



# 信息工程系

## 教 案

课程名称： 面向对象程序设计

教 师： 谢梓莹

总学时： 72

理论学时： 36

实训学时： 36

上课班级： 物联网应用技术 241

授课学期： 2025-2026 学年第一学期

# 课程思政

## 一、课程思政总体目标

本门课程面向对象程序设计培养了学生的编程逻辑思维和工匠精神，同时建立工程理论与职业素养。理解技术发展与社会文明的互动关系，增强制度自信与技术自信。

## 二、思政元素与专业知识点融合设计

### 1. 类与对象→工匠精神与社会分工

通过强调代码的规范性（如命名规则、注释要求），类比工匠精神的严谨态度。以“类”的模块化设计映射社会分工协作。

### 2. 继承与多态→文化传承与创新

以“继承”比喻传统文化与现代技术的融合。通过“多态”特性引申解决问题的多样化思维，培养创新意识。

### 3. 接口与抽象→契约精神与职业道德

通过接口的契约性，强调程序员遵守行业标准（如开源协议、数据安全法）。以模块间松耦合映射团队协作中的边界意识。要求学生设计符合《个人信息保护法》的用户数据接口，讨论技术实现与法律约束的平衡。

4. 引入“大数据杀熟”真实事件，理解算法歧视的社会危害，培养科技向善意识，分析代码层面的实现逻辑。

5. 引入游戏中青少年模式讨论技术社会影响分析，讨论如何让科技造福人类社会规避科技对青少年的负面影响。

# Java 概述和环境搭建

**教学目标：**了解 Java 的特点；掌握 JDK 的下载与安装，能够独立下载并安装 JDK；掌握环境变量的配置，能够独立配置 Path 环境变量；掌握第一个 Java 程序的开发，能够独立编写并运行 HelloWorld 程序；掌握 IDEA 的下载、安装与启动，能够独立安装并启动 IDEA；掌握使用 IDEA 进行程序开发，能够使用 IDEA 编写并运行“HelloWorld”程序；掌握 IDEA 调试工具的使用方法，能够在程序中正确设置断点并进行程序的调试。

**课型：**新授

**教学重点：**JDK 的下载与安装；第一个 Java 程序；使用 IDEA 进行程序开发。

**教学难点：**Java 程序的运行机制；IDEA 调试工具。

**教学方法：**概念讲解、案例演示

**教学节数：**理论 2 节、实训 2 节

## 教学过程

### 一 Java 的概念和特点

1. 计算机语言是人与计算机之间进行交流的一种语言形式，计算机语言种类繁多，按照其抽象层次通常将计算机语言分为机器语言、汇编语言、高级语言三大类。Java 是计算机语言中的一种高级语言，于 1995 年首次发布。作为一种支持跨平台和完全面向对象的编程语言，Java 已经成为企业级应用程序和互联网应用开发的首选语言之一。
2. Java 的 3 个技术平台：Java SE (Standard Edition, 标准版)、Java EE (Enterprise Edition, 企业版)、Java ME (Micro Edition, 微型版)。
3. Java 语言的 6 个特点：简单性、面向对象、安全性、跨平台性、支持多线程、分布性。

### 二 Java 开发环境安装说明

想要开发 Java 程序需要先搭建 Java 环境，Java 环境主要指的是支撑 Java 语言开发和执行的一系列工具和组件，搭建 Java 环境包括下载和安装 JDK (Java Development Kit, Java 开发工具包)，以及配置系统环境变量。

1. Java 应用程序需要自己的一套运行环境，通常把这个运行环境成为 JDK (Java

Development Kit)。

2.设置环境变量。打开我的电脑/计算机 -> 属性 -> 系统属性。打开系统高级设置，打开环境变量。

JAVA\_HOME:JDK 安装路径

Path: java 编译程序和运行程序的目录。

CLASS\_PATH: 配置类库文件的位置。

JAVA\_HOME 的值 : D:/Java (根据自己的安装地址填写)

Path 的值: %JAVA\_HOME%\bin;

CLASS\_PATH 值: .;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

3.使用集成开发环境开发 Java 程序。在实际项目开发中，使用记事本编辑代码受到许多限制，例如无法提供实时调试和排错功能、缺乏项目管理和构建工具等，难以满足开发需求。正所谓“工欲善其事，必先利其器”，为了提高程序的开发效率，大部分开发人员通常都会使用集成开发环境进行 Java 程序的开发。本节将对 Java 常用的开发工具—IntelliJ IDEA 进行讲解。

- (1) 创建 Java 项目
- (2) 创建 Java 类
- (3) 编写程序代码
- (4) 运行程序
- (5) 断点的概念和作用
- (6) 以 Debug 模式启动程序进行调试，介绍 Debug 模式界面各个视图的作用。
- (7) 介绍 IDEA 的 Debug 模式界面中一些常用的调试按钮以及它们的作用。

## 实验作业

上机练习主要针对本章中需要重点掌握的知识点，以及在程序中容易出错的内容进行练习，通过上机练习可以考察学生对知识点的掌握情况。

**上机：**（考察知识点为第一个 Java 程序、使用 IDEA 进行程序开发）

**形式：**独立完成

**具体要求：**

- (1)在计算机上安装 JDK，并配置环境变量。
- (2)使用记事本开发第一个 Java 程序，并在命令行编译运行。
- (3)在计算机上安装 IDEA，并在 IDEA 中创建项目，开发第一个 Java 程序。  
使用 IDEA 调试工具对 Java 程序进行调试。

# Java 语言的基本语法

**教学目标：**了解 Java 程序的基本结构，能够简述 Java 的基本组成结构和各个部分的含义；熟悉 Java 中的注释，能够简述 Java 中注释的类型和各自的作用；熟悉关键字和标识符，能够简述标识符和关键字的含义和使用规则；掌握 Java 的数据类型，能够简述 Java 的所有基本数据类型；掌握变量的定义，能够定义基本数据类型的变量；掌握数据类型转换，能够简述自动类型转换和强制类型转换的区别；掌握 Java 中的运算符，能够使用算术运算符、赋值运算符、比较运算符、逻辑运算符和三元运算符对程序中的数据 and 表达式进行运算。

**课型：**复习+新授

**教学重点：**Java 的数据类型；变量；算术运算符；赋值运算符；比较运算符；逻辑运算符；三元运算符。

**教学难点：**Java 的数据类型、变量、运算符的应用。

**教学方法：**概念讲解、案例演示

**教学节数：**理论 4 节、实训 4 节

## 教学过程

### 一 Java 程序的基本结构、注释

1. Java 程序的基本组成部分.:

(1)包声明（使用关键字 package）

(2)导入类（使用关键字 import）

(3)定义类（使用关键字 class）

(4)入口方法（主方法 main()）

(5)主体代码（编写主体代码需要注意的问题）

2. Java 中的注释:

(1) 单行注释

(2) 多行注释

(3) 文档注释

注释：//                    注释一行

/\*       \*/       注释一行或多行

/\*\* 可用 javadoc 命令转化为 HTML 文件\*/

### 3. 关键字和标识符

#### 1) 定义标识符应遵循规则

- `_, $` 不作为变量名，方法名开头。
- 变量名、方法名首单词小写，其余单词只有首字母大写，如 `anyVariableWorld`。
- 接口名、类名首单词第一个字母大写。
- 常量：完全大写。

#### 2) Java 中的关键字有哪些？

<code>abstract</code>	<code>double</code>	<code>int</code>	<code>strictfp **</code>
<code>boolean</code>	<code>else</code>	<code>interface</code>	<code>super</code>
<code>break</code>	<code>extends</code>	<code>long</code>	<code>switch</code>
<code>byte</code>	<code>final</code>	<code>native</code>	<code>synchronized</code>
<code>case</code>	<code>finally</code>	<code>new</code>	<code>this</code>
<code>catch</code>	<code>float</code>	<code>package</code>	<code>throw</code>
<code>char</code>	<code>for</code>	<code>private</code>	<code>throws</code>
<code>class</code>	<code>goto *</code>	<code>protected</code>	<code>transient</code>
<code>const *</code>	<code>if</code>	<code>public</code>	<code>try</code>
<code>continue</code>	<code>implements</code>	<code>return</code>	<code>void</code>
<code>default</code>	<code>import</code>	<code>short</code>	<code>volatile</code>
<code>do</code>	<code>instanceof</code>	<code>static</code>	<code>while</code>

### 4. 常量与变量。

#### 1) Java 中的常量。

- 整形常量（二进制、八进制、十进制、十六进制）、浮点数常量、字符常量、字符串常量、布尔常量、`null` 常量。
- 常量的声明：`final` 常量类型 常量标识符=常量值；

#### 2) Java 中的变量。

布尔型变量（`boolean`，取值：`true`，`false`）

文字型变量（`char`、`String`）

整数类型变量（`byte`，`short`，`int` 和 `long`）

浮点型（`float` 和 `double`）

引用数据类型（类（class）、数组、枚举（enum）等）

## 二 变量、自动类型转换、强制类型转换

1. 变量：变量的声明、变量的定义、变量的作用域

2. 自动类型转换。

◆ 自动类型转换需要满足的条件：

(1)两种数据类型彼此兼容，即二者在语义上或逻辑上是相关联的。

(2)目标类型的取值范围大于源类型的取值范围。

◆ 常见的自动类型转换

(1)整数之间的转换

(2)整数类型转换为 float 类型

(3)其他类型转换为 double 类型

◆ 表达式类型自动提升的规则

(1)byte、short、char 类型自动提升为 int 类型

(2)表达式的数据类型自动提升到表达式中取值范围最大的操作数的数据类型。

3. 强制类型转换：类型彼此不兼容、或者目标类型取值范围小于源类型时，自动类型转换无法完成，这时就需要强制类型转换。强制类型转换可能导致数据截断或精度丢失的问题。

## 三 运算符

1. 算术运算符。

◆ 算术运算符的作用（对数值类型的数据进行加、减、乘、除等运算）

◆ 列举 Java 中的算术运算符

(1)+：加、正号

(2)-：减、负号

(3)\*：乘

(4)/：除

(5)%：取模（求余数）

(6)++：自增（前缀、后缀）

(7)--：自减（前缀、后缀）

◆ 算术运算符使用时需要注意的问题。

(1)除法运算时操作数类型和结果类型的关系

(2)取模运算时结果的正负取决于被模数

(3)自增自减在单独使用和混合使用时的区别

## 2. 赋值运算符

◆ 赋值运算符的作用（主要用于给变量赋值）

◆ 列举 Java 中常用的赋值运算符

(1)=：赋值

(2)+=：加等于

(3)-=：减等于

(4)\*=：乘等于

(5)/=：除等于

(6)%=：模等于

◆ 使用赋值运算符需要注意的问题

(1)赋值运算符的左边只能是常量，右边可以是常量、变量或表达式。

(2)赋值运算符“=”与数学中的等号不同，后者强调左右相等之意，而前者无此含义。

## 3. 比较运算符

◆ 比较运算符的作用（用于对两个数值或变量进行比较）

◆ 列举 Java 中的比较运算符。

(1)==：等于

(2)!=：不等于

(3)<：小于

(4)>：大于

(5)<=：小于或等于

(6)>=：大于或等于

◆ 强调不能将比较运算符“==”误写成赋值运算符“=”。

## 4. 逻辑运算符

◆ 逻辑运算符的作用（用于对布尔型的数据进行操作，结果仍是布尔值）

- ◆ 列举 Java 中的逻辑运算符
  - (1)!: 逻辑非
  - (2)&: 逻辑与
  - (3)&&: 短路与
  - (4)|: 逻辑或
  - (5)||: 短路或
  - (6)^: 逻辑异或
- ◆ 讲解运算符“&”和“&&”的区别以及“|”和“||”的区别。

## 5. 三元运算符

- ◆ 三元运算符的作用（用于根据条件的真假来选择返回不同的值）
- ◆ 三元运算符的语法格式
- ◆ 条件表达式 ? 表达式 1 : 表达式 2

## 实验作业

### 【案例 2-1】学生成绩单

在 Java 程序中，如果想读取用户输入到命令行中的内容，可以使用 Scanner 类，Scanner 是 Java 内置的一个实用工具类，用于读取用户输入和文件等数据源。下面是使用 Scanner 类读取命令行输入数据的示例代码。

```
Scanner sc = new Scanner(System.in);  
int count = sc.nextInt();
```

在上述代码中，第 1 行代码创建一个 Scanner 对象 sc，用于从命令行中键入的字符流中读取数据，第 2 行代码用于从输入流中读取下一个整数，即读取命令行中输入的内容并将其赋值给整数变量 count。

现有三名学生的考试成绩需要录入系统并打印出成绩单。本案例要求编写一个模拟录入学生考试成绩的程序，在控制台输入三名学生的三门考试成绩，并计算每个学生的平均分，最后输出三名学生的考试成绩单和平均成绩，以及此次考试学生平均成绩的最高分。学生成绩单程序的运行效果如图 1 所示。

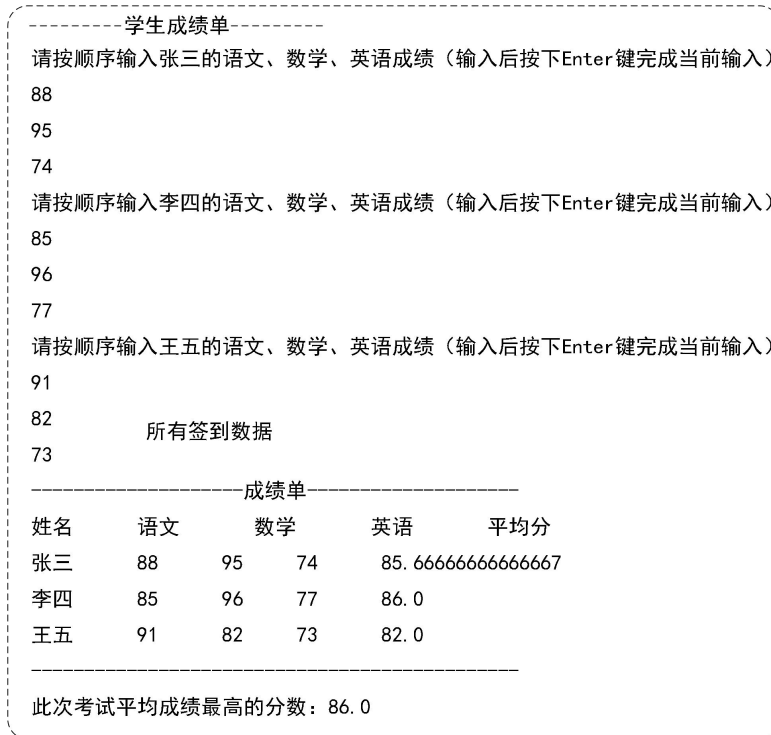


图 1 学生成绩单效果图

## 课堂问答作业

1.二进制数 11010，转换成十进制数是？

2.下列代码的执行结果是？

```
public class Test1{
    public static void main(String args[])
    {
        float t=9.0f;
        int q=5;
        System.out.println((t++)*(--q));
    }
}
```

A. 40      B.40.0      C.36      D36.0

# 程序流程控制语句

**教学目标：**掌握选择结构语句的使用、掌握循环结构语句的使用、掌握数组的创建和操作

**课型：**复习+新授

**教学重点：**if 语句、switch 语句、while 循环语句、for 循环语句、数组的声明和初始化、数组元素的访问和赋值

**教学难点：**对数据类型范围的判断、数据强制转换的理解、控制循环语句应用

**教学方法：**概念讲解、案例演示

**教学节数：**理论 6 节、实训 6 节

## 教学过程

### 一 选择结构语句

1. if 语句的作用。
  - if 语句的三种应用格式
  - (1) if 语句
  - (2) if...else 语句
  - (3) if...else if...else 语句
2. switch 语句的作用。使用 switch 语句需要注意的问题。
  - (1) switch 语句中表达式的数据类型限制。
  - (2) case 条件后的目标值不允许重复
  - (3) 每个 case 后加 break 语句的作用
  - (4) 多个 case 条件后的执行语句相同时，可以只书写一次

### 二 循环结构语句

1. while 循环语句的功能、结构和用法。
2. do...while 循环语句的功能、结构和用法。
3. For 循环语句、for 循环嵌套语句的功能、结构和用法。

### 三 跳转语句

1. 跳转语句的概念。
2. 常见的跳转语句（break 和 continue）。
3. break 语句在 switch 条件语句和循环语句中分别起什么作用。

### 四 数组

- 1.数组的声明、静态初始化、动态初始化。
- 2.数组的访问和赋值、遍历、排序。
- 3.二维数组的定义和使用

## 实验作业

1. 计算  $1+2+3+\dots+999$  等于多少，并于控制台输出。（重点难点：循环语句的运用。）
2.  $2.123$  乘以多少次  $2$  之后结果大于  $344499934993$ ? 控制台输出次数。（重点难点：循环语句的运用、整数类型的取值范围。）
3.  $3.87487$  除以多少次  $2$  之后结果小于  $0$ ? 控制台输出次数。（重点难点：循环语句的运用、强制类型转换。）
- 4.使用 `for` 循环语句计算  $8+88+888+\dots$ 前 10 项之和。（重点难点：循环语句的运用、整数类型的取值范围。）
5. 编写应用程序，输出满足  $1+2+3+\dots+n<8888$  的最大正整数  $n$ 。（重点难点：循环语句的运用、跳出循环变量分析。）

# 方法、类与对象

**教学目标：**掌握方法的定义、调用和重载；类的定义、创建与使用；掌握 this 关键字、static 关键字的应用；

**课型：**复习+新授

**教学重点：**方法的定义、调用和重载；访问控制；创建对象、访问对象的属性和方法；定义构造方法、this 关键字、static 关键字的应用；

**教学难点：**方法的重载及应用；

**教学方法：**概念讲解、案例演示

**教学节数：**理论 4 节、实训 4 节

## 教学过程

### 一 方法的使用和重载

1. 方法定义的概念、格式和调用。
2. 方法的重载。

### 二 面向对象

1. 面向对象的相关概念。
2. 面向对象的特性：封装、继承、多态。

### 三 类的定义

1. 类的概念。
2. 类的成员包括什么。
3. 定义类的语法格式。

### 四 对象的创建、使用和引用传递

1. 对象与类的关系。
2. 创建对象的过程
  - (1) 声明对象
  - (2) 实例化对象
3. 通过对象访问其属性和方法

4. 通过案例代码演示对象的创建、为对象属性赋值和调用对象的方法
5. 提及值传递的概念引出引用传递，案例代码演示值传递和引用传递的区别。

## 五 封装及封装的实现、访问控制

1. 封装的概念和必要性。
2. 访问控制：(1)public (2)protected (3)default (4)private
3. 封装需要进行的操作：(1)成员变量私有化 (2)提供公共访问方法：getter 方法和 setter 方法的命名和使用

## 六 构造方法和 this 关键字

1. 定义构造方法：(1)访问修饰符 (2)返回值类型 (3)构造方法名 (4)参数列表。
2. 构造方法的重载。
3. this 关键字：(1)this 关键字访问成员变量 (2)this 关键字调用成员方法 (3)this 关键字调用构造方法

## 七 static 关键字

1. 静态变量的概念和访问，使用静态变量时需要注意的问题：(1)静态变量访问方式规范 (2)访问修饰符对静态变量可见性的限制 (3)static 修饰变量的限制
2. 静态方法的概念和访问，使用静态方法时需要注意的问题：(1)静态方法的成员访问限制 (2)静态方法的关键字使用限制 (3)静态方法的重写和继承能力差异

## 实验作业

### 一、题目：计算图形面积

要求实现一个计算图形面积的程序，分别包括计算矩形面积，正方形面积，圆形面积和三角形面积。

#### 具体要求：

- (1) 要求展示一个包含操作编号的菜单，用户输入对应的操作编号进行对应图形的面积计算。
- (2) 计算矩形面积需要输入矩形的长和宽。
- (3) 计算正方形面积需要输入正方形的边长。
- (4) 计算圆形面积需要输入圆的直径，其中圆周率  $\pi$  使用 Java 中的数学常量 Math.PI 表示。
- (5) 计算三角形面积需要输入三角形的底和高。

(6) 要求这些图形输入的边长或直径可以为小数。

## 二、题目：打印购物小票

当顾客在超市购物结账时，收银员会提供一张购物小票，上面会列出顾客购买的商品数量、单价、小计和总计等购物信息，以供顾客核对。本案例要求编写一个模拟打印购物小票的程序，收银员输入顾客购买商品的信息后，打印出顾客的购物小票。

### 具体要求：

- (1) 要求先输入顾客购买的商品种类数
- (2) 然后依次输入每种商品的名称、单价和数量
- (3) 输入完成后打印购物小票
- (4) 购物小票要求显示每种商品订单名称、单价、数量、小计和总计，小计是指某种商品一共所花金额。

## 三、几何图形工具类

在实际开发中，为了提高编程效率和代码质量，Java 提供了许多已经封装好的工具类，供开发人员直接调用。例如，用于执行数学计算的工具类 `Math`，它内置了多种用于常见数学运算和操作的方法。工具类通常注重简单易用、单一职责，本案例要求根据工具类的设计思想，编写一个自定义的几何图形工具类，提供常见几何图形相关数据的获取功能，具体功能如下。

计算圆的面积

计算圆的周长

计算圆的弧长

计算扇形面积

计算圆柱表面积

计算圆柱体积

计算圆锥表面积

计算圆锥体积

计算上面的数据时， $\pi$  取近似值 3.1415926535。

### 具体要求：

- (1) 要求在工具类中定义静态变量  $\pi$  和静态代码块，静态代码块提示“几何图形工具类被调用...”
- (2) 程序运行后控制台显示四个图形选项，包括圆形、扇形、圆柱、圆锥。
- (3) 提示用户输入要获取数据的图形
- (4) 选择圆形则需要输入圆形的半径
- (5) 选择扇形则需要输入圆的半径和扇形的圆心角
- (6) 选择圆柱则需要输入圆柱的底半径和高
- (7) 选择圆锥则需要输入圆锥的底半径和高
- (8) 要求每次获取完一个几何图形的数据后，程序支持用户选择继续获取几何图形的数据或者退出程序。

# 继承

**教学目标：**掌握继承的概念、掌握方法的重写、掌握 `super` 关键字的使用、掌握 `final` 关键字的使用；

**课型：**复习+新授

**教学重点：**继承的概念和方法的重写；

**教学难点：**对象类型转换

**教学方法：**概念讲解、案例演示

**教学节数：**理论 4 节、实训 4 节

## 教学过程

### 一 继承的概念

1. 现实生活中子女继承父辈的概念类比面向对象的继承，继承的语法格式。
2. 方法重写的概念和应用。

### 二 `super` 关键字

1. `super` 关键字的使用场景：（1）`super` 关键字访问父类成员变量 （2）（2）`super` 关键字调用父类构造方法
2. `super` 关键字和 `this` 关键字的共同点与不同点。

### 三 `final` 关键字

1. `final` 关键字的含义和作用。（1）`final` 关键字修饰类 （2）`final` 关键字修饰方法  
（3）`final` 关键字修饰变量
2. `final` 关键字修饰类、方法和变量时的使用。

## 实验作业

### 一、题目：交通工具运行模拟

市面上的交通工具类型数不胜数，每种交通工具的行驶方式，行驶速度都不相同。本案例要求设计一个交通工具运行的模拟程序，包括三种交通工具，分别为自行车、电动车和小汽车。交通工具的属性和行为如表 1 所示。

表 1 交通工具的属性和行为

交通工具	共同特征	共同行为	特有方法
Bicycle (自行车)	1.交通工具的名称 2.轮子数量 3.工具颜色 4.行驶速度	1.行驶 2.加速 3.减速	无
ElectricBicycle (电动车)			充电
Car (小汽车)			加油

本案例要求在程序中模拟各个交通工具的行驶、充电或加油、加速、减速等行为。

### 具体要求：

(1) 要求各个交通工具行驶的方法中展示出交通工具的名称、轮子个数、颜色以及行驶速度，例如“2个轮子的黑色自行车正在路上以15km/h的速度行驶，前进方式为踩踏脚蹬”。

(2) 加速和减速时，可以传入参数，具象化速度的增加量和减少量，控制台输出加速后的速度或者减速后的速度。

(3) 充电和加油的方法只输出提示即可。

## 二、题目：公司薪酬系统

某公司为更好地贴合公司业务的多样化发展，将其薪酬制度分为三大类：固定工资制、时薪制、业绩制，每种制度对应一类员工群体。下面对这三种薪酬制度的具体计算方式进行介绍。

- 固定工资制：每个月工资固定不变，但是每个人的固定工资不同。

- 时薪制：工资由工作时长决定，每人每小时工资不同，每月最低工作时长为120小时。

- 业绩制：工资由底薪和提成组成，每个人的底薪都不同，提成根据销售金额和提成比例计算得出，具体如表1所示。

表1 销售金额和提成比例

销售业绩金额	提成比例
5000元以下的部分	12%
5000元~10000元的部分	15%
10000元以上的部分	18%

例如，某个员工是按业绩制核算工资，他的底薪为5000元，当月的销售总额为20000元，那么他这个月的工资计算方式如下。

$$5000+5000*12\%+5000*15\%+10000*18\%=8150 \text{ (元)}$$

同时，为了调动员工积极性，公司提出了考勤制度，每缺勤（迟到或早退）1小时，当月工资扣除 50 元。

针对上述薪酬规则，现在需要做一个公司的薪酬系统，系统支持录入不同薪酬体系的员工信息，并且可以根据员工的 ID 查询当月工资明细。

**具体要求：**

- (1) 要求初始提供三个员工的信息，分别对应三种薪酬制度，方便后续测试。
- (2) 三种薪酬制度的员工共同信息包括，员工 ID、员工姓名、工资和缺勤时长。
- (3) 固定工资制无特有信息，时薪制员工特有信息包括每小时工资和工作时长，业绩制员工特有信息为销售业绩金额。
- (4) 录入员工本月薪酬信息时需要录入每类员工的所有信息

工资明细显示员工 ID、员工姓名、应发工资（未扣款前的工资）、缺勤时长、扣款金额以及实发工资（扣款后的工资）。

# 抽象类和接口

**教学目标：**掌握抽象类的使用、掌握接口的使用、简述抽象类和接口的相同点与不同点；掌握对象类型转换、熟悉 instanceof 关键字的应用；

**课型：**复习+新授

**教学重点：** 抽象类和接口的比较；对象类型转换；

**教学难点：** 抽象类和接口的比较；对象类型转换；

**教学方法：** 概念讲解、案例演示

**教学节数：** 理论 4 节、实训 4 节

## 教学过程

### 一 抽象类

1. 抽象类的概念和特征。
2. 定义抽象方法和抽象类的格式（abstract 关键字）。
3. 抽象类、抽象方法和普通方法的关系。
4. 抽象类的使用。

### 二 接口

1. 接口的概念、与抽象类的关系。
2. 接口的定义（interface 关键字）。
3. 实现接口（implements 关键字）。

### 三 多态

1. 多态的概念。
2. 多态的分类：(1)对象多态 (2)行为多态
3. 对象类型转换。对象类型转换的两种转换形式：（1）向下转型（父类引用变量转换为子类引用变量）（2）向上转型（子类引用变量转换为父类引用变量）
4. instanceof 关键字。使用 instanceof 关键字判断变量的数据类型。

## 实验作业

### 一、题目：打印不同图形

本案例要求编写一个程序，根据用户的输入在控制台打印出不同的图形。要求用户输入要打印的图形，以及图形的尺寸，控制台使用“\*”打印相应图案。要求提供三种图形，正方形、圆形和等边三角形。

**具体要求：**

- (1) 正方形需要提供边长，控制台输出指定边长的实心正方形
- (2) 圆形需要提供半径，控制台输出指定半径的空心圆。
- (3) 等边三角形需要提供高，控制台输出指定高的实心等边三角形。
- (4) 本案例要求使用抽象类实现。

**二、题目：餐厅外卖配送**

随着科技的发展和人们生活节奏的加快，许多餐厅开始提供外卖服务。某餐厅在顾客在外卖软件上下单后，餐厅员工通过该软件管理用户的订单，如果发现存在待配送，则寻找外卖员对订单进行配送。为了节省资金，餐厅也会安排空闲的服务员对待配送的订单进行配送。本案例需要给这个餐厅做一个外卖订单配送的管理系统。

**具体要求：**

- (1) 该订单配送管理系统有两个功能，分别为查询订单和配送订单。
- (2) 系统默认初始化几个订单，订单信息包括订单编号和订单状态，初始状态都为“待配送”。
- (3) 用户选择查询订单时，控制台输出所有订单的信息，若有正在配送的订单，则还需输出该订单的配送员信息。
- (4) 系统默认初始化几个人员，分为外卖员和服务员。
- (5) 用户选择配送订单时，为未配送订单分配外卖员或者服务员进行配送，并输出此时的配送信息，例如“外卖员张三在配送订单编号为 0001 的订单”。
- (6) 选择配送订单时，若所有订单都在配送中，则提示“不存在需要配送的订单”。

# 内部类、异常类

**教学目标：**熟悉内部类；掌握 Java 中异常的处理；

**课型：**复习+新授

**教学重点：**使用 try...catch 和 finally 语句、throw、throws 处理异常

**教学难点：**匿名内部类、try...catch 和 finally 语句、异常的抛出和声明；

**教学方法：**概念讲解、案例演示

**教学节数：**理论 4 节、实训 4 节

## 教学过程

### 一 内部类

1. 成员内部类的定义和使用。
2. 局部内部类的定义和使用。
3. 匿名内部类的定义和使用。
4. 静态内部类的定义和使用。

### 二 异常类

1. Java 提供的内置异常类（Throwable 类的继承体系）。Throwable 类的直接子类：（1）Error：错误类 （2）Exception：异常类
2. Java 虚拟机会创建异常对象去表示异常，从而能够对其进行捕获或处理。
3. try...catch 语句的异常处理流程。finally 代码块的作用。try...catch...finally 语句的异常处理流程。
4. 抛出和声明异常：(1)throw 关键字抛出异常 (2)throws 关键字声明异常
5. 自定义异常类。

## 实验作业

### 一、题目：多媒体播放器

某款多媒体播放器提供音乐频道和评书频道供用户选择。用户可以选择想要进入的频道，并进行多媒体播放器支持的操作，支持的操作有添加播放内容、播放指定序号的内容，以及停止播放。本案例要求编写一个程序，模拟该多媒体播

放器的操作流程，要求使用内部类实现。

**具体要求：**

(1)进入音乐频道，可以选择添加歌曲、播放指定序号的歌曲、停止播放以及退出多媒体四个功能。

(2)添加歌曲需要输入歌曲的名称和时长，播放歌曲时显示例如“正在播放歌曲/团结就是力量/，时长 73 秒”。

(3)进入评书频道，可以选择添加评书、播放指定序号的评书、停止播放以及退出多媒体四个功能。

(4)添加评书需要输入评书的书籍名称和评书人，播放评书时显示例如“正在播放评书《三国演义》，评书人为单田芳”。

**二、题目：用户登录验证**

在大多数网站或应用程序中，都会有用户登录验证程序来验证用户身份，以记录操作的用户信息和防止恶意登录。本案例要求编写一个用户登录验证程序，模拟网站登录时的身份验证过程。

**具体要求：**

(1)首先输入 5 个用户信息（包括用户名和密码），模拟用户的注册。

(2)用户注册完成后进入登录界面，需要输入用户名和密码进行登录验证。

(3)如果用户名不存在于 5 个合法用户中，程序将提示“您不是合法用户！”。

(4)如果用户名存在于 5 合法用户中，但密码不正确，程序将提示“密码错误！”。

(5)如果用户名和密码都正确，程序将提示“登录成功”。

(6)每个用户每天只有三次登录机会，如果连续三次登录失败，程序则自动退出。

(7)要求使用自定义的异常类实现账号不合法和密码错误的登录失败提示。

# Java API

**教学目标：**熟悉 Object 类；掌握 String 类、了解 StringBuffer 类和 StringBuilder 类；掌握 LocalDate 类、LocalTime 类和 LocalDateTime 类；掌握 DateTimeFormatter 类；熟悉 Duration 类和 Period 类；熟悉 System 类；掌握 Math 类和 Random 类；熟悉 Lambda 表达式；掌握正则表达式；

**课型：**复习+新授

**教学重点：**String 类；LocalDate 类、LocalTime 类、LocalDateTime 类和 DateTimeFormatter 类；Math 类；Random 类；

**教学难点：**Lambda 表达式

**教学方法：**概念讲解、案例演示

**教学节数：**理论 6 节、实训 6 节

## 教学过程

### 一 Object 类

- Object 类常用的几种方法：(1)equals(Object obj) (2)hashCode() (3) getClass() (4)toString()
- equals()方法和 toString()方法的重写和使用。

### 二 String 类、StringBuffer 类和 StringBuilder 类

- String 类的常用方法：字符串的获取、判断、转换、替换、去空格、拼接、截取和分割操作。
- StringBuffer 类和 StringBuilder 类与 String 类相比的优势和特征。
- StringBuffer 类、StringBuilder 类、String 类使用上的区别。

### 三 包装类

- 装箱与拆箱。
- 包装类中用于各种数据类型之间转换的方法：(1)Integer valueOf(int i) (2) intValue() (3)Integer valueOf(String s) (4)int parseInt(String s)
- 基本数据类型与 String 类型之间的转换。

## 四 LocalDate 类、LocalTime 类、LocalDateTime 类和

### DateTimeFormatter 类

1. LocalDate 类和 LocalTime 类的常用方法。
2. LocalDateTime 类是 LocalDate 类和 LocalTime 类的综合，还额外提供了日期时间的转换方法。
3. DateTimeFormatter 类的作用和特征。获取 DateTimeFormatter 对象：(1)使用预定义的静态常量创建 DateTimeFormatter 格式器。(2)使用本地化样式创建 DateTimeFormatter 格式器。(3)根据模式字符串创建 DateTimeFormatter 格式器。使用 DateTimeFormatter 将日期、时间格式化为字符串：(1)调用 DateTimeFormatter 的 format()方法进行格式化。(2)调用 LocalDate、LocalDateTime 等日期、时间对象的 format()方法进行格式化。

## 五 Duration 类和 Period 类

1. Duration 类和 Period 类的常用方法和使用。

## 六 System 类、Math 类、Random 类

1. System 类的常用方法：(1)static void exit(int status) (2)static void currentTimeMillis() (3)static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
2. Math 类 Math 类的常用方法。
3. Random 类的作用。创建 Random 类对象：(1)Random() (2)Random(long seed), 使用 Random 类生成随机数。

## 七 Lambda 表达式、Pattern 类和 Matcher 类

1. Lambda 表达式的组成结构和语法格式。
2. 正则表达式元素的概念：(1)字符和特殊字符 (2)量词 (3)边界
3. Pattern 类和 Matcher 类常用方法的使用。

### 实验作业

#### 一、题目：食材入库日志

为了更好地掌握餐厅每天的经营情况，餐厅负责人每天下班前都会盘点当天食材的入库和消耗情况，进而计算当天的营收和次日的采购计划。对此餐厅负责

人要求餐厅食材采购人员采购结束后记录采购明细。本任务要求编写一个程序，对食材入库进行日志记录。

**具体要求：**

- (1) 要求食材入库的信息包含食材的名称和数量，格式为“青菜-20”。
- (2) 停止入库后，控制台输出入库详情，包括每种食材的名称和数量，格式为“青菜-20”。
- (3) 输出最后入库的总数量（所有食材的总数量）。

**二、题目：日程安排管理**

随着生活节奏的加快，人们每天都面临的事务比较多样化，有时候可能会忘记一些重要的日程安排，例如会议，采访等。为了更好地管理和组织日常安排，本案例要求编写一个日程安排管理程序。

**具体要求：**

- (1) 用户可以多次创建日程，但是最多只能创建 20 个，创建日程时包括日程的标题、日期和时间。要求日程的时间不能在当前时间之前。
- (2) 创建完成后，程序按照日程时间由近及远，展示用户的所有日程信息，包括标题、日期、时间、目前距离这个日程还有多久（格式：x 天 x 小时 x 分钟），以及日程的创建时间。

**三、题目：小明的算术题卡**

某小学为了加强学生的算术能力，要求每位学生每天做一篇算术题卡，其中包括 50 道加减乘除的算术题。本案例要求为小明随机出 50 道 100 以内的加减法算式和 20 以内的乘除法算式，作为今天的算术题卡。

**具体要求：**

- (1) 100 以内指 1~100，包括 1 和 100。20 以内指 1~20，包括 1 和 20。
- (2) 加法算式只要求两个加数小于 100，结果可以大于 100。减法算式要求结果为正数。
- (3) 乘法算式只要求两个乘数小于 20，结果可以大于 20。除法算式的结果可以为分数。

**三、题目：模拟用户注册**

许多网站和 APP 在新用户注册时，都会对用户输入信息内容进行限制，例如用户名必须为 6 个字符以内，密码必须包含大小写字母和数字等。本案例要求编写一个程序，模拟一个系统的用户注册过程，注册时需要输入用户名、密码、

邮箱、手机号和身份证，只有每项信息格式都填写正确才会注册成功。

**具体要求：**

- (1) 用户名：必须为 6 个以内的字符，只能包含字母、数字和下划线。
- (2) 密码：必须为 12 个以内的字符，必须包含大小写字母、数字和特殊字符（特殊字符包括! @# \$ % ^ &七种）。
- (3) 邮箱：为一般邮箱格式，通常由用户名和域名组成，中间用"@ "符号连接。用户名由字母、数字、点（“.”）、下划线（“\_”）和连字符（“-”）任意组合而成。域名由字母、数字、连字符（“-”）和点号（“.”）组成。
- (4) 手机号：为一般手机号格式，通常由 11 位数字组成，第 1 位是 1，第 2 位是 3~9 的任意数字，后面 11 位为 0~9 的任意数字。
- (5) 身份证号：为一般身份证号格式，通常由 18 位数字组成，第 1 位为 1~9 的任意数字，中间 16 位为 0~9 的任意数字，最后 1 位为 0~9 的任意数字或 X。需要注意，这里仅是模拟用户注册，旨在巩固读者对正则表达式的使用，为了降低难度，因此没有对身份证内的出生日期进行校验。

要求用户每输入一项信息，程序会对该信息的格式进行检测，如果正确则继续输入下一项，否则提示该项信息格式错误，并重新输入，直到输入正确