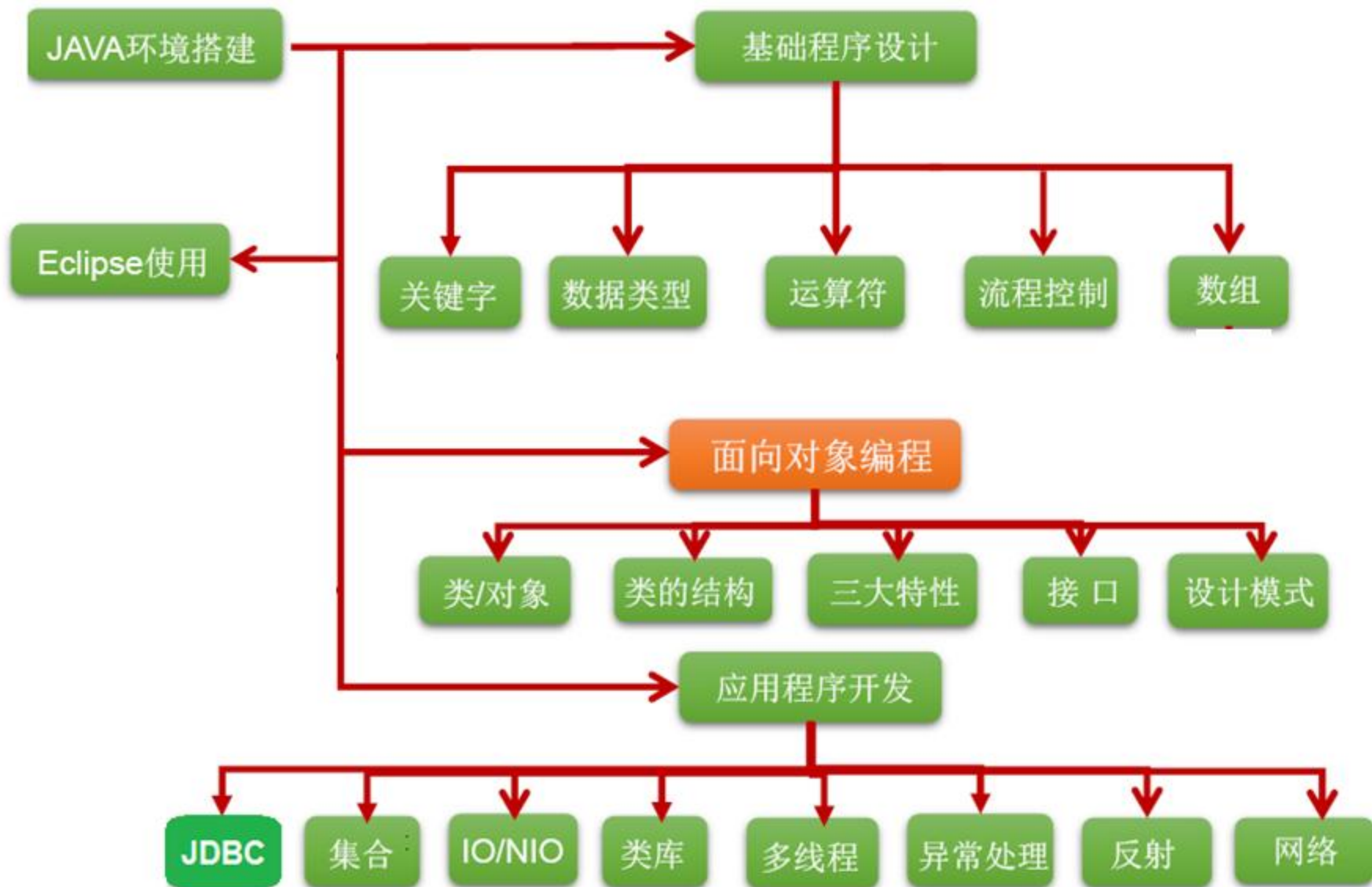


Java语言与系统设计

第1讲 概述

- 课程知识体系
 - JAVA语言基础
 - JAVA程序基础
-

1. 课程知识体系



□ 主要内容:

- 1、 编程语言核心结构： 变量、基本语法、分支、循环、数组
 - 2、 Java面向对象的核心知识： 面向对象、封装、继承、多态、接口
 - 3、 高级应用程序： 异常、集合、I/O、多线程、网络编程、数据库、设计模式
 - 4、 课程实验
-

□ 课时安排:

1、课时安排: 课堂40+上机8

2、参考教材:

[1] Java编程思想 (第4版)

[2] Effective Java 中文版 (第2版)

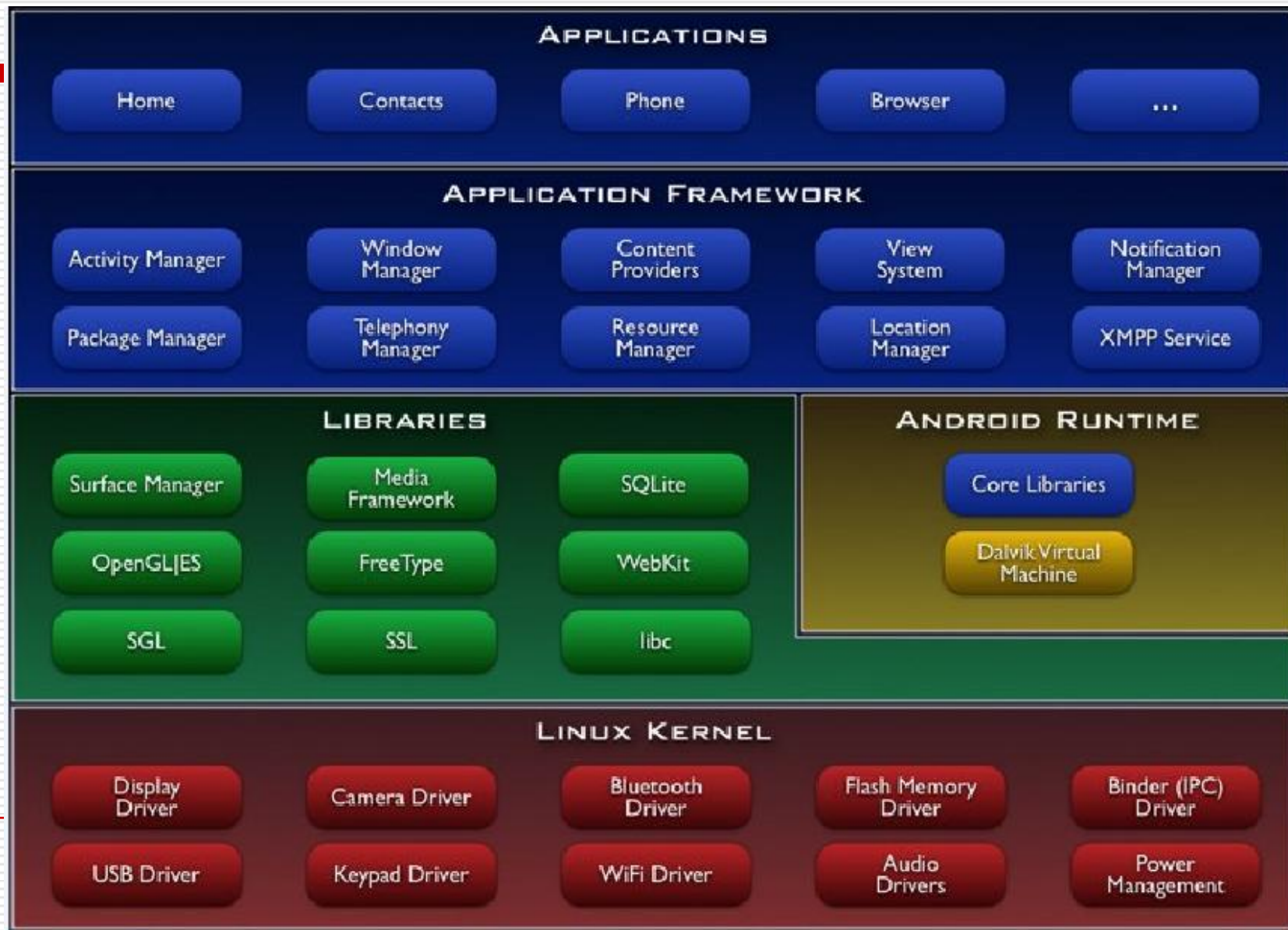
3、实验安排: 8次实验

2. JAVA语言基础

- ▣ **第一代语言：机器语言**。指令以二进制代码形式存在。
- ▣ **第二代语言：汇编语言**。使用助记符表示一条机器指令。



2. JAVA语言基础

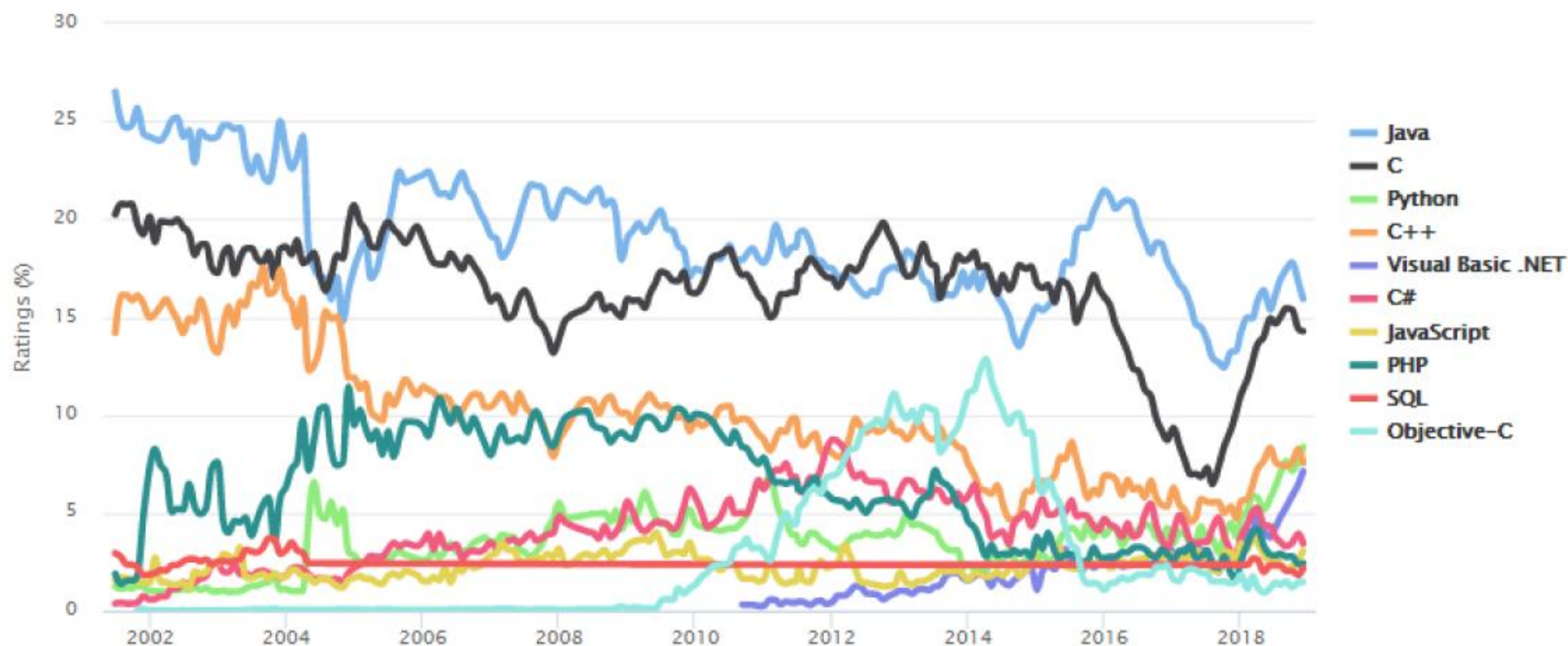


2. JAVA语言基础

□ 为什么要学习JAVA? TIOBE排名第1

Dec 2018	Dec 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.932%	+2.66%
2	2		C	14.282%	+4.12%
3	4	▲	Python	8.376%	+4.60%
4	3	▼	C++	7.562%	+2.84%
5	7	▲	Visual Basic .NET	7.127%	+4.66%
6	5	▼	C#	3.455%	+0.63%
7	6	▼	JavaScript	3.063%	+0.59%
8	9	▲	PHP	2.442%	+0.85%
9	-	▲	SQL	2.184%	+2.18%
10	12	▲	Objective-C	1.477%	-0.02%

为什么要学习JAVA?



2. JAVA语言基础

□ 历史

- 80年代初，美国斯坦福大学的几位学生创办了斯坦福大学网络公司(Stanford University Network)，即 Sun。
- 1992年，James Gosling等开发了适合网络编程和跨平台编程的新型语言 Oak，能使用 Java小程序（applet）在Web浏览器中运行。
- 1994年，Oak更名为Java。



2. JAVA语言基础

- 1991年 Green项目，开发语言最初命名为Oak (橡树)
- 1994年，开发组意识到Oak 非常适合于互联网
- 1996年，发布JDK 1.0，约8.3万个网页应用Java技术来制作
- 1997年，发布JDK 1.1，JavaOne会议召开，创当时全球同类会议规模之最
- 1998年，发布JDK 1.2，同年发布企业平台J2EE
- 1999年，Java分成J2SE、J2EE和J2ME，JSP/Servlet技术诞生
- 2004年，发布里程碑式版本：JDK 1.5，为突出此版本的重要性，更名为JDK 5.0
- 2005年，J2SE -> JavaSE，J2EE -> JavaEE，J2ME -> JavaME
- 2009年，Oracle公司收购SUN，交易价格74亿美元
- 2011年，发布JDK 7.0
- 2014年，发布JDK 8.0，是继JDK 5.0以来变化最大的版本
- 2017年，发布JDK 9.0，最大限度实现模块化
- 2018年3月，发布JDK 10.0，版本号也称为18.3
- 2018年9月，发布JDK 11.0，版本号也称为18.9

2. JAVA语言基础

▣ JAVA体系平台

Java SE(Java Standard Edition)标准版

支持面向桌面级应用（如Windows下的应用程序）的Java平台，提供了完整的Java核心API，此版本以前称为J2SE

Java EE(Java Enterprise Edition)企业版

是为开发企业环境下的应用程序提供的一套解决方案。该技术体系中包含的技术如：Servlet、Jsp等，主要针对于Web应用程序开发。版本以前称为J2EE

Java ME(Java Micro Edition)小型版

支持Java程序运行在移动终端（手机、PDA）上的平台，对Java API有所精简，并加入了针对移动终端的支持，此版本以前称为J2ME

Java Card

支持一些Java小程序（Applets）运行在小内存设备（如智能卡）上的平台

2. JAVA语言基础

■ JAVA应用

- 企业级应用：主要指复杂的大企业的软件系统、各种类型的网站。
 - Android平台应用：Android应用程序使用Java语言编写。
 - 大数据平台开发：各类框架有Hadoop、spark等框架以及工具大多数用Java编写而成。
 - 移动领域应用：主要表现在消费和嵌入式领域各种小型设备上的应用用Java开发。
-

2. JAVA语言基础

□ JAVA与C、C++比较:

- 相同点: Java是从C语言和C++语言继承了许多成份, Java语言的变量声明, 操作符形式, 参数传递, 流程控制等方面和C语言、C++语言完全相同。
 - 不同点: Java是一个纯粹的面向对象的程序设计语言, 它继承了C++语言面向对象技术的核心。Java舍弃了C语言中容易引起错误的指针 (以引用取代)、多重继承 (以接口取代) 等特性。
-

2. JAVA语言基础

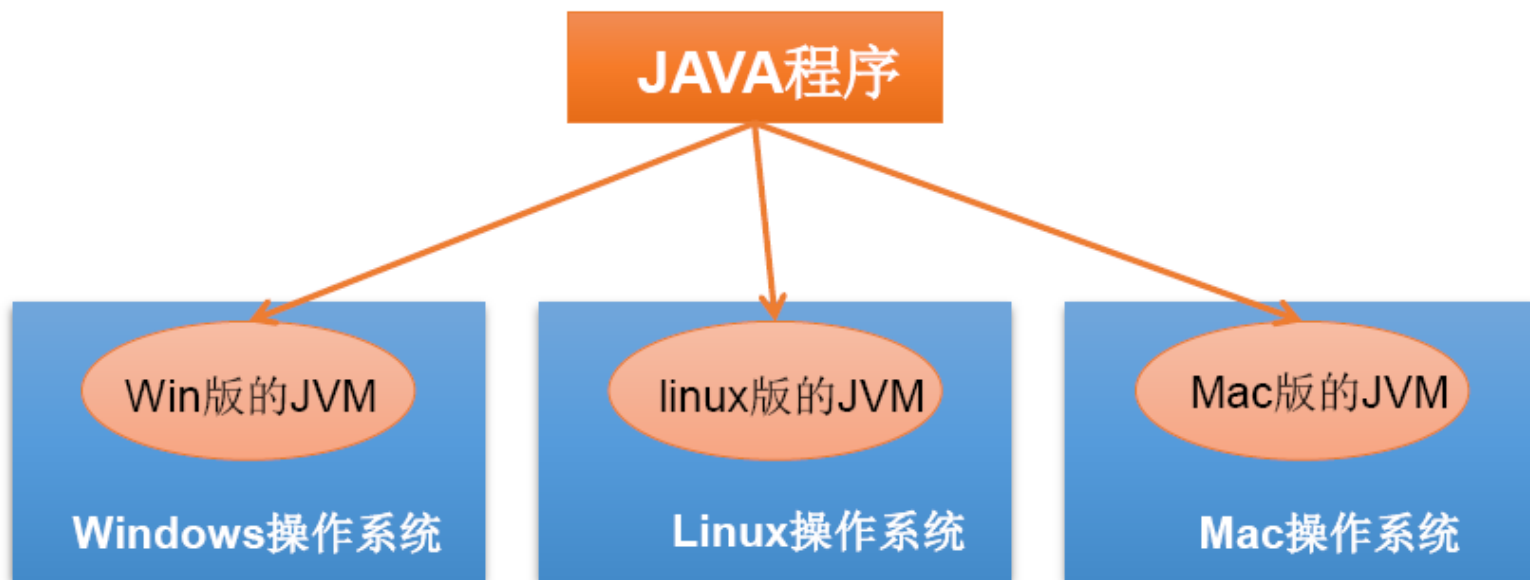
□ JAVA特点：

- 特点一：面向对象。两个基本概念：类、对象；三大特性：封装、继承、多态
- 特点二：健壮性。吸收了C/C++语言的优点，但去掉了其影响程序健壮性的部分（如指针、内存的申请与释放），提供了一个相对安全的内存管理和访问机制
- 特点三：跨平台性。通过Java语言编写的应用程序在不同的系统平台上都可以运行。“Write once , Run Anywhere”。

2. JAVA语言基础

□ JAVA跨平台性:

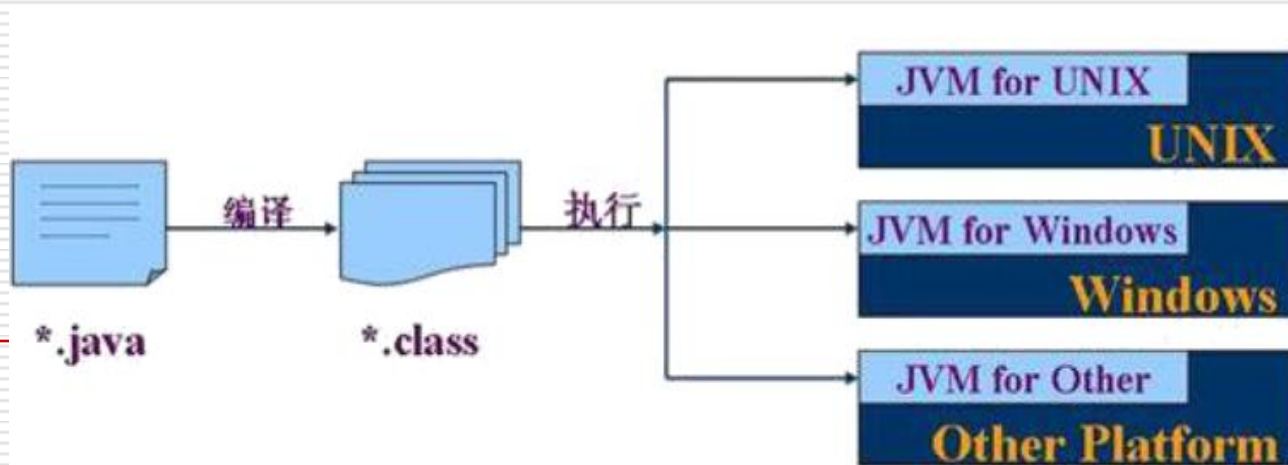
原理：只要需要在需要运行java 应用程序的操作系统上，先安装一个Java虚拟机(JVM Java Virtual Machine) 即可。由JVM来负责Java程序在该系统中的运行。



2. JAVA语言基础

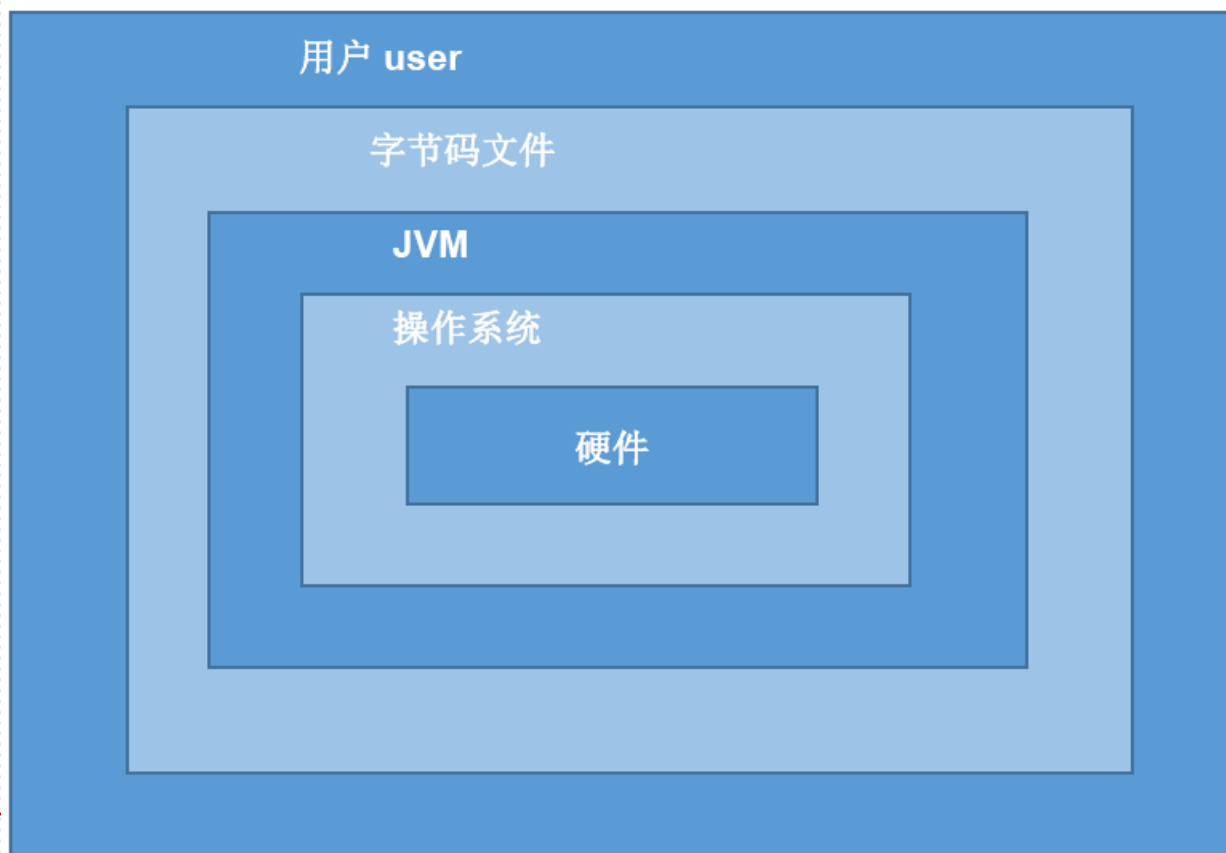
□ JAVA虚拟机JVM:

- JVM是一个虚拟的计算机，具有指令集并使用不同的存储区域。负责执行指令，管理数据、内存、寄存器。
- 对于不同的平台，有不同的虚拟机。只有某平台提供了对应的Java虚拟机，Java程序才可在此平台运行。



2. JAVA语言基础

□ JAVA虚拟机JVM:



3. JAVA程序基础

□ JDK和JRE

JDK(Java Development Kit Java开发工具包)

JDK是提供给Java开发人员使用的，其中包含了java的开发工具，也包括了JRE。所以安装了JDK，就不用在单独安装JRE了。

➤ 其中的开发工具：编译工具(javac.exe) 打包工具(jar.exe)等

JRE(Java Runtime Environment Java运行环境)

包括Java虚拟机(JVM Java Virtual Machine)和Java程序所需的核心类库等，如果想要运行一个开发好的Java程序，计算机中只需要安装JRE即可。

简单而言，使用JDK的开发工具完成的java程序，交给JRE去运行。

3. JAVA程序基础

□ JDK下载安装：

- JDK从www.oracle.com或java.sun.com 下载。
 - 安装后记得设置环境变量中的路径path变量：
JDK的安装路径\bin
 - 命令行下javac测试参数信息。
-

3. JAVA程序基础

□ JAVA程序编译:

- 1. 编写源代码(.java 文件)。一般用文本编辑工具即可。
- 2. 编译源代码, javac命令生成.class 文件。
- 3. 在命令行中, java命令运行.class 文件



3. JAVA程序基础

□ 简单程序：

```
public class test  
{  
  
}
```

// 类名称为test，要和文件名一致

3. JAVA程序基础

■ 简单程序：

```
public class test
{
    public static void main(String[] args)
    {
    }
    //main主方法是程序的入口
}
```

3. JAVA程序基础

□ 简单程序：

```
public class test
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    } // System.out.println在屏幕输出字符
}
```

3. JAVA程序基础

关键点:

- ▣ Java源文件以“java”为扩展名。源文件的基本组成部分是类（class），如本例中的test类。
- ▣ Java应用程序的执行入口是main()方法。它有固定的书写格式：`public static void main(String[] args) {...}`
- ▣ Java语言严格区分大小写。Java方法由多条语句构成，每个语句以“;”结束。大括号都是成对出现的，缺一不可。
- ▣ 一个源文件中最多只能有一个public类。其它类的个数不限，~~如果源文件包含一个public类，则文件名必须按该类名命名。~~

3. JAVA程序基础

□ 常见错误:

- 1、主类名和文件不一致
- 2、注意必要的分号
- 3、类文件名写错，类文件不在当前路径下，或者不在classpath指定路径下

◆ 学会看编译器报告的错误信息

3. JAVA程序基础

□ 注释:

- 单行注释: //
 - 多行注释: /* */
 - 对于单行和多行注释, 被注释的文字, 不会被JVM (java虚拟机) 解释执行。
-

3. JAVA程序基础

□ JAVA API文档

- API (Application Programming Interface,应用程序编程接口) 是Java提供的基本编程接口。
- Java语言提供了大量的基础类, 因此Oracle也为这些基础类提供了相应的API文档, 用于告诉开发者如何使用这些类, 以及这些类里包含的方法。
- 下载:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

3. JAVA程序基础

■ JAVA API文档

The screenshot shows the Java Platform API Specification website. The interface is divided into several sections, with red boxes and labels highlighting specific areas:

- 包列表区 (Package List Area):** Located on the left side, it contains a list of packages under the heading "All Classes". The visible packages are: java.applet, java.awt, java.awt.color, java.awt.datatransfer, java.awt.dnd, java.awt.event, java.awt.font, and java.awt.geom.
- 类列表区 (Class List Area):** Located below the package list, it contains a list of classes under the heading "All Classes". The visible classes are: AbstractAction, AbstractAnnotationValueVisitor, AbstractAnnotationValueVisitor7, AbstractBorder, AbstractButton, AbstractCellEditor, AbstractCollection, AbstractColorChooserPanel, AbstractDocument, AbstractDocument.AttributeCor, AbstractDocument.Content, AbstractDocument.ElementEdit, AbstractElementVisitor6, AbstractElementVisitor7, AbstractExecutorService, AbstractInterruptibleChannel, and AbstractLayoutCache.
- 详细说明区 (Detailed Description Area):** Located on the right side, it contains the main content of the API specification. It includes the title "Java™ Platform, Standard Edition 7 API Specification", a brief description of the document, and a table of packages.

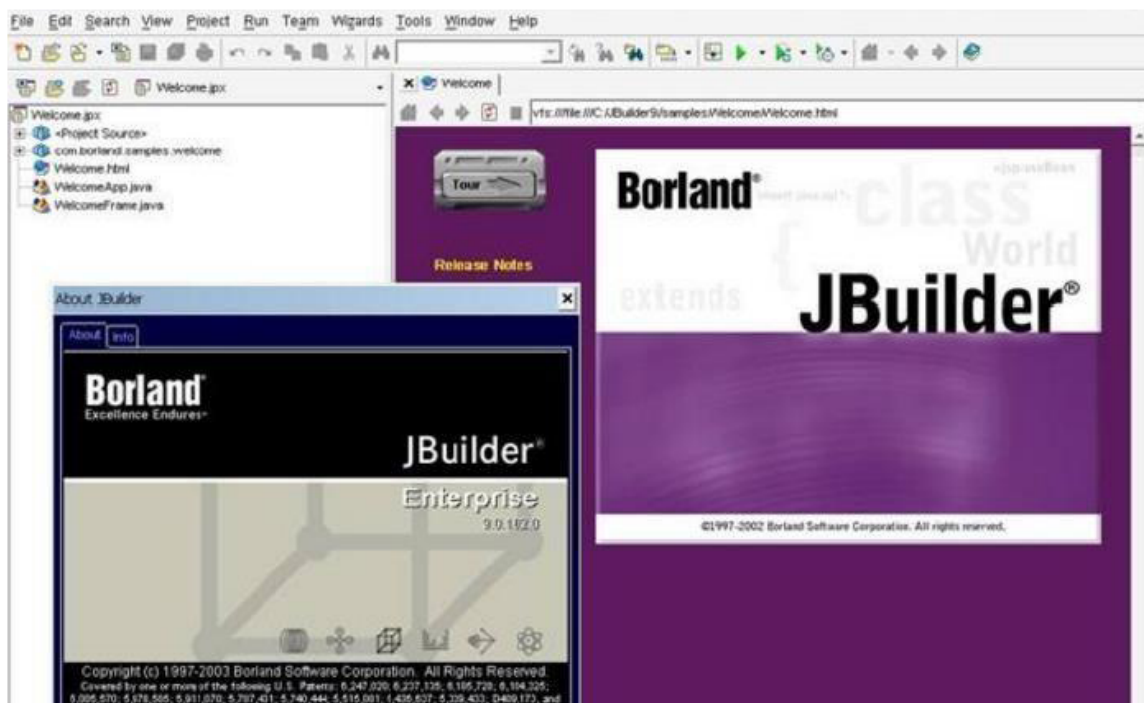
The table of packages is as follows:

Package	Description
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects.

3. JAVA程序基础

□ JAVA 开发环境

JBuilder是Borland公司开发的针对java的开发工具，使用JBuilder将可以快速，有效的开发各类java应用。Jbuilder支持各种应用服务器。Jbuilder与Inprise Application Server紧密集成，同时支持WebLogic Server，支持EJB 1.1和EJB 2.0，可以快速开发J2EE的电子商务应用。支持远程调试和多线程调试，调试器支持各种JDK版本



3. JAVA程序基础

▣ JAVA 开发环境

NetBeans是一款用Java编写的开源IDE。既可用于Java开发，也支持其他语言，特别是PHP、C/C++，和HTML5。NetBeans开发环境提供了丰富的产品文档和培训资源以及大量的第三方插件。



3. JAVA程序基础

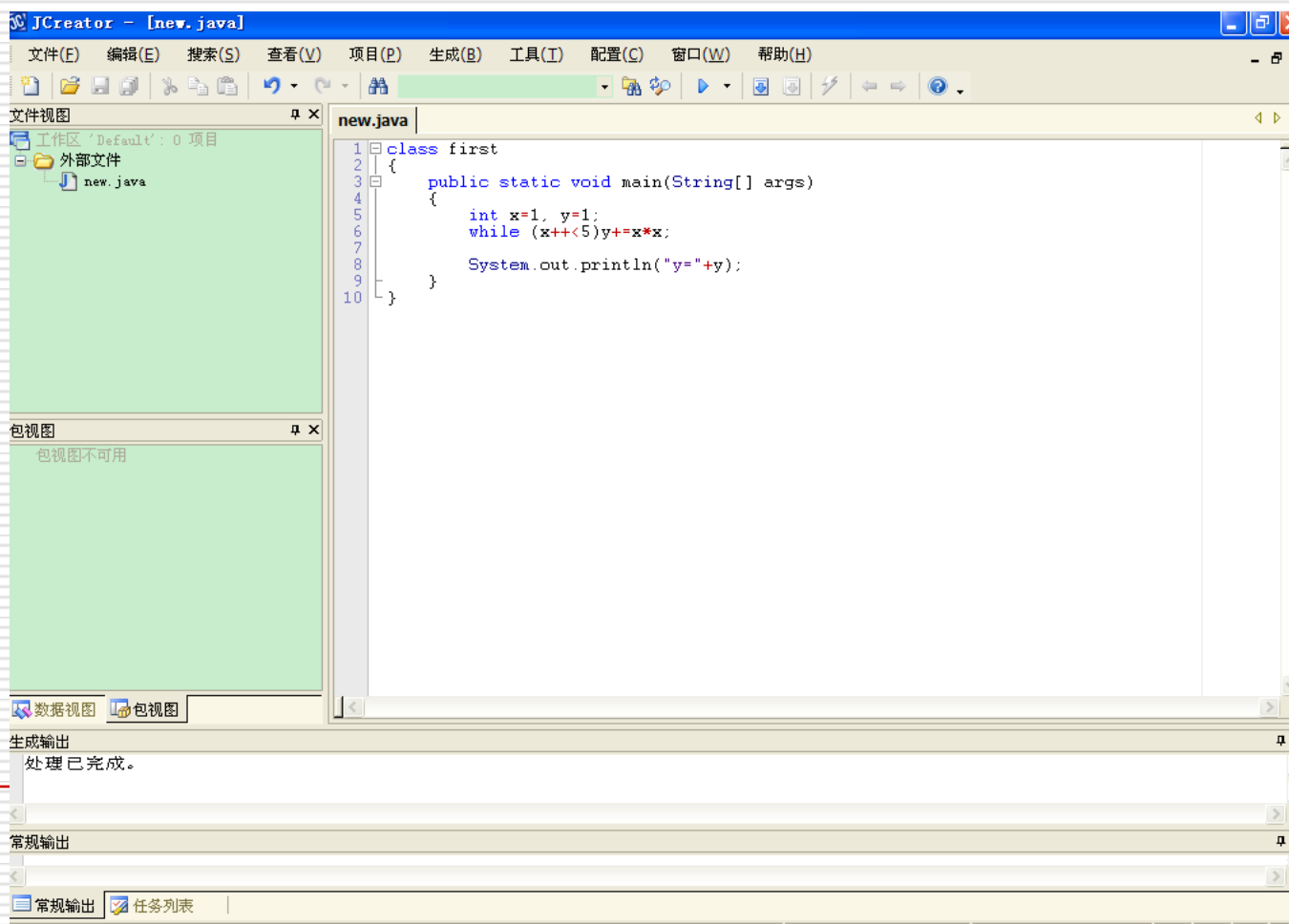
▣ JAVA 开发环境

Eclipse应该是大多数Java程序员使用的第一个IDE。众所周知的、最流行、也最受欢迎的Java开发工具。优点很多：免费、更新快、代码智能化、ANT构建等，拥有众多插件，完全免费、有中文版、上手比较快。缺点也非常明显，安装插件麻烦、插件对版本要求比较严格。



3. JAVA程序基础

□ JAVA 开发界面（以JCreator为例）



实例：

程序1:

```
class Example_1 {    //主类
    public static void main(String[] args) {
        int x, y=0; //定义两个变量x和y
        for(x=1; x<5; x++) y+=x*x; //把1至4平方值累加到y
        System.out.println("y="+y); //输出y值
    }
}
```

实例：

程序2：

```
class Example_2 {    //定义含有主方法的类
    public static void main(String[] args) {
        int x=1, y=1;    //定义变量和初始化
        while(x++<5) y+=x*x;    //累加计算，第1次累加2*2
        System.out.println("y="+y);    //输出y的值： y=55
    }    //1+2*2+3*3+4*4+5*5=1+4+9+16+25=55
}
```

练习题：

1. 具有相同属性和服务的一组对象的模板称为（ ）。
A. 程序 B. 类 C. 代码 D. 概念
2. 使用JDK编译Java程序时使用的命令是（ ）。
A. java B. javac C. appletviewer D. javadoc
3. Java开发工具包简称为（ ）。
A. JKD B. JRK C. JDK D. JRE
4. 编译Java程序后生成的面向JVM的字节码文件的扩展名是（ ）。
A. .java B. .class C. .obj D. .exe

□ 如果变量a中有一个值，变量b中有一个值，如何将两个变量中的值互换？

```
public class test1
{ public static void main(String[] args)
{int a = 10;
int b = 20;
int temp;
temp = a;
a = b;
b = temp;
System.out.println("a="+a);
System.out.println("b="+b); }
}
```

可以不用
临时变量吗?



```
public class test1
{ public static void main(String[] args)
{  int a = 10;
  int b = 20;
  a = a + b;
  b = a - b;
  a = a - b;
  System.out.println("a="+a);
  System.out.println("b="+b);  }
}
```
