深 圳 大 学 实 验 报 告

课程名称：软件工程

实验项目名称：实验二：模块测试

学院：计算机与软件学院

专业：软件工程

指导教师：杜文峰

报告人：__学号：_班级：_____

实验时间：2025 年 11 月 27 日

教务部制

实验目的与要求：

实验目的：

（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。

实验内容：

（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。

实验要求：

（1）分析给定的程序代码，绘制该程序对应的流图，并分析流图复杂度；

（2）结合得到的流图，采用 3 种方法分析流图的独立路径；

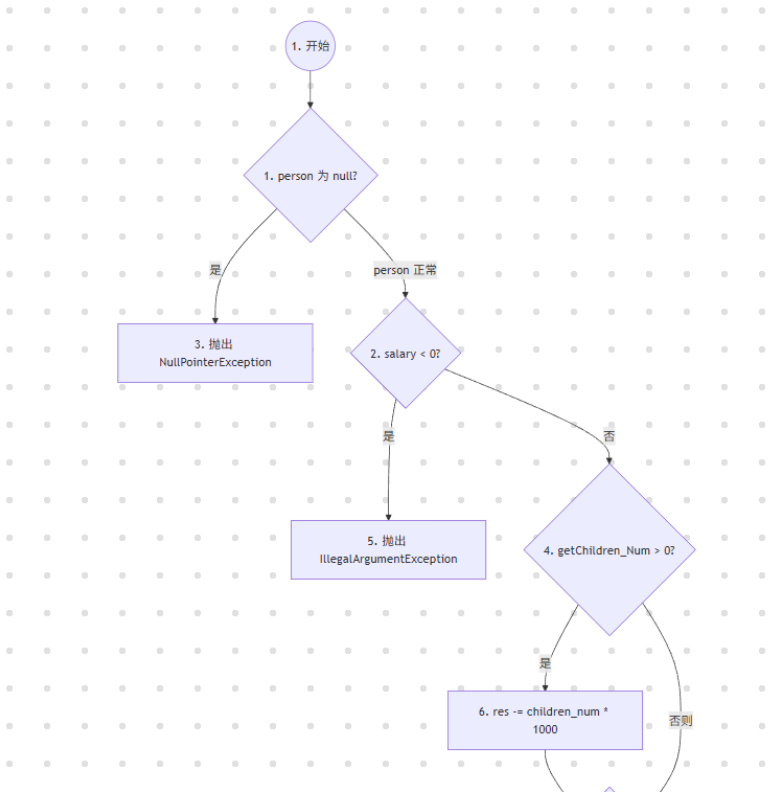
（3）针对独立路径设计该函数的白盒测试用例，并给出测试结果；

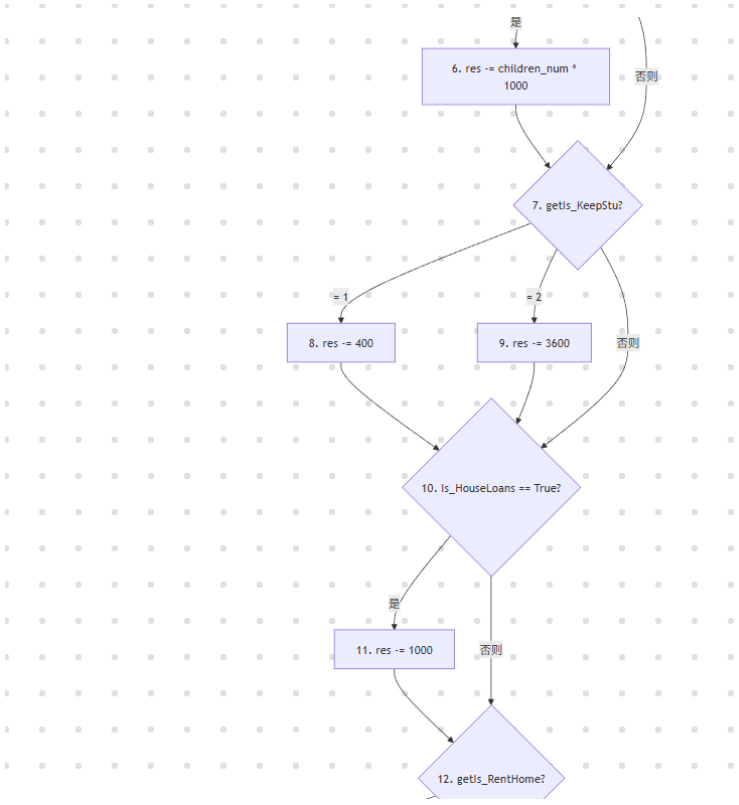
（4）结合软件需求为该程序代码设计黑盒测试用例，并开展测试。

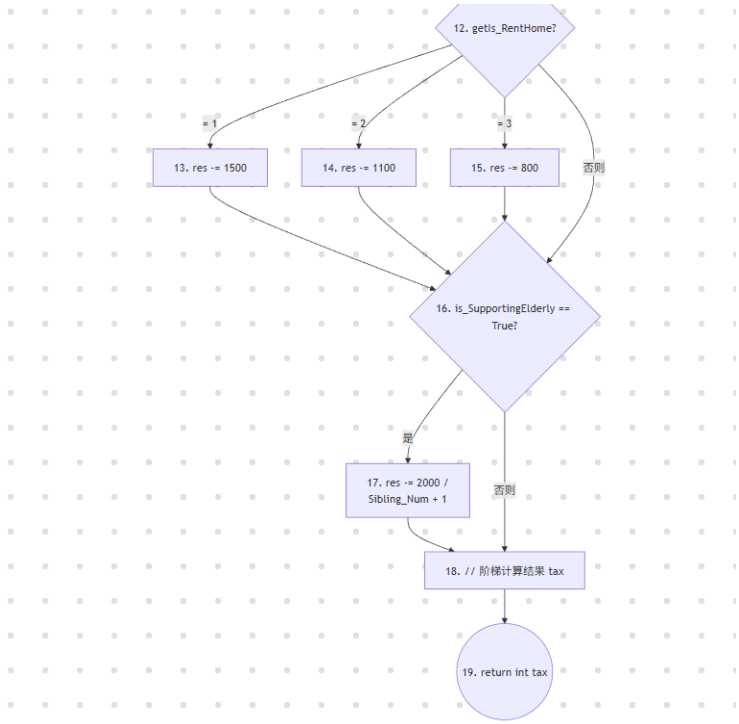
实验过程及内容：

（1）分析给定的程序代码，绘制该程序对应的流图，并分析流图复杂度；

绘制 IncomeTaxCal.java 的控制流图。这个图显示了处理个人所得税计算的逻辑流程，包括各种条件判断和相关操作的决策节点，如下图所示：







分析流图复杂度：
边数（E）=26，节点数（N）=19，连通分支数（P）=1。
使用 V(G)=E−N+2P 来估算流图的复杂度的话，得到 V(G)=9

（2）结合得到的流图，采用 3 种方法分析流图的独立路径；
节点覆盖：我们会确保测试用例覆盖所有的决策点，如是否有子女、是否继续教育、是

否有房贷等。
边覆盖：除了覆盖所有节点，还需要确保测试用例能覆盖从一个决策到另一个决策的转移，例如，不仅测试有
子女的情况，还要测试没有子女的情况。
路径覆盖：这需要更多的测试用例，覆盖例如同时有子女教育扣除和住房贷款扣除的情况，或者有住房贷款但没有子女教育扣除的情况等所有可能的组合。
由于原方法主要都是 if 分支后立马合并，所以上述的三种覆盖方法实际上不需要分开讨论，而可以直接设计完整的属性进行测试。

（3）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
		Person	salary		
1	Test Null Person	null	30000	NullPointerException	与预期相同
2	Test Negative Salary	(0, 0, false, 0, false, 0)	-500	IllegalArgumentException	与预期相同
3	Test No Deductions	(0, 0, false, 0, false, 0)	10000	290	与预期相同
4	Test Children Deduction	(1, 0, false, 0, false, 0)	15000	690	与预期相同
5	Test Education Deduction NoCert	(0, 1, false, 0, false, 0)	12000	450	与预期相同
6	Test Housing Loan Deduction	(0, 0, true, 0, false, 0)	20000	2690	与预期相同
7	Test Supporting Elderly Deduction	(0, 0, false, 0, true, 1)	30000	4690	与预期相同

针对上述表格里的测试用例，写好白盒测试类如下：

```
        // 用例 1: 空的Person项
        try {
            incomeTaxCalculator.cal(null, 30000);
            System.out.println("Test Null Person: Failed - No exception thrown");
        } catch (NullPointerException e) {
            System.out.println("Test Null Person: Passed - Caught NullPointerException");
        } catch (Exception e) {
            System.out.println("Test Null Person: Failed - Wrong exception: " + e);
        }

        // 用例 2: 负薪资输入
        try {
            Person p2 = new Person(0, 0, false, 0, false, 0);
            incomeTaxCalculator.cal(p2, -500);
            System.out.println("Test Negative Salary: Failed - No exception thrown");
        } catch (IllegalArgumentException e) {
            System.out.println("Test Negative Salary: Passed - Caught IllegalArgumentException");
        } catch (Exception e) {
            System.out.println("Test Negative Salary: Failed - Wrong exception: " + e);
        }

        // 用例 3: 无扣除项
        runTestCase("Test No Deductions",
                new Person(0, 0, false, 0, false, 0), 10000, 290);

        // 用例 4: 子女教育扣除
        runTestCase("Test Children Education Deduction",
                new Person(1, 0, false, 0, false, 0), 15000, 690);

        // 用例 5: 继续教育扣除(无证书)
        runTestCase("Test Continuing Education Deduction (No Cert)",
                new Person(0, 1, false, 0, false, 0), 12000, 450);

        // 用例 6: 住房贷款扣除
        runTestCase("Test Housing Loan Deduction",
                new Person(0, 0, true, 0, false, 0), 20000, 2690);

        // 用例 7: 赡养老人扣除
        runTestCase("Test Supporting Elderly Deduction",
                new Person(0, 0, false, 0, true, 1), 30000, 4690);
    }
```

编译并执行，显示所有的测试都为通过，与预期相同。输出如下图：

```
PS E:\Programs\VSC\C++\SE> javac .\testWhiteBox.java
PS E:\Programs\VSC\C++\SE> java testWhiteBox
Test Null Person: Passed - Caught NullPointerException
Test Negative Salary: Passed - Caught IllegalArgumentException
Test No Deductions: Passed - Tax Calculated: 290
Test Children Education Deduction: Passed - Tax Calculated: 690
Test Continuing Education Deduction (No Cert): Passed - Tax Calculated: 450
Test Housing Loan Deduction: Passed - Tax Calculated: 2690
Test Supporting Elderly Deduction: Passed - Tax Calculated: 4690
```

（4）结合软件需求为该程序代码设计黑盒测试用例，并开展测试。

如下表所示，这时不再是“测试用例名称”，而是黑盒测试的“测试用例描述”，符合在不知代码结构的情况下，从逻辑角度做的测试，主要是根据软件的需求和功能来设计。

编号	测试用例名称	输入数据		预期结果	测试结果
		Person	salary		
1	普通收入无任何扣除项	(0, 0, false, 0, false, 0)	5000	0	与预期相同
2	边界条件测试起征点	(0, 0, false, 0, false, 0)	5000	0	与预期相同
3	起征点以上少量收入	(0, 0, false, 0, false, 0)	6000	30	与预期相同
4	多项扣除测试	(2, 1, true, 1, true, 1)	50000	8620	与预期相同
5	高收入测试	(0, 0, false, 0, false, 0)	100000	28890	与预期相同
6	负收入测试	(0, 0, false, 0, false, 0)	-1000	IllegalArgument Exception	与预期相同
7	非法参数测试	null	30000	NullPointerExce ption	与预期相同

针对以上表格里的测试用例，写好黑盒测试类如下：

```
public class testBlackBox {
    public static void main(String[] args) {

        // 用例 1: 普通收入无任何扣除项
        runTestCase("1. 普通收入无任何扣除项",
                new Person(0, 0, false, 0, false, 0), 5000, 0);

        // 用例 2: 边界条件测试起征点
        runTestCase("2. 边界条件测试起征点",
                new Person(0, 0, false, 0, false, 0), 5000, 0);

        // 用例 3: 起征点以上少量收入
        runTestCase("3. 起征点以上少量收入",
                new Person(0, 0, false, 0, false, 0), 6000, 30);

        // 用例 4: 多项扣除测试
        runTestCase("4. 多项扣除测试",
                new Person(2, 1, true, 1, true, 1), 50000, 8620);

        // 用例 5: 高收入测试
        runTestCase("5. 高收入测试",
                new Person(0, 0, false, 0, false, 0), 100000, 28890);

        // 用例 6: 负收入测试
        try {
            Person p = new Person(0, 0, false, 0, false, 0);
            incomeTaxCalculator.cal(p, -1000);
            System.out.println("6. 负收入测试: Failed - No exception thrown");
        } catch (IllegalArgumentException e) {
            System.out.println("6. 负收入测试: Passed - Caught IllegalArgumentException");
        } catch (Exception e) {
            System.out.println("6. 负收入测试: Failed - Wrong exception: " + e);
        }

        // 用例 7: 非法参数测试
        try {
            incomeTaxCalculator.cal(null, 30000);
            System.out.println("7. 非法参数测试: Failed - No exception thrown");
        } catch (NullPointerException e) {
            System.out.println("7. 非法参数测试: Passed - Caught NullPointerException");
        } catch (Exception e) {
            System.out.println("7. 非法参数测试: Failed - Wrong exception: " + e);
        }

    }
```

编译并执行，如图所示，依然通过测试。

```
PS E:\Programs\VSC\C++\SE> javac .\testBlackBox.java
PS E:\Programs\VSC\C++\SE> java testBlackBox
1. 普通收入无任何扣除项: Passed - Tax Calculated: 0
2. 边界条件测试起征点: Passed - Tax Calculated: 0
3. 起征点以上少量收入: Passed - Tax Calculated: 30
4. 多项扣除测试: Passed - Tax Calculated: 8620
5. 高收入测试: Passed - Tax Calculated: 28890
6. 负收入测试: Passed - Caught IllegalArgumentException
7. 非法参数测试: Passed - Caught NullPointerException
```

图文并茂地撰写所有实习过程，并配有步骤的截图。

实验结论：

通过这次实验，我基本掌握了单元测试的重要性和实际应用。通过绘制控制流图并分析复杂度，我加深了对程序结构的理解。实际设计和执行白盒及黑盒测试用例的过程中，我学习了如何从不同角度验证软件功能的正确性和健壮性。

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。