

# 深圳大学实验报告

课程名称 软件工程

项目名称 模块测试

学 院 计算机与软件学院

专 业 计算机科学与技术

指导教师 李俊杰

报 告 人 吴嘉楷 学号 2022150168

实验时间 2024 年 11 月 07 日

提交时间 2024 年 11 月 12 日

教务处制

## 一、实验目的与要求

- (1) 理解单元测试的原理及工作内容；
- (2) 掌握使用流图设计测试用例；
- (3) 掌握测试用例设计方法及应用。

## 二、实验内容与方法

- (1) 程序控制流图，流图复杂度分析；
- (2) 白盒测试用例设计；
- (3) 黑盒测试用例设计；
- (4) 软件测试驱动程序设计。

## 三、实验步骤与过程

### 1. 分析 IncomeTaxCal.java 的程序代码

- (1) **Person** 类：封装了纳税人相关信息

**Person** 类定义了几个属性，包括子女个数、是否继续教育、住房贷款和租房的情况、赡养老人状态，以及兄弟姐妹的数量。通过构造器初始化这些属性，并提供相应的 **getter** 方法访问。

- (2) **incomeTaxCalculator** 类（核心）：计算税收

**incomeTaxCalculator** 类的核心方法是 **cal** 方法，它接受一个 **Person** 对象和工资作为输入参数，输出计算后的所得税金额。

首先，进行一些基本的安全性检查，**Person** 对象为空时会抛出异常，而工资为负数时则抛出非法参数异常。应纳税所得额通过从工资中减去 5000 元的免征额开始计算。然后依次根据不同的扣除项减少所得额，依次包括子女教育扣除（每个子女每月 1000 元）、继续教育扣除（未获得证书时每月 400 元，获得证书当年为 3600 元）、住房贷款扣除（每月 1000 元）、租房费用（根据城市人口规模不同，分别扣除 1500 元、1100 元或 800 元），以及赡养老人扣除（2000 元按兄弟姐妹数量分摊）。

最后，计算个人所得税。通过 **if-else** 语句，根据不同的应纳税所得额区间，税率从 3% 逐渐递增至最高的 45%，将不同区间的计算逻辑逐步累加。

- (3) **test** 类：测试类

主函数中设计了一个具体的测试用例，模拟一个有两个孩子、正在接受继续教育但尚未取得证书、有住房贷款但未租房、并赡养老人且有一个兄弟姐妹的纳税人，月薪为 50000 元。通过调用静态方法 **cal** 方法，计算并输出最终的税额。

### 2. 绘制程序对应的程序流程图

流程图逻辑：

首先，程序开始并创建一个代表个人的 **Person** 对象。接着，程序检查这个对象是否为 **Null** 或者其 **salary**（工资）是否小于 0，如果是，则抛出异常错误并结束程序。如果对象有效，程序会继续初始化应纳税所得额，即月收入减去免征额。然后，依据应纳税额计算公式：**应纳税所得额=月度收入-5000 元（免征额）-专项扣除（三险一金等）-专项附加扣除-依法确定的其他扣除**，程序会从应纳税所得额中减去专项附加扣除、专项扣除和其他扣除，得到最终的应纳税所得额。接下来，程序使用应纳税所得额乘以税率来计算个人所得税，并返回

取整后的个人所得税值，最后结束程序。

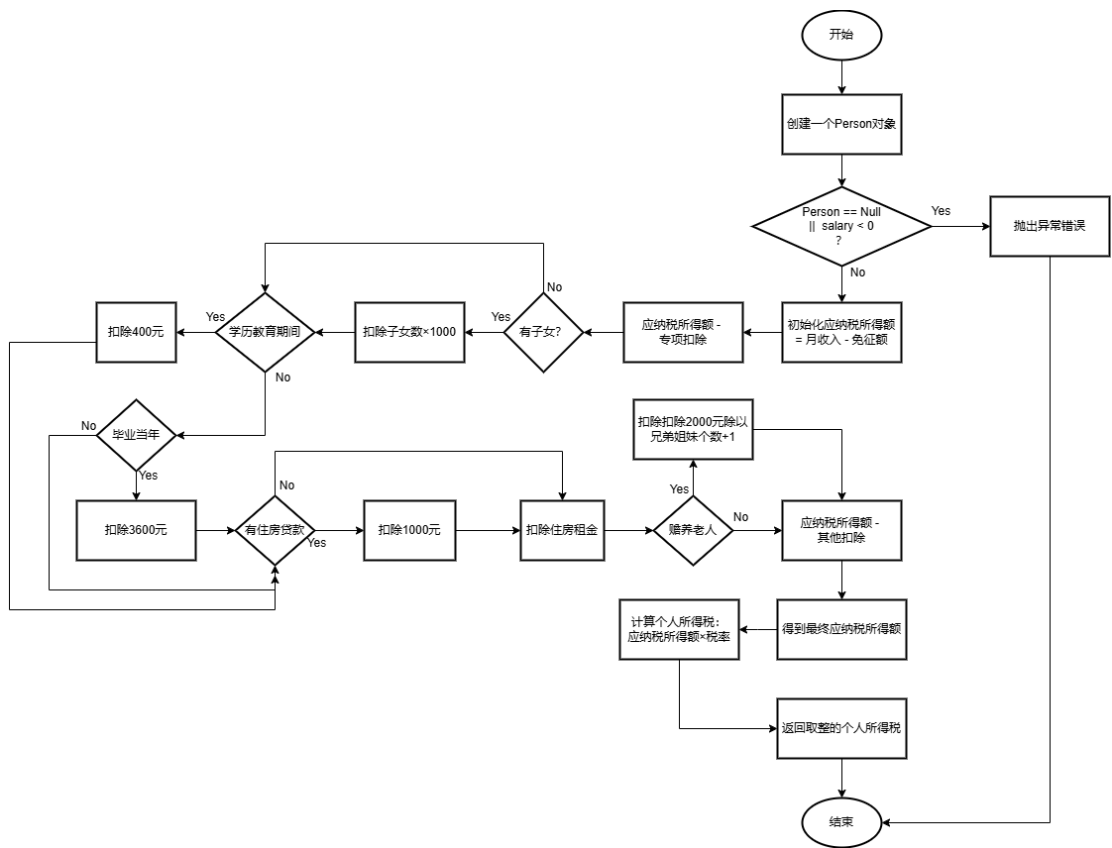


图 1 程序流程图

3. 将程序流程图转化为程序控制流图

(1) 先将具体操作替换为编号

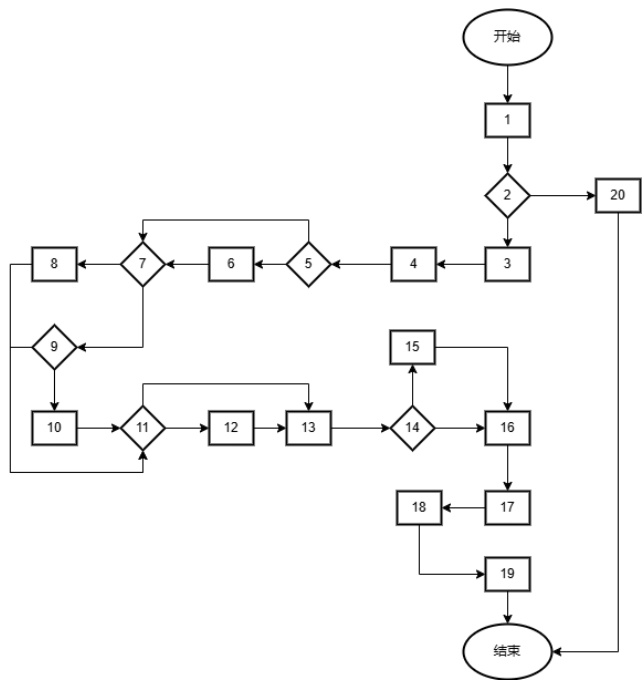


图 2 转化为程序控制流图的中间图

(2) 转化为控制流图

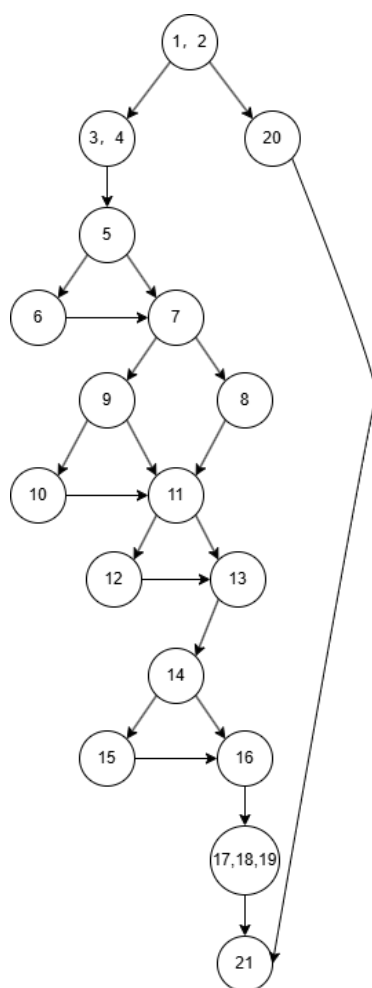


图 3 程序控制流图

4. 分析流图复杂度，确定独立路径的数量

使用三种方法分析流图复杂度  $V(G)$ ：

(1)  $V(G)$  = 流图中的区域数

因为程序控制流图区域数为 7，所以  $V(G) = 7$ 。

(2)  $V(G)$  = 图中边的数量 - 流图中的节点数 + 2

边数  $E = 22$ ，节点数  $N = 17$ ，所以  $V(G) = E - N + 2 = 7$ 。

(3)  $V(G)$  = 流图中判定节点的数量 + 1

判定节点数  $P = 6$ ，所以  $V(G) = P + 1 = 7$ 。

独立路径：

路径一：1,2 -- 20 -- 21

路径二：1,2 -- 3,4 -- 5 -- 7 -- 8 -- 11 -- 13 -- 14 -- 16 -- 17,18,19 -- 21

路径三：1,2 -- 3,4 -- 5 -- 7 -- 8 -- 11 -- 13 -- 14 -- 15 -- 16 -- 17,18,19 -- 21

路径四：1,2 -- 3,4 -- 5 -- 7 -- 8 -- 11 -- 12 -- 13 -- 14 -- 16 -- 17,18,19 -- 21

路径五：1,2 -- 3,4 -- 5 -- 7 -- 9 -- 11 -- 13 -- 14 -- 16 -- 17,18,19 -- 21

路径六：1,2 -- 3,4 -- 5 -- 7 -- 9 -- 10 -- 11 -- 13 -- 14 -- 16 -- 17,18,19 -- 21

路径七：1,2 -- 3,4 -- 5 -- 6 -- 7 -- 8 -- 11 -- 13 -- 14 -- 16 -- 17,18,19 -- 21

5. 针对独立路径设计该函数的白盒测试用例，并给出测试结果

编号	测试用例名称	输入数据							预期结果	测试结果
		孩子数	is_KeepStu	HouseLoans	is_RentHouse	SupportingElderly	sibling_Num	月薪		
1	路径一	2	1	false	0	false	0	-100	error	error
2	路径二	0	1	false	1	false	0	10000	100	100
3	路径三	0	1	false	2	true	0	10000	45	45
4	路径四	0	1	true	2	false	0	20000	2390	2390
5	路径五	0	0	false	3	false	0	15000	710	710
6	路径六	0	2	false	1	false	0	15000	280	280
7	路径七	2	1	false	2	false	0	30000	4190	4190

6. 结合软件需求为该程序代码设计黑盒测试用例

编号	测试用例名称	输入数据							预期结果	测试结果
		孩子数	is_KeepStu	HouseLoans	is_RentHouse	SupportingElderly	sibling_Num	月薪		
1	收入=免征额	0	0	false	0	false	0	5000	0	0
2	收入<免征额	0	0	false	0	false	0	4000	0	0
3	收入>免征额	0	0	false	0	false	0	6000	30	30
4	有一个孩子	1	0	false	0	false	0	7000	30	30
5	有多个孩子	2	0	false	0	false	0	10000	90	90
6	继续教育但未得证书	0	1	false	0	false	0	8000	78	78
7	继续教育并拿到证书	0	2	false	0	false	0	15000	430	430
8	有住房贷款	0	0	true	0	false	0	10000	190	190
9	无贷款且租房在直辖市	0	0	false	1	false	0	15000	640	640
10	赡养老人，有1个兄弟姐妹	0	0	false	0	true	1	20000	2690	2690
11	多个扣除项	2	1	true	2	true	2	50000	8840	8840

## 四、实验结论或体会

实验总结：

在这次软件工程实验中，我深入理解了单元测试的原理及其重要性。通过实践，我掌握了如何使用流图来设计测试用例，并应用了白盒测试和黑盒测试的方法来设计测试用例。这个过程不仅加深了我对理论知识的理解，而且锻炼了我的实际操作能力。

在实验过程中，我首先分析了 `IncomeTaxCal.java` 的代码，这是一个计算所得税的程序。程序创建一个 `Person` 类来封装纳税人的信息，并通过 `incomeTaxCalculator` 类中的 `cal` 方法来计算所得税。

接着，我绘制了程序的流程图，并将其转化为程序控制流图。这个步骤让我对程序的执行逻辑有了更清晰的认识，并且通过分析流图的复杂度，我确定了独立路径的数量等于 7。这一步是至关重要的，因为它帮助我识别了可能存在的测试路径，并为设计测试用例提供了依据。

设计测试用例是我这次实验中最具挑战性的部分。我需要根据独立路径来设计白盒测试用例，并确保覆盖所有的逻辑分支。这个过程让我意识到，测试用例的设计需要非常细致和周到，以确保程序的所有功能都能被充分测试。同时，我也学习了如何根据黑盒测试用例来设计测试，这主要是基于输入和输出的预期来进行的。

实验难点：

1. **程序流程图转换为控制流图。**
2. **分析流图复杂度。**这里一般有三种方法。
  - (1)  $V(G) = \text{流图中的区域数}$
  - (2)  $V(G) = \text{图中边的数量} - \text{流图中的节点数} + 2$
  - (3)  $V(G) = \text{流图中判定节点的数量} + 1$
3. **确定独立路径。**
4. **设计白盒测试用例。**白盒测试是基于程序内部逻辑路径的测试方法，需要对程序源码有一定的理解。
5. **设计黑盒测试用例。**黑盒测试是基于程序功能和输出的测试方法，我们只需要根据功能需求设计即可，可使用等价类划分、边界值分析等方法。
6. **结构化测试。**步骤如下：
  - (1) 将模块的内部处理流程转换为程序控制流图
  - (2) 分析程序控制流图的环形复杂度，确定程序控制流图中独立路径的数量
  - (3) 根据程序控制流图和程序控制流图的环形复杂度确定程序基本路径集合中的独立路径
  - (4) 为每一条独立路径设计一组测试用例，确保每条独立路径至少被执行一次

（实验报告的篇幅控制在 6-8 页）

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。