1. 需求分析

所得税计算器是一种工具，用于计算个人所得税。它需要获取输入信息，包括税前工资、免税额、扣除项、其他收入等。它还需要计算应纳税所得额，并使用适当的税率计算个人所得税。计算器应该具有用户友好的界面，并能够正确显示计算结果。

2. 类定义

2.1. taxComputer类 方法

> taxComputer 方法是一个表示所得税计算器的类，它包含以下属性和方法

属性：

- salary: 税前工资
- standard: 征税标准
- taxRate: 适用的税率
- sPoint: 起征点

计算步骤：

- 设置税前工资
- 设置扣除项
- 设置免税额
- 计算应纳税所得额
- 计算个人所得税

2.2. opInterface 方法

> opInterface 方法是一个表示用户界面的类，它包含以下方法

方法：

- 获取税前工资输入
- 获取扣除项输入
- 获取免税额输入
- 获取其他收入输入
- 显示计算结果

2.3 taxHelper 方法

> taxHelper 方法负责处理TaxCalculator 和 opInterface 的依赖关系

3. 类关系和依赖关系

3.1 TaxCalculator 方法依赖于 opInterface 方法，以获取输入信息和显示计算结果

> TaxCalculator 和 opInterface 通过 taxHelper 完成关系的构建和输出

4. 方法实现

4.1. TaxCalculator 方法实现

> 构建UI面板

4.2 opInterface 方法实现

> 根据输入进行遍历计算所得税并返回给taxHelper

4.3 taxHelper 方法实现

> 收集结果并且返回用户可视化界面

5.设计代码

```
/**
 * @author 任铭20337231
 * @Description 个人所得税计算器
 * @Date 下午12:05 13/3/2023
 * @Param
 * @return
 **/
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        int opNum = 0;
        try {
            opNum = opInterface();
        } catch (Exception e) {
            System.out.println("Wrong input!");
            System.exit(1);
        }
        taxHelper(opNum);
        System.out.println("thanks for your using!");
    }
/**
 * @Author 任铭20337231
 * @Description 这个方法作用是设计用户交互的UI面板
 * @Date 下午12:06 13/3/2023
 * @Param []
 * @return 用户选择的税收模板号
 **/
    public static int opInterface() {
        System.out.println("Please choose a pattern:");
        System.out.println("1. standard 2022.ver");
        System.out.println("2. standard 2019.ver");
```

```
        System.out.println("3. standard 2077.ver");
        System.out.println("4. standard your.ver");
        System.out.println("0. exit");
        Scanner scanner = new Scanner(System.in);
        return scanner.nextInt();
    }
/**
 * @Author 任铭20337231
 * @Description 这个方法作用是计算税以及返回缴税值和剩余工资
 * @Date 下午12:07 13/3/2023
 * @Param [salary, standard, taxRate, sPoint]
 * @return void
 **/
    public static void taxComputer(double salary, double[] standard, double[]
taxRate, double sPoint) {
        System.out.println("You salary is(before tax): " + salary);
        double taxSalary = salary - sPoint; // 求出用户的应纳税所得额
        if (taxSalary <= 0) {
            System.out.println("It seems you don't need to pay any tax. Life
sucks sometime, right? Try go pick up yourself.");
            return;
        }
        for (int i = standard.length - 1; i >= 0; i--) {
            // 如果应纳税所得额在对应区间
            if (taxSalary >= standard[standard.length - 1] || i !=
standard.length - 1 && taxSalary >= standard[i] && taxSalary < standard[i + 1])
{
                int curLevel = i;
                double totalTax = 0;
                totalTax += (taxSalary - standard[curLevel]) *
taxRate[curLevel];
//                System.out.println(totalTax);
                while (--curLevel >= 0) {
                    totalTax += (standard[curLevel + 1] - standard[curLevel]) *
taxRate[curLevel];
                    System.out.println((standard[curLevel + 1] -
standard[curLevel]) * taxRate[curLevel]);
                }
                System.out.println("Your tax is " + totalTax + ".");
                System.out.println("The rest of your salary is " + (salary -
totalTax) + ".");
                System.out.println("Enjoy your life!");
                break;
            }
        }
    }
/**
 * @Author 任铭20337231
 * @Description 这个函数作用是根据接受的opNum进行检错和执行
 * @Date 下午12:07 13/3/2023
 * @Param [opNum]
 * @return void
 **/
    public static void taxHelper(int opNum) {
        while (opNum != 0) {
```

```
            System.out.println("Please input your salary: ");
            Scanner scanner = new Scanner(System.in);
            double salary;
            // 判断是否有猴子在键盘上乱按
            try {
                salary = scanner.nextDouble();
            } catch (Exception e) {
                System.out.println("Please input right format!");
                // System.exit(1); // 如果想检错停止就用这个
                continue; // 输入错误 继续输入
            }
            // 判断是否出现资本家
            if (salary < 0) {
                System.out.println("You can't give a worker negative salary!");
                System.exit(0);
            }
            if (opNum == 1) {
                double[] standard = {0, 3000, 12000, 25000, 35000, 55000,
80000}; // 每级应纳税所得额
                double[] taxRate = {0.03, 0.1, 0.2, 0.25, 0.3, 0.35, 0.45}; // 每
级税率
                double sPoint = 5000;  // 起征点
                taxComputer(salary, standard, taxRate, sPoint);
            }
            if (opNum == 2) {
                double[] standard = {0, 2000, 5000, 20000, 30000, 55000, 75000};
// 每级应纳税所得额
                double[] taxRate = {0.02, 0.1, 0.2, 0.25, 0.3, 0.35, 0.45}; // 每
级税率
                double sPoint = 3500;  // 起征点
                taxComputer(salary, standard, taxRate, sPoint);
            }
            if (opNum == 3) {
                double[] standard = {0, 10000, 50000, 250000, 350000, 550000,
800000}; // 每级应纳税所得额
                double[] taxRate = {0.01, 0.11, 0.12, 0.215, 0.13, 0.135,
0.145}; // 每级税率
                double sPoint = 50000;  // 起征点
                taxComputer(salary, standard, taxRate, sPoint);
            }
            if (opNum == 4) {
                System.out.println("input taxable income level:");
                int sLevel = scanner.nextInt();
                double[] standard = new double[sLevel];
                double[] taxRate = new double[sLevel];
                System.out.println("input taxable income: (example:{[0,200),
[200,300)}, you input 0 200 300)");
                for (int i = 0; i < sLevel; i++) {
                    standard[i] = scanner.nextDouble();
                }
                System.out.println("input tax rate: (example:{0.03,0.1,0.2}, you
input 0.03 0.1 0.2)");
                for (int i = 0; i < sLevel; i++) {
                    taxRate[i] = scanner.nextDouble();
                }
```

```
                System.out.println("input start_tax_point:");
                double sPoint = scanner.nextDouble();
                taxComputer(salary, standard, taxRate, sPoint);
            }
            try {
                opNum = opInterface();
            } catch (Exception e) {
                System.out.println("Wrong input!");
                System.exit(1);
            }
        }
    }
}
```

6. 测试用例

> 运行文件夹下的taxComputer.bat文件

假设在2022年中国税收标准下 税前工资为50000