



第1章 概 述

1.1 面向对象编程思想

1.1.1 面向对象编程语言

Java是一种典型的面向对象编程语言，除此以外还有：C++、C#等编程语言。

Java程序能在各种不同的计算机平台上运行，非常**适合编写网络应用程序**。随着移动互联网技术和移动智能终端的发展，在**Android操作系统的智能终端**上的开发语言也是Java。

实例

```
public class Main{  
    Public static void main(String[] args){  
        int x;  
        x = 6;  
        System.out.println( x*x*3.14);  
    }  
}
```

Java程序的基本构造

```
public class Main{
```

类 (Class)

```
    Public static void main(String[] args){
```

方法 (Method)

```
        System.out.println( “Hello World ” );
```

```
    }
```

```
}
```

Java程序的基本构造

`public class Main{` 类的名称：源文件名为【Main.java】
首字母须大写

`Public static void main(String[] args){`

`System.out.println(“Hello World ”);`

`}`

`}`

Java程序的基本构造

```
public class Main{  
    public static void main(String[] args){  
        System.out.println( "Hello World " );  
    }  
}
```

The diagram illustrates the hierarchical structure of the Java code using red arrows. A long arrow on the left points from the opening curly brace of the `public class Main{` line down to the closing curly brace of the class. A shorter arrow points from the opening curly brace of the `main` method down to its closing curly brace. Another arrow points from the `main` method's opening curly brace down to the `System.out.println` statement, indicating its scope within the method.

注释

```
public class Main{  
    // 注释1  
    Public static void main(String[] args){  
        /* 注释2 */  
        System.out.println( “Hello World ” );  
    }  
}
```

1.1.2 类和对象



小王, 23岁

人: 类 (class)

属性: 姓名、年龄等

行为: 吃饭、睡觉等



小徐, 19岁



小李, 25岁

1.1.2 类和对象

类（class）是具有**相同或相似性质**的**对象的抽象**。**对象的抽象是类**；**类的具体化就是对象**，也可以说类的实例是对象。

类具有**属性**（如姓名name、年龄age），属性是对象状态的抽象。

类具有**操作**，是对象行为的抽象（如eat（）方法）。

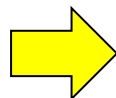
【例1.1】 Person类

```
public class Person{  
    String name;  
    int age;  
    void eat(){  
        System.out.println( “This is a method. ” );  
    }  
}
```

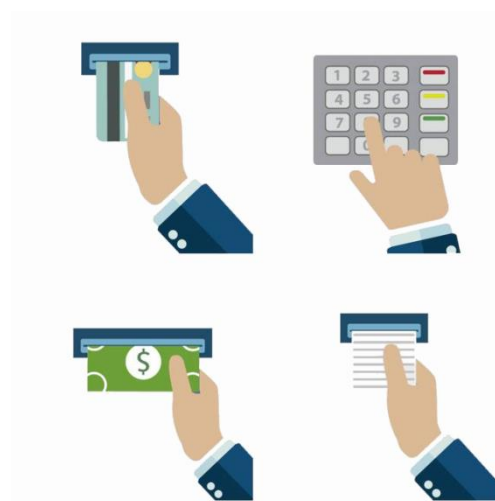
1.1.3 面向对象的特征

- ◆ 封装
- ◆ 继承
- ◆ 多态

1. 封装



存款、取款、查询等



具体里面是如何实现的不向外提供

1. 封装

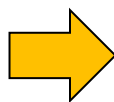
面向对象程序设计的核心就是封装。在面向对象程序设计中，封装是通过把一组**数据**和与数据有关的**操作**集合放在一起形成**对象**来实现的。**对象**通过特定的**接口**与外部发生联系，内部的具体细节被隐藏起来，**对外是不可见的**。封装的目的就是**防止非法访问**，用户只能通过**对象的接口**利用对象提供的服务，**看不到其中的具体实现细节**。

2. 继承

人：类 (class)

属性：姓名、年龄等

行为：吃饭、睡觉等



学生：类 (class)

属性：学号、班级等

行为：学习等



教师：类 (class)

属性：教工号等

行为：教学等



小李，20岁



小徐，19岁

2. 继承

为了提高代码的重用性，面向对象程序提供了继承策略。继承是在扩展现有类的基础上定义新类的过程。当一个新定义的类是基于原有类时，新类将共享原有类的属性和方法，并且还可以添加新的特性，其中新类被称为原有类的子类，原有类称为父类。

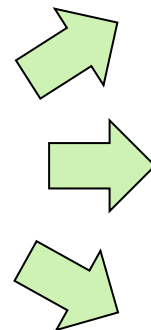
【例1.2】 Student类

```
class Student extends Person{  
    String studentNo;  
    void takeClass(){  
        .....  
    }  
}
```


3. 多态



去运动



跑步

游泳

踢足球

在面向对象的程序设计中，通过**多态性**来支持这一思想。多态使得**相同的消息被不同的对象**接收时，可能导致**不同的动作**。

面向对象编程语言通过类**继承**过程中的**方法覆盖**和**方法调用时的动态绑定**来实现多态性，从而达到**不同的对象按照自身的需求**对同一消息进行**正确处理**的目的。

1.1.4 Java语言的特点

1. 跨平台(JVM)

2. 可移植性

3. 简单

4. 面向对象

5. 分布式

6. 解释性

7. 健壮性

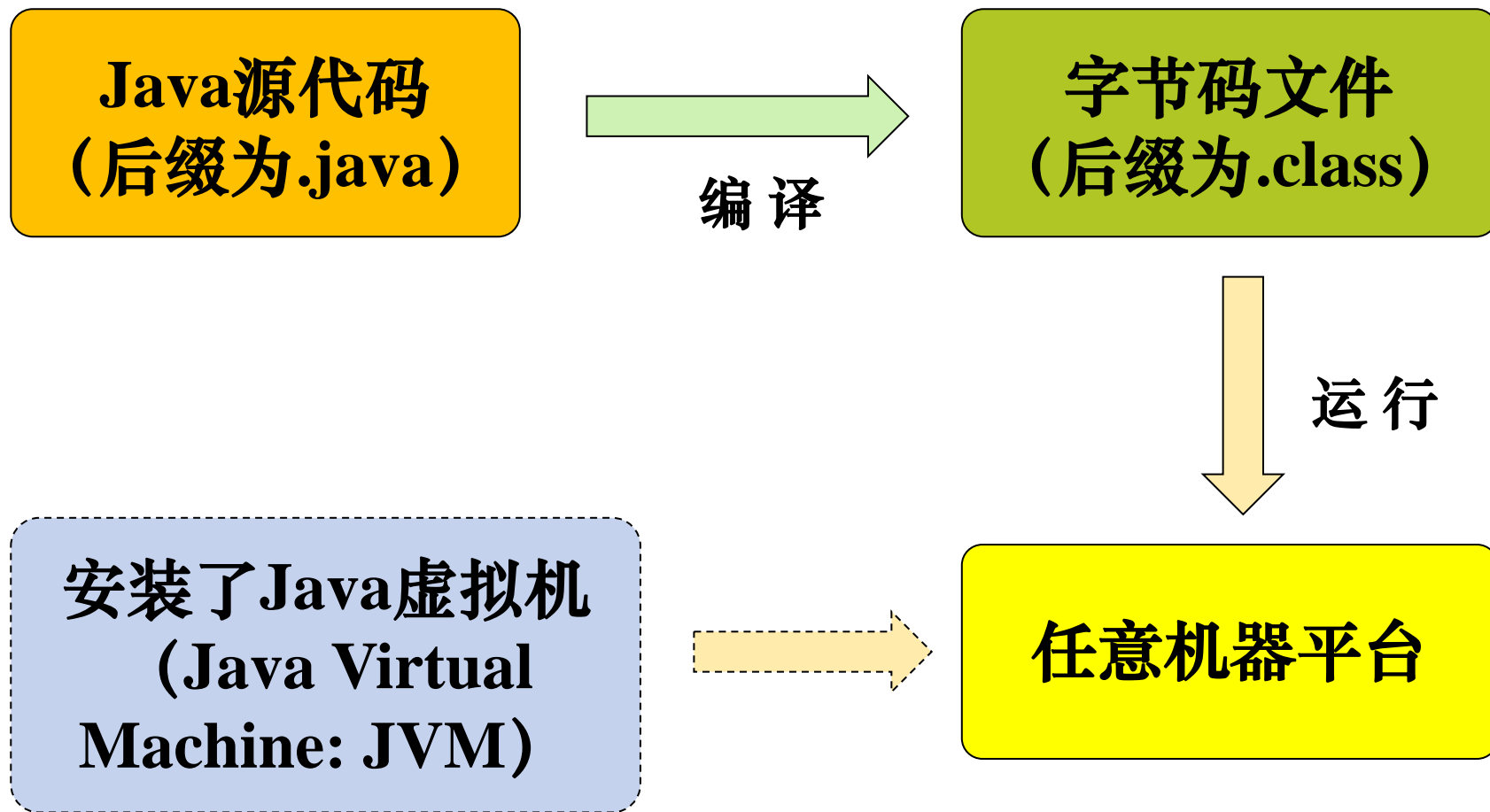
8. 安全性

9. 体系结构中立

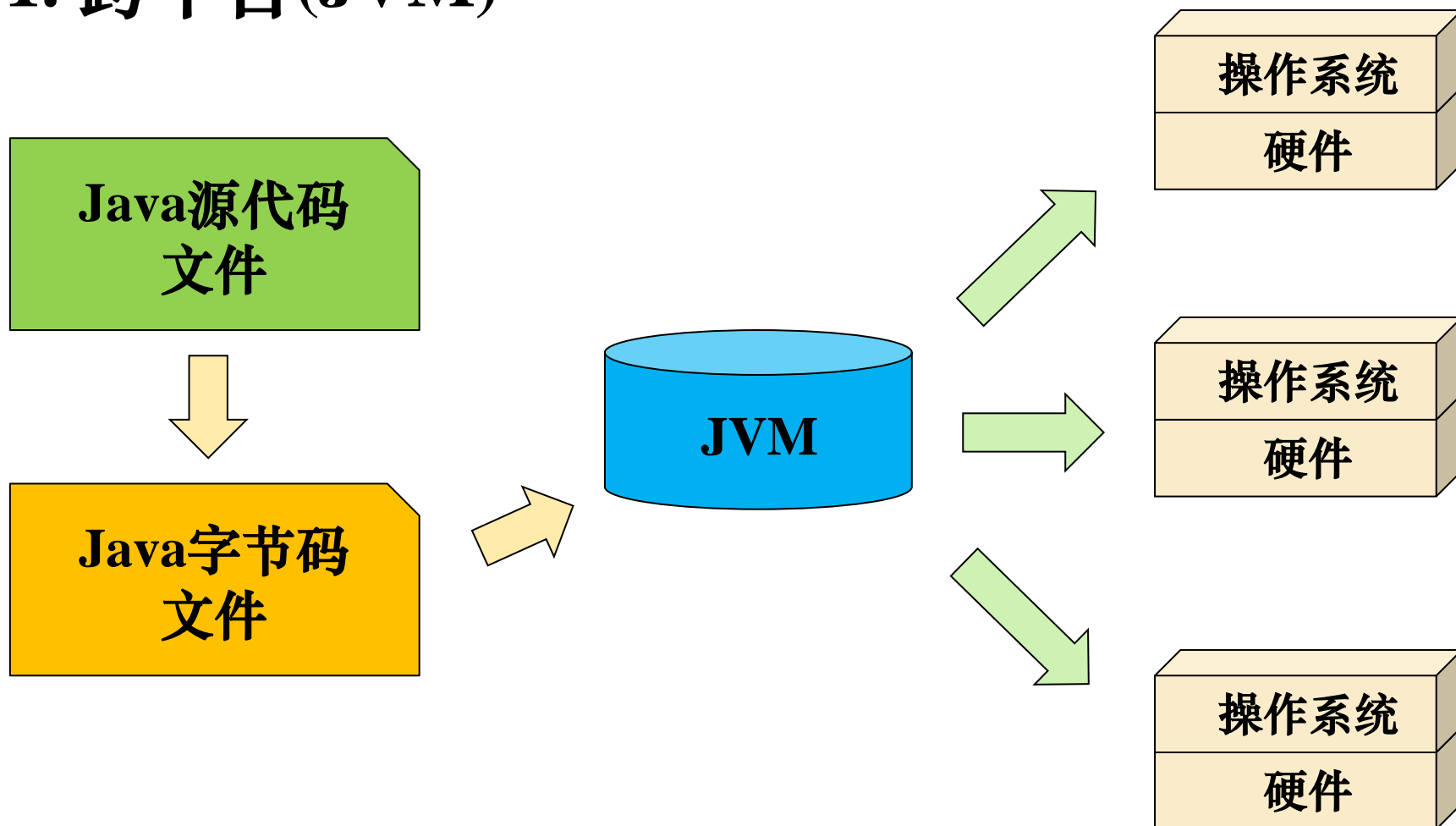
10. 多线程

11. 动态性

1. 跨平台(JVM)

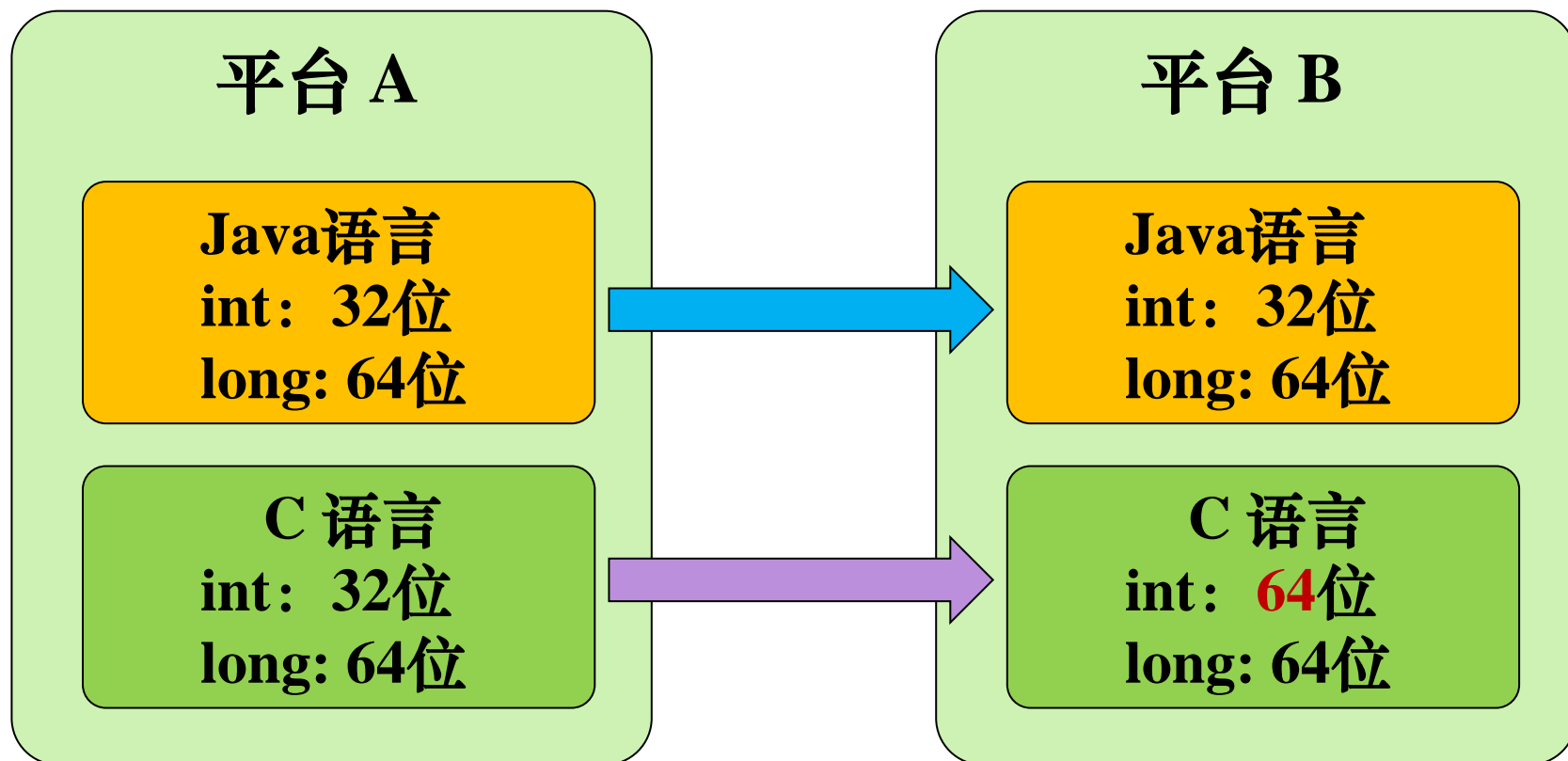


1. 跨平台(JVM)



2.可移植性

体系结构中立性使得Java程序不需要重新编译就能在不同的平台上运行。



2.可移植性

体系结构中立性使得Java程序不需要重新编译就能在不同的平台上运行。

◆ Java系统提供的**类库**可以访问不同平台的基本操作系统。

◆ **Java编译程序使用Java语言编写的**，这使得Java系统本身也具有可以执行。

3.简单

Java语言是从C++语言衍生而来的，但它在C++语言的基础上做了重大改进。

- ◆ Java语言抛弃了C++语言中许多比较复杂的部分，包括指针、运算符重载、多重继承等。
- ◆ Java语言还提供了一个被称为“垃圾回收器”的机制来自动进行内存管理。

4. 面向对象

C++语言为了兼容C语言，因此不是一种纯粹的面向对象语言，而Java则是纯粹面向对象的。



5. 分布式

Java语言提供的类库可以处理TCP/IP协议，利用Java语言可以方便地通过统一资源定位器在网络上访问其他对象，取得用户需要的资源。因此Java语言非常适合Internet和分布式环境下的编程。

6. 解释性

Java源程序经过**编译**后生成**字节码文件**，**字节码**是一种**虚拟的机器指令代码**，不针对特定的机器。运行时，Java解释器负责将字节码**解释成本地机器指令代码**。Java解释器包含在Java虚拟机中。

7. 健壮性

Java语言抛弃了C、C++语言中的指针数据类型，**不允许对内存进行直接操作**。Java语言的**垃圾回收器**能自动进行内存管理，防止程序员在管理内存时产生错误。同时，**Java虚拟机**在编译和运行程序时对**可能出现的问题进行检查**。

8. 安全性

Java语言特有“沙箱”机制，使得网络和分布式环境下的Java程序不会充当攻击本地资源的病毒或其他恶意操作的传播者，确保了安全。

9. 体系结构中立

Java源程序编译后生成的字节码文件与平台无关，只要机器上安装了Java解释程序，就能在任意机器平台上运行，而其他程序设计语言（如C、C++等）则需要针对特定的机器重新编译。

10. 多线程

- ◆ 线程是指一个程序可以**独立运行**的片段。
- ◆ 多线程处理能使**同一程序中的多个线程同时运行**。
- ◆ Java语言内建立多线程机制，同时提供了**同步机制**保证对**共享资源**的**正确操作**，因此Java语言的多线程编程相对容易。

11. 动态性

- ◆ 在一个类中可以自由地增加新的方法和数据成员而不会影响到原来使用该类的程序的运行。
- ◆ 运行Java程序时，每个类文件只在有必要时（第一次使用这个类时）才被加载。
- ◆ Java运行环境中，始终存在着一个系统级的线程，专门跟踪内存的使用情况，定期检测出不再使用的内存并进行自动回收，避免了内存的泄露，也减轻了程序员的工作量。
- ◆ Java虚拟机采用的是“沙箱”运行模式，即把Java程序的代码和数据都限制在一定的内存空间里执行，不允许程序访问该内存空间以外的内存，保证了程序的安全性。

1.2 编写Java程序的步骤

一、准备Java开发环境

1. 下载、安装JDK

2. 配置环境变量

二、准备编程工具

1.2.3 编写第一个Java程序

【例1.1】：使用Eclipse开发HelloWorld程序

第一步：新建工程，并设置工程名为FirstProject。

第二步：新建类，Java程序是由多个类组成的，类名称HelloWorld，一般首字母大写，此时将自动产生一个名为“HelloWorld.java”的文件。

第三步：进入编程环境，开始写代码，文件保存为HelloWorld.java。

【例1.3】 HelloWorld类

```
public class HelloWorld {  
    public static void main(String[]args){  
        System.out.println("This is my first java program:");  
        System.out.println("Hello World");  
    }  
}
```

1.3 标准I/O

- ◆ **输入、输出**是所有高级语言都必须提供的基本系统功能。
- ◆ Java语言中输入、输出是以**流的方式**实现的。
- ◆ Java程序可**通过键盘输入与外界进行简短的信息交换**，同时也可能通过文件与外界进行**任意数据形式的信息交换**。

【例1.4】 从控制台获取字符并输出

```
import java.io.*;

public class KeyboardIO {

    public static void main(String [ ] args) throws
        IOException{

        System.out.println("Enter a Char:");
        char i = (char) System.in.read();
        System.out.println("your char is:"+i);
    }

}
```


【例1.5】从控制台接收一个字符串并输出

```
import java.io.*;

public class KeyboardIO {
    public static void main(String [] args) throws
        IOException{
        BufferedReader br = new BufferedReader(
            new InputStreamReader(System.in));
        String str = null;
        System.out.println("Enter a string:");
        str = br.readLine();
        System.out.println("the string is: "+str);
    }
}
```

【例1.6】 从键盘获取数据并输出

```
import java.util.Scanner; // 导入Scanner类
public class KeyboardIO {
    public static void main(String[]args){
        // 以下部分借用java.util.Scanner类获取键盘输入并输出
        Scanner sc=new Scanner(System.in);
        System.out.println("请输入姓名: ");
        String s=sc.nextLine();
        System.out.println("身高: ");
        float h=sc.nextFloat();
        System.out.println("年龄: ");
        float age=sc.nextInt();
        System.out.println("个人信息如下: ");
        System.out.println("姓名: "+s+" 身高:"+h+" 年龄: "+age);
    }
}
```

运行结果如下所示：

请输入姓名：

Linux

身高：

157.7

年龄：

60

个人信息如下：

姓名：Linux 身高：157.7 年龄：60.0

习题

1. 面向对象程序设计语言的特点是什么？
2. 谈谈类和对象的联系和区别。
3. 编译和运行Java程序的命令是什么？
4. MyDemo.java文件中定义了两个类，分别是：Person和Student。编译后将生成几个类文件（**.class）？文件名是什么？

作业

1. 写程序实现以下功能：输入专业、姓名，显示如下信息：

“欢迎 ***专业的***同学开始学习面向对象程序设计！”

2.编写一个Java应用程序，使其运行后能够在屏幕上显示如下图案：



2.编写一个Java应用程序，使其运行后能够在屏幕上显示如下图案：

*

System.out.print (“\n”); //回车