

# 基于JAVA语言的个人所得税计算器

## 一、已实现的功能

- 1、用户可在命令行与程序进行交互，输入相关指令得到想要的效果。
- 2、用户输入自己的工资后，程序会计算出用户应交的税收为多少。
- 3、用户可根据指令，修改税收表的基本信息，如起征点、税收范围、税率等。
- 4、默认的税收表信息为：

```
欢迎使用个人所得税计算器！
个人所得税计算信息：

起征点为：1600

+-----+-----+-----+-----+
| Income Start | To | Income End | Rate |
+-----+-----+-----+-----+
| 0.00         | 至 | 500.00     | 5.00%|
| 500.00       | 至 | 2000.00    | 10.00%|
| 2000.00      | 至 | 5000.00    | 15.00%|
| 5000.00      | 至 | 20000.00   | 20.00%|
+-----+-----+-----+-----+
```

## 二、文件结构

doc： 为该项目生成的javadoc文件夹，打开文件夹中的 index.html 即可查看可视化后的文件结构。

src： 文件夹 main 中存放了所有项目组件的实现代码，分别为 Main.java, TaxView.java, TaxTable.java, TaxCalculator.java, ChangeTable.java。

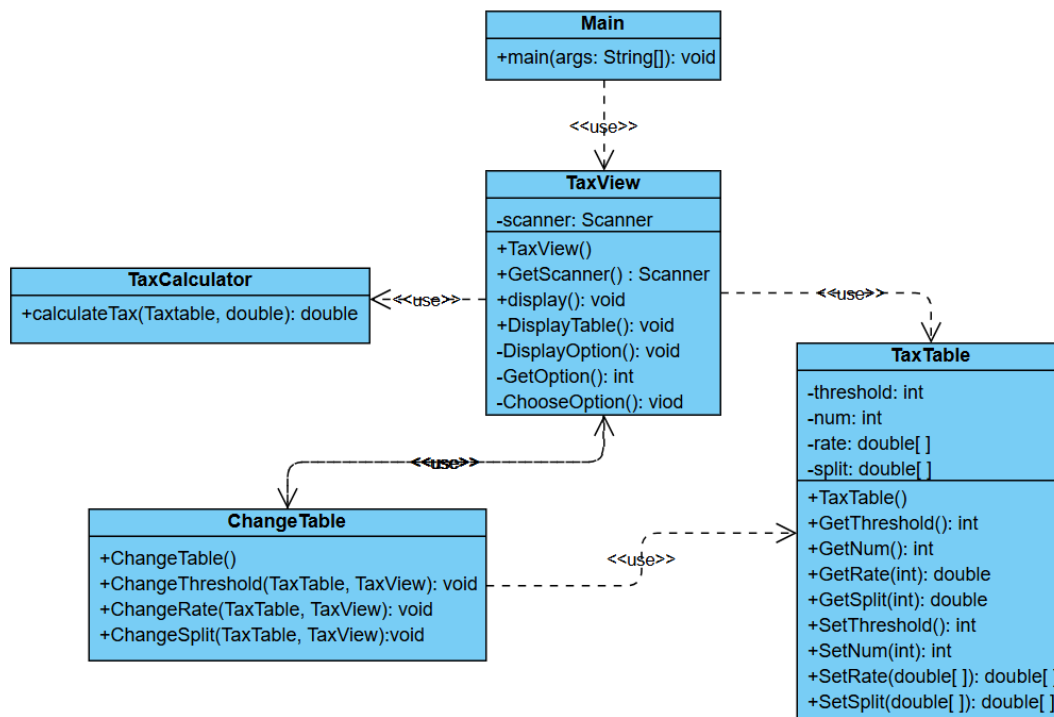
文件夹 test 中存放了测试代码，由 junit 实现对项目代码是否能算出正确答案作出判断。

pom.xml： 是 Maven 用来管理项目构建、依赖和运行配置的核心文件，让Java项目更高效自动化。

run.bat： 这个批处理脚本依次自动执行 Maven 的清理、编译和运行主类操作，用于快速启动 Java 项目。

## 三、设计思路

### UML设计图



## 设计描述

该项目涉及到了五个类：

类 `Main` 是程序的入口。

类 `TaxView` 实现可视化页面以及用户交互。

类 `TaxTable` 用于存储个人所得税计算所需的基本信息，也可以更改税收表的信息。

类 `TaxCalculator` 计算个人应缴纳的税款。

类 `ChangeTable` 提供了用于修改个人所得税计算器参数的方法。

## 四、实现效果

### 运行脚本启动程序

```
PS D:\25fall\编译原理实验\lab1\tax-calculator> .\run.bat
Cleaning project...
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.taxcalculator:tax-calculator >-----
[INFO] Building tax-calculator 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ tax-calculator ---
[INFO] Deleting D:\25fall\编译原理实验\lab1\tax-calculator\target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 0.405 s
[INFO] Finished at: 2025-03-19T17:51:59+08:00
[INFO] -----
Compiling project..
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.taxcalculator:tax-calculator >-----
[INFO] Building tax-calculator 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ tax-calculator ---
[INFO] skip non existing resourceDirectory D:\25fall\编译原理实验\lab1\tax-calculator\src\main\resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ tax-calculator ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 5 source files with javac [debug release 23] to target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 1.447 s
[INFO] Finished at: 2025-03-19T17:52:02+08:00
[INFO] -----
Running project...
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.taxcalculator:tax-calculator >-----
[INFO] Building tax-calculator 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ tax-calculator ---
欢迎使用个人所得税计算器！
个人所得税计算信息：
```

## 欢迎页面

```
[INFO] --- exec:3.1.0:java (default-cli) @ tax-calculator ---
欢迎使用个人所得税计算器！
个人所得税计算信息：

起征点为：1600

+-----+-----+-----+-----+
| Income Start | To | Income End | Rate |
+-----+-----+-----+-----+
| 0.00         | 至 | 500.00     | 5.00% |
| 500.00       | 至 | 2000.00    | 10.00% |
| 2000.00      | 至 | 5000.00    | 15.00% |
| 5000.00      | 至 | 20000.00   | 20.00% |
+-----+-----+-----+-----+

1. 输入您的工资并进行计算
2. 更改起征点
3. 更改税收范围
4. 更改税率
5. 退出程序
请选择模式：
```

# 计算税收

请选择模式：  
1  
请输入您的工资：  
4300  
您需要支付的税收为： 280.00  
-----

# 更改税收表

请选择模式：  
2  
请修改您的起征值：  
1500  
起征值修改成功！  
-----

个人所得税计算信息：  
  
起征点为： 1500

Income Start	To	Income End	Rate	
0.00	至	500.00	5.00%	
500.00	至	2000.00	10.00%	
2000.00	至	5000.00	15.00%	
5000.00	至	20000.00	20.00%	

  
-----

1. 输入您的工资并进行计算  
2. 更改起征点  
3. 更改税收范围  
4. 更改税率  
5. 退出程序  
请选择模式：  
3  
请修改您的税收范围（以空格分隔，首位需为0）：  
输入不能为空，请重新输入：  
0 1000 2000 3000 4000  
税收范围修改成功！  
-----

个人所得税计算信息：  
  
起征点为： 1500

Income Start	To	Income End	Rate	
0.00	至	1000.00	5.00%	
1000.00	至	2000.00	10.00%	
2000.00	至	3000.00	15.00%	
3000.00	至	4000.00	20.00%	

  
-----

请选择模式：

4

请修改您的税率范围（以空格分隔，不能超过1）：

0 0.1 0.2 0.3 0.4

税率修改成功！

个人所得税计算信息：

起征点为：1500

Income Start	To	Income End	Rate
0.00	至	1000.00	0.00%
1000.00	至	2000.00	10.00%
2000.00	至	3000.00	20.00%
3000.00	至	4000.00	30.00%

## 验证计算器可正常运行并且得出的答案正确

使用 `junit` 进行判断，由下图可以看出，我们的计算器得出的答案是正确的。

```
tax-calculator > src > test > java > com > taxcalculator > J TaxCalculatorTest.java > ...
1  package com.taxcalculator;
2
3  import org.junit.Test;
4  import static org.junit.Assert.*;
5
6  public class TaxCalculatorTest {
7
8      @Test
9      public void testTaxCalculator() {
10          TaxTable taxtable = new TaxTable();
11          assertEquals(0, TaxCalculator.calculateTax(taxtable, salary:300), 0.1);
12          assertEquals(0, TaxCalculator.calculateTax(taxtable, salary:1600), 0.1);
13          assertEquals(0, TaxCalculator.calculateTax(taxtable, salary:1599), 0.1);
14          assertEquals(280, TaxCalculator.calculateTax(taxtable, salary:4300), 0.1);
15          assertEquals(310, TaxCalculator.calculateTax(taxtable, salary:4500), 0.1);
16      }
17  }
18
```

问题 2 输出 调试控制台 测试结果 终端 端口

%TESTC 0 v2  
%RUNTIME0