《Java SE程序设计》

实 验 指 导 手 册

授课教师： 崔毅东

实验指导教师： 崔毅东

教学对象：软件工程专业  
二年级本科生

开课时间： 秋季学期

北京邮电大学计算机学院  
（国家示范性软件学院）

2020年10月

# 实验一：Java程序编制（基础练习）

## 实验目的

学生通过使用Java语言进行基本程序的开发，掌握Java通用IDE，练习类的封装使用、Java基本类库的使用、Java GUI编程和事件驱动编程、利用UML进行简单建模。

## 实验内容（详见【二、实验内容说明】）

## 实验环境

1. Eclipse 或者 NetBean
2. 你**应该使用git管理你的代码**。建议你使用代码托管网站托管你的代码。比如阿里云一站式devops (<http://devops.aliyun.com>) 、码云等。
3. 你在托管网站上的代码仓库应该设置为【**私有**】，不然会导致别人抄袭你的代码，从而使得你的实验成绩为0

## 实验要求

1. 独立完成实验内容要求
2. 熟练使用Java常用 IDE 进行编程
3. 上交源程序文件（纸质版或者电子版，以课程指导教师要求为准）
4. 上交实验报告（纸质版或者电子版，以课程指导教师要求为准。标准格式见附件二）

## 实验步骤

1. 启动Java常用 IDE。
2. 建立project。
3. 编辑源程序。
4. 编译、链接并执行源程序，看结果是否正确。
5. 如果报错或告警，做必要修改，重复3－5步骤直到没有错误和告警。

# 实验内容说明

本次实验一共3个Project。

## Tax Table

### Description

2018年8月31日发布的《中华人民共和国个人所得税法》规定：居民个人的综合所得，以每一纳税年度的收入额减除费用六万元以及专项扣除、专项附加扣除和依法确定的其他扣除后的余额，为应纳税所得额。

在中国境内有住所，或者无住所而一个纳税年度内在中国境内居住累计满一百八十三天的个人，为**居民个人**。居民个人从**中国境内和境外**取得的所得，依照本法规定缴纳个人所得税。

在中国境内无住所又不居住，或者无住所而一个纳税年度内在中国境内居住累计不满一百八十三天的个人，为非居民个人。**非居民个人**从**中国境内**取得的所得，依照本法规定缴纳个人所得税。

纳税年度，自公历一月一日起至十二月三十一日止。

以下四项所得称为“综合所得”：工资、薪金所得；劳务报酬所得；稿酬所得；特许权使用费所得。

综合所得，适用百分之三至百分之四十五的**超额累进税率**，参见下面的税率表。



### Design & Output

Develop a program that prints a tax table for annual taxable income from RMB 60,000 to RMB 1,000,000 with intervals of RMB 10,000 for annual and monthly tax, as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Annual Taxable  Income | Annual Tax | Monthly Taxable  Income | Monthly Tax |
| 60000 | 0 | 5000 | 0 |
| 70000 | 300 | 58333 | 25 |
| … | … | … | … |
| 990000 | 428090 | 82500 | 35674 |
| 1000000 | 432590 | 83333 | 36049 |

### Submits

Submit the following items:

1. Analysis: Restate the problem in your own words (e.g., what is input, output if any, what needs to be computed, what data needs to be stored and their type).
2. Design: Clearly describe the steps to solve this problem using English or pseudo code.
3. Coding: Implement the solution in Java.
4. Testing: Submit the screen shots for first 20 lines in the tax table.

## Design Classes with Inheritance and Exceptions

### Design a class named Triangle that extends GeometricObject.

The class contains:

1. Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
2. A no-arg constructor that creates a default triangle.
3. A constructor that creates a triangle with the specified side1, side2, and side3.
4. The getter methods for all three data fields.
5. A method named getArea() that returns the area of this triangle.
6. A method named getPerimeter() that returns the perimeter of this triangle.
7. A method named toString() that returns a string description for the triangle.

For the formula to compute the area of a triangle, see Exercise 5.19. The toString() method is implemented as follows:

return "Triangle: side1 = " + side1 + " side2 = " + side2 + " side3 = " + side3;

### Design an exceptional class named IllegalSideException

The class extends Exception class and should meet the following criteria:

1. Contains data fields side1, side2, side3 to denote three sides of a triangle.
2. A no-arg constructor that stores the information "Illegal Sides"
3. A constructor that stores custom information passed by the parameter
4. Override getMessage() function to return string like "Illegal Sides: side1, side2, side3", where "sideX" is the value of the side of the triangle.
5. When creating a Triangle object with its constructor, check the sides. If any side is no greater than 0, throw IllegalSideException

### Draw UML diagram

Draw the UML diagram that involving the classes Triangle, GeometricObject and IllegalSideException

### Implement and Test

Implement the classes. Write a test program that creates two Triangle objects. One is with sides 1, 1.5, 1, setting color yellow and filled true, and displaying the sides (with toString()), area, perimeter, color, and whether filled or not. Another is with sides -1, 0, 1 which will trigger the IllegalSideException exception and print out the exceptional message.

### Submissions

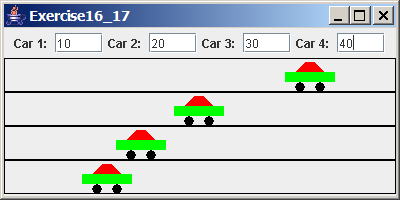
What to submit?

The source code and the screen shot of a sample run.

## GUI, Graphics, and Event-Driven Programming

### Description

Develop a GUI application that simulates four cars racing, as shown in the following figure. You can set the speed for each car (*1 represents the highest speed, 100 means the lowest speed*)



### Design:

Draw a sketch of the user interface that shows the components, containers, and the layout managers.

Design a class named **Car** for illustrating one racing car with appropriate data fields, constructors, and methods.

Draw a **UML diagram** that involves the main frame class, its superclass, and the Car class.

### Implementation

Implement the Car class.

Implement the main application class.

### Submissions

What to submit?

The source code and the screen shot of a sample run.

# 附件二：

实验报告模版，参见Word文档《实验报告模版》