

# 一、《编译原理》预备实验

---

## 个人所得税计算器 *PersonalTaxCalc*<sup>1</sup>

### 1. 实验目标

本实验面向 Java 语言的初学者，假设学生已掌握结构化程序设计方法，对面向对象程序设计方法有初步了解。学生最好至少已掌握一门 Java 之外的面向对象程序设计语言，例如 C++ 或 Smalltalk-80。

学生通过本实验可掌握 Java 程序的开发过程，以及面向对象编程技术中封装、信息隐藏、数据抽象、异常处理等机制在实际应用中的有机运用，并养成遵循良好程序设计编码规范的风格。

学生从本实验还可体会到传统的结构化程序设计方法与当前主流的面向对象程序设计方法在建模与设计上的主要区别，为后续的编译原理实验打下初步基础。

### 2. 实验环境

**编程语言：**Java JDK 1.5 或以上版本

**开发工具：**可自由选择 Eclipse、JBuilder 等 IDE 环境，也可直接采用 UltraEdit、EditPlus 等编辑器在命令行工作。但提交的实验结果必须独立于特定的 IDE，可直接运行在 JDK 上。

**编码规范：**要采用面向对象风格来设计实验中的所有类，并遵循一种良好的程序设计习惯。例如，如果你将一个程序的很多功能全部放在一个长长的 `main()` 方法中实现，这样的设计与编码风格会被扣分。

在实验过程中应注意培养规范的编码风格。本实验要求所有源代码严格遵循 Sun 公司（现为 Oracle 公司）关于 Java 程序设计语言的编码规范（Code Conventions

---

1. Credit: Prof. Wenjun Li @ SYSU. 若对实验内容有任何疑问或改进意见，请联系任课教师。

for the Java Programming Language, Revised April 1999)，详细内容请参见：[Code Conventions for the Java Programming Language: Contents \(oracle.com\)](http://codeconventions.oracle.com)。

完成项目代码后，应使用 JDK 附带的文档工具 javadoc，根据源程序中的文档化注释，自动生成相应的说明性文档。

## 3. 实验内容

### 3.1 基本要求

设计一个基于命令行界面的计算个人所得税（工资、薪金所得）应用程序，根据用户输入的当月工资薪金总额计算应缴纳的个人所得税额。为适应未来个人所得税法的可能变化（历史上曾有不同城市执行不同标准的情况，也曾发生过立法修改个人所得税起征基数），该程序应支持对个人所得税起征点进行调整的功能，并且个人所得税各级税率也允许修改。请使用简单的命令行菜单方式让用户选择所需的功能。

### 3.2 个人所得税计算方法

我国税法规定，公民每月工资、薪金收入总额超过 5,000 元者应缴纳个人所得税，即**个人所得税起征点**，2006 年修订立法改为 1,600 元，2011 年调至 3,500 元。收入总额减去免征基数后剩余部分称为**应纳税所得额**，应纳税所得额按下表规定的超额累进税率计算应缴纳的个人所得税。实际上，个人所得税税率表共分 7 级，最高税率为 45%：

级数	应纳税所得额	税率 (%)
1	0-3000 元	3
2	3000-12000 元	10
3	12000-25000 元	20
4	25000-35000 元	25
5	35000-55000 元	30
6	55000-80000 元	35
7	超过 80000 元部分	45

### 3.3 测试用例

例如，张三某月工资薪金总收入为 50,000 元，减去免征基数 5,000 元，则应纳税所得额为 45,000 元。应纳税所得额中，超过 35000 部分为  $45000 - 35000 = 10000$  元，按 30% 税率计算应纳税 3000 元；而 25000-35000 元的部分为 10000 元，按 25% 税率计算应纳税 2500 元；12000-25000 元部分为 13000 元，按 20% 税率计算应纳税 2600 元；3000-12000 元部分为 9000 元，按 10% 税率计算应纳税 900 元；0-3000 元部分为 3000 元，按 3% 税率计算应纳税 90 元。所以，该月张三应缴纳的个人所得税金额为  $3000 + 2500 + 2600 + 900 + 90 = 9090$  元。

## 4. 结果提交

注意提交的实验结果不应只包含源程序代码，还应包括一个面向对象程序的设计文档（建议命名为 `design.doc` 或 `design.pdf`）、编译与运行程序的脚本（`.bat` 文件）、各种测试用例（尽量支持回归测试，Regression Testing）等。

此外，至少还要包括一个纯文本的自述文件 `readme.txt`，其中描述姓名、学号、等基本信息，与提交代码文件的简要解释。

实验结果全部存放在一个名为“学号+姓名”（中间不要任何空格）的文件夹中，压缩成单个文件后，在 **2025/3/19 23:59** 前提交至

<https://yunbiz.wps.cn/c/collect/c9EAZfrH6wn>。

## 5. 评分标准

编程风格：

1. 源程序中标识符命名、程序版面、程序注释等影响程序可理解性的要素（需要遵循 Sun 公司建议的 Java 编码规范）。
2. 是否能熟练运用 Java 语言的文档化注释（同样须遵循 Sun 公司的编码规范）
3. 作为一个完整的实验结果，文档和源程序文件的组织是否合理。

面向对象设计：

1. 程序结构的设计是否合理，譬如程序中对类或方法的抽象是否自然。
2. 程序接口（包括对象接口和方法接口）的设计是否合理。
3. 程序对未来变化的适应性如何，譬如是否容易修改或不必修改就应对可能发生的变化，是否方便在不同的环境下（例如一个基于 J2EE 的 Web 应用中）

复用程序或程序中的构件。

**其他：**

1. 为尽可能提高程序的正确性，测试用例的选择是否完善，有无其他措施有助于提高程序的正确性（譬如单元测试、回归测试），譬如使用单元测试框架JUnit 作为自己的测试工具。
2. 源程序中对异常的处理是否充分，可执行程序的健壮性如何（在异常情况下工作的能力）。
3. 其他可能影响程序质量的要素，例如用户界面是否友好、是否分别提供了构建与运行程序的脚本文件等。