

Exercise 02

班级：2017211106 学号：2017212116 姓名：杨诺诚

问题一：通过汇率转换货币金额

解决思路：

在此问题中，需要用户输入汇率转换的方向以及某一种货币，但是汇率是一个实时变化的量，于是我是用 Java 和 selenium 库函数，通过 Google 浏览器查找人民币和美元兑换的当前汇率，用户可以选择使用自己输入的汇率，也可以使用最新的数据做运算。

selenium 是常用的爬虫库之一，通过页面中的 xpath 定位到汇率数字，并取出放置在变量中。

代码实现：

```
package com.company;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import java.text.DecimalFormat;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.setProperty("webdriver.chrome.driver", "/Users/yangnuocheng/Desktop/JavaExercise/第二次作业/JavaJsoup_Ecode/JavaSelenium/Googledrive/chromedriver");
        //chromedriver 地址，必须先初始化后调用
        WebDriver driver = new ChromeDriver();
        driver.get("https://finance.sina.com.cn/money/forex/hq/USDCNY.shtml");
        //新浪财经，外汇板块，中美汇率页面
        String USDCNY_Xpath="/html/body/div[@class='wrap']/div[@class='part_01 clearfix']/div[@class='p_middle']/div[@id='quoteWrap']/div[@class='price_bar up']/div[@class='price']/h5";
        String USDCNY_Exchange =
        driver.findElement(By.xpath(USDCNY_Xpath)).getText();
        String
        USDCNY_Current=driver.findElement(By.xpath(USDCNY_Xpath)).getText();
        //通过页面信息中的xpath 查找汇率具体值
        System.out.println("Current exchange rate
        is :"+driver.findElement(By.xpath(USDCNY_Xpath)).getText()+".");
        driver.close();
        int quit=0;
```

```

DecimalFormat df=new DecimalFormat("0.00");//设置保留位数
while(quit==0) {
    System.out.print("Enter the exchange rate from dollars to RMB or you
can enter 0.0 to use current exchange rate:");
    double USDCNY_MyExchange = input.nextDouble();
    System.out.print("Enter 0 to convert dollars to RMB and 1 vice versa:");
    double Number = input.nextInt();
    if (Number == 0) System.out.print("Enter the dollar amount:");
    else if (Number == 1) System.out.print("Enter the RMB amount:");
    else {
        System.out.print("Input Wrong,please try again.");
        continue;
    }
    double amountOfMoney = input.nextDouble();
    double aim = 0.0;
    if (USDCNY_MyExchange == 0)
        USDCNY_MyExchange = Double.parseDouble(USDCNY_Current);
    if (Number == 0) {
        //dollars 2 RMB
        aim = amountOfMoney * USDCNY_MyExchange;
    }
    else {
        aim = amountOfMoney / USDCNY_MyExchange;
    }
    String output = df.format(aim);
    if (Number == 0) {
        System.out.println(amountOfMoney+" dollars is ¥"+output);
    }
    else {
        System.out.println(amountOfMoney+"RMB is $" +output);
    }
}
}
}

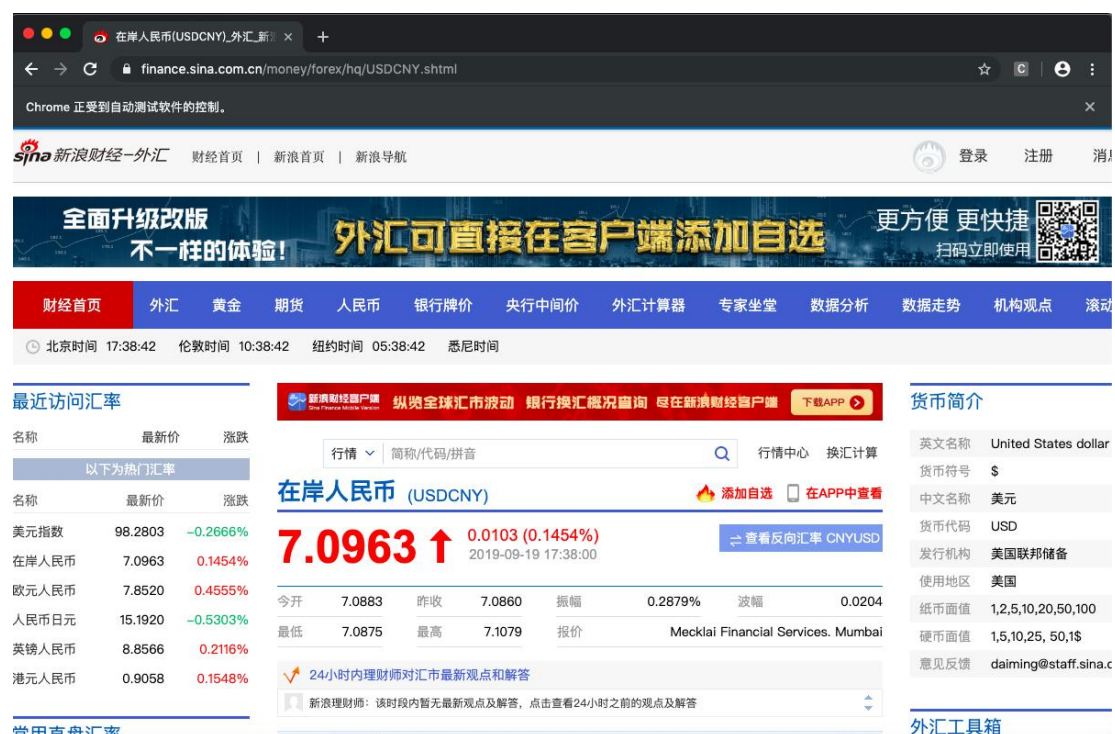
```

效果图展示:

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...
Starting ChromeDriver 76.0.3809.68 (420c9498db8ce8fcd190a954d51297672c1515d5-refs/branch-heads/3809@{#864}) on port 19110
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
[1568885904.334] [WARNING]: This version of ChromeDriver has not been tested with Chrome version 77.
[1568885904.784] [WARNING]: This version of ChromeDriver has not been tested with Chrome version 77.
9月 19, 2019 5:38:25 下午 org.openqa.selenium.remote.ProtocolHandshake createSession
信息: Detected dialect: W3C
Current exchange rate is :7.0963.
Enter the exchange rate from dollars to RMB or you can enter 0.0 to use current exchange rate:7.2
Enter 0 to convert dollars to RMB and 1 vice versa:0
Enter the dollar amount:1
1.0 dollars is ¥7.20
Enter the exchange rate from dollars to RMB or you can enter 0.0 to use current exchange rate:0.0
Enter 0 to convert dollars to RMB and 1 vice versa:0
Enter the dollar amount:1
1.0 dollars is ¥7.10
Enter the exchange rate from dollars to RMB or you can enter 0.0 to use current exchange rate:
```

程序控制台，这部分可以通过输入汇率的位置输入 0.0 选择使用网页中的汇率，否则使用刚刚输入的汇率。

这部分操作会不断循环，一次转换结束会立马刷新重新开始下一次询问。



Java 控制 GoogleChrome 访问外汇网站。

在这个 Java 爬虫的实践过程中，我也学习了 Java 引入外部.jar 文件的方法，可以方便的加入其他工程库，丰富了 Java 语言的功能和构建工程的速度。

问题二：一百以内的加减法 Quiz 设计

解决思路:

为方便学生使用 Quiz，应该使 Quiz 方便查看，方便纠错，方便清除。

故此我是用 swing 库，在第一次作业（计算总投入的 GUI 界面）的基础上，制作

随机的加减法习题代替原本的静态标签，Text 框供用户输入结果，点击“回答完毕”按钮后可以检查全部（五道题目）的正确性，并将结果反馈到下方的文本框中，若有题目没有做对，可以点击“重新计算”按钮，会自动将错误的答案清除，方便重新填写答案，当所有题目都正确时，系统还会提出表扬。

代码实现：

```
package com.company;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class ch02_2_Quiz {
    public static void main(String[] args) {
        // 创建 JFrame 实例
        JFrame frame = new JFrame("加减法测试");
        // Setting the width and height of frame
        frame.setSize(350, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel();
        // 添加面板
        frame.add(panel);

        placeComponents(panel);

        // 设置界面可见
        frame.setVisible(true);
    }

    private static void placeComponents(JPanel panel) {

        panel.setLayout(null);

        //arrayRefVar = new dataType[arraySize];
        String[] CalNumber = new String[5];
        int[] Answer = new int[5];
        // 创建计算公式的数组
        Random random = new Random();
```

```

for (int i = 0; i < 5; ) {
    int a = Math.abs((random.nextInt()) % 100);
    int b = Math.abs((random.nextInt()) % 100);
    if (i % 2 == 0 && a != b && a >= 10 && b > 10) {
        // 减法
        if (a < b) {
            int bet = a;
            a = b;
            b = bet;
        }
        if (a - b > 0) {
            CalNumber[i] = (String) (a + "-" + b);
            Answer[i] = a - b;
            i++;
        } else if (a + b < 100) {
            CalNumber[i] = (String) (a + "+" + b);
            Answer[i] = a + b;
            i++;
        } else continue;
    } else if (a >= 10 && b >= 10) {
        // 加法
        if (a + b < 100) {
            CalNumber[i] = (String) (a + "+" + b);
            Answer[i] = a + b;
            i++;
        } else continue;
    }
}

for (int i = 0; i < 5; i++)
    System.out.println(CalNumber[i]);

JLabel TotLabel = new JLabel(CalNumber[0]);
TotLabel.setBounds(10, 20, 80, 25);
panel.add(TotLabel);

JLabel PriceLabel = new JLabel(CalNumber[1]);
PriceLabel.setBounds(10, 50, 80, 25);
panel.add(PriceLabel);

JLabel YearLabel = new JLabel(CalNumber[2]);
YearLabel.setBounds(10, 80, 80, 25);
panel.add(YearLabel);

JLabel AnsLabel = new JLabel(CalNumber[3]);

```

```
AnsLabel.setBounds(10, 110, 80, 25);
panel.add(AnsLabel);

JLabel Text4Label = new JLabel(CalNumber[4]);
Text4Label.setBounds(10, 140, 80, 25);
panel.add(Text4Label);

JTextField Text_1 = new JTextField(20);
Text_1.setBounds(100, 20, 165, 25);
panel.add(Text_1);

JTextField Text_2 = new JTextField(20);
Text_2.setBounds(100, 50, 165, 25);
panel.add(Text_2);

JTextField Text_3 = new JTextField(20);
Text_3.setBounds(100, 80, 165, 25);
panel.add(Text_3);

JTextField Text_4 = new JTextField(20);
Text_4.setBounds(100, 110, 165, 25);
panel.add(Text_4);

JTextField Text_5 = new JTextField(20);
Text_5.setBounds(100, 140, 165, 25);
panel.add(Text_5);

JButton loginButton = new JButton("回答完毕");
loginButton.setBounds(100, 200, 80, 25);
panel.add(loginButton);

JButton RestartButton = new JButton("重新计算");
RestartButton.setBounds(200, 200, 80, 25);
panel.add(RestartButton);

RestartButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        double [] Student_Ans = new double[5];
        Student_Ans[0] = Double.parseDouble(Text_1.getText());
        Student_Ans[1] = Double.parseDouble(Text_2.getText());
        Student_Ans[2] = Double.parseDouble(Text_3.getText());
        Student_Ans[3] = Double.parseDouble(Text_4.getText());
        Student_Ans[4] = Double.parseDouble(Text_5.getText());
```

```

        int Wrong = 0;
        int Right = 0;
        int[] Wrong_Number = new int[5];
        for (int i=0;i<5;i++) {
            Wrong_Number[i]=0;
        }
        for(int i=0;i<5;i++)
        {
            if(Student_Ans[i]!=Answer[i])
            {
                switch (i) {
                    case 0:Text_1.setText("");break;
                    case 1:Text_2.setText("");break;
                    case 2:Text_3.setText("");break;
                    case 3:Text_4.setText("");break;
                    case 4:Text_5.setText("");break;
                }
            }
        }
    });
    // 创建文本输出框
    JTextArea AnswerText = new JTextArea();
    AnswerText.setBounds(10,170,255,25);
    panel.add(AnswerText);

    loginButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            double [] Student_Ans = new double[5];
            Student_Ans[0] = Double.parseDouble(Text_1.getText());
            Student_Ans[1] = Double.parseDouble(Text_2.getText());
            Student_Ans[2] = Double.parseDouble(Text_3.getText());
            Student_Ans[3] = Double.parseDouble(Text_4.getText());
            Student_Ans[4] = Double.parseDouble(Text_5.getText());
            int Wrong = 0;
            int Right = 0;
            int[] Wrong_Number = new int[5];
            for (int i=0;i<5;i++) {
                Wrong_Number[i]=0;
            }
            for(int i=0;i<5;i++)
            {
                if(Student_Ans[i]!=Answer[i])

```

```

        {
            Wrong++;
            Wrong_Number[i]=1;
        }
        else Right++;
    }
    AnswerText.setText("你做错了"+Wrong+"道题目");
    if(Wrong!=0) {
        AnswerText.append("他们分别是: ");
        for(int j=0;j<5;j++)
        {
            if(Wrong_Number[j]==1)
                AnswerText.append(j+1+" ");
        }
    }
    else AnswerText.setText("恭喜你, 你全都做对了!");
    System.out.println("Wrong: "+Wrong);
    System.out.println("Right: "+Right);
    //    AnswerText.setText(String.valueOf(Answer));
    }
    });
}

}

```

效果图展示





在这个实例中，我使用 **Random** 函数随机两个一百以内的数字，通过加减组合得到计算公式数组中并将它们的答案保存在另一个答案数组中，实践了字符串数组的使用，巩固了 **swing** 下的文本框元素操作。

问题三：缴纳税金的计算

解决思路：

根据我国的个人所得税法（不考虑租房基金、医疗基金等方面问题），将全年个人所得税分成 7 个部分（见下表），分别计算需要缴纳的税金。

个人所得税税率表一

（综合所得适用）

级数	全年应纳税所得额	税率（%）
1	不超过36000元的	3
2	超过36000元至144000元的部分	10
3	超过144000元至300000元的部分	20
4	超过300000元至420000元的部分	25
5	超过420000元至660000元的部分	30
6	超过660000元至960000元的部分	35
7	超过960000元的部分	45

代码实现：

```
package com.company;

import java.util.Scanner;

public class Income_Tax {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("请输入年收入: ");
        double Tot=input.nextDouble();
        double Tax=0;
        if(Tot>=36000)
        {
            Tot-=36000;
            if(Tot<=108000){
                // 第二段缴税
                Tax+=Tot*0.1;
            }
            else {
                Tax+=108000*0.01;
                Tot-=144000;
                if(Tot<=156000)
                    Tax+=Tot*0.2;
                else{
                    Tax+=156000*0.2;
                    Tot-=156000;
                    if(Tot<=120000)
                    {
                        Tax+=Tot*0.25;
                    }
                    else{
                        Tax+=120000*0.25;
                        Tot-=120000;
                        if(Tot<=240000) {
                            Tax += Tot * 0.3;
                        }
                        else{
                            Tot-=240000;
                            Tax+=240000 * 0.3;
                            if(Tot<=300000){
                                Tax+=Tot*0.35;
                            }
                        }
                    }
                    else {
                        Tot-=300000;
                        Tax+=300000*0.35;
                    }
                }
            }
        }
    }
}
```

```

        Tax+=Tot*0.45;
    }
}
}
}
}
}
else Tax=Tot*0.03;
System.out.print("需要缴纳"+Tax+"的税金.");
}
}
}

```

效果图展示:

```

/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...
请输入年收入: 10000
需要缴纳300.0的税金.
Process finished with exit code 0

```

对未计算税金的部分做判断, 当判断值小于达到下一层级的最低金额时, 即可在当前环节计算税金并退出程序, 若达到了下一层级的最低金额, 需要计算本环节的最大税金并重新给出未计算税金的部分并进入下一阶段。

问题四: IntelliJ idea debug 过程分析报告

解决思路:

常用的 Debug 方法有:

- 1.设置断点
- 2.查看栈区变量
- 3.智能步进
- 5.其他方法 (包含变量查找, 包管理等)

操作实现:

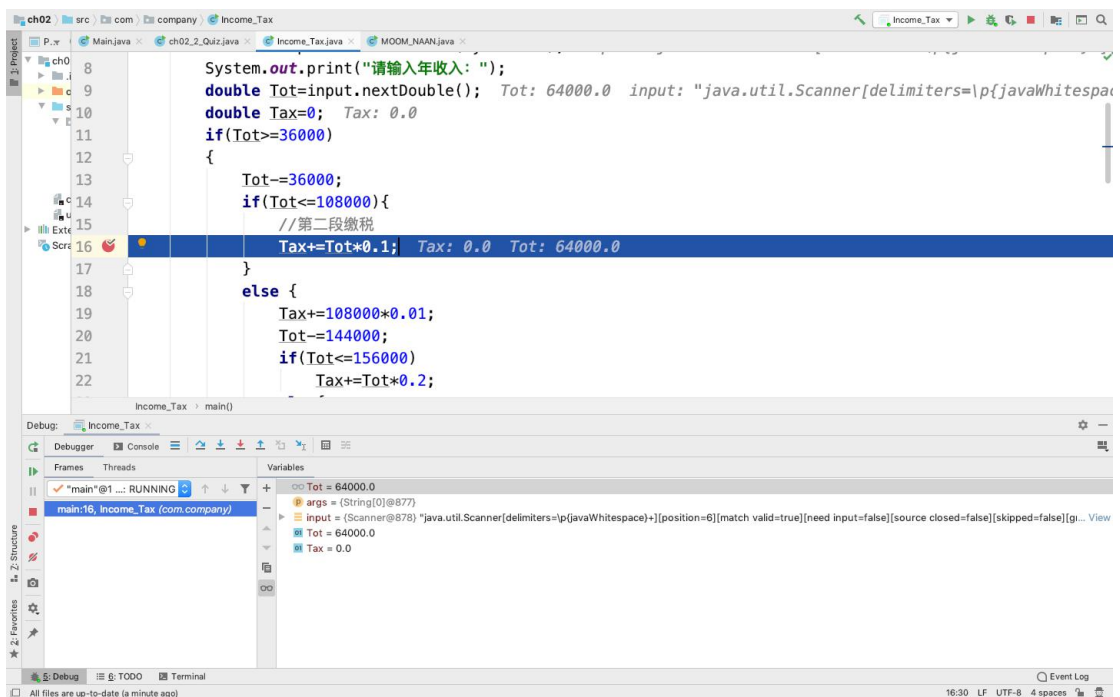
- 1.设置断点



在 debug 开始之前, 可以点击代码的行数后, 设置红色断点, 运行 debug 后, 程序会停止在断点处 (前提是会接触断点), 之后可以再按步骤单步运行或者查看当前的变量值。

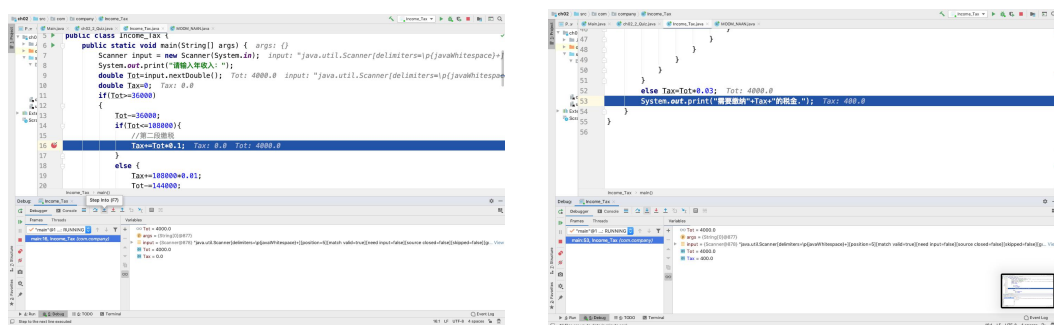
- 2.查看栈区变量

程序停止在断点处时, 在 Variables 一栏中可以查看当前程序中变量的大小。



3.智能步进

程序停止在断点处时，可以选择按步执行，分步查看变量属性。



程序会直接跳转到下一条应执行的位置。

控制步数执行的方法很多他们分别是：

> **Show Execution Point:** 如果你的光标在其它行或其它页面，点击这个按钮可跳转到当前代码执行的行。

> **Step Over:** 步过，一行一行地往下走，如果这一行上有方法不会进入方法。

> **Step Into:** 步入，如果当前行有方法，可以进入方法内部，一般用于进入自定义方法内，不会进入官方类库的方法，如第 25 行的 put 方法。

> **Force Step Into:** 强制步入，能进入任何方法，查看底层源码的时候可以用这个进入官方类库的方法。

> **Step Out:** 步出，从步入的方法内退出到方法调用处，此时方法已执行完毕，只是还没有完成赋值。

> **Drop Frame:** 回退断点。

> **Run to Cursor:** 运行到光标处，你可以将光标定位到你想要查看的那一行，然后使用这个功能，代码会运行至光标行，而不需要打断点。

> **Evaluate Expression:** 计算表达式。

4.其他方法

1.异常断点:

可以设置在发生需要拦截的特殊断点，在程序运行过程中会自动定位并停止。

2.计算表达式

在断点处，可以选择 **Evaluate Expression** 方法，直接计算布尔表达式（或者变量间做数值运算）。

3.特殊的 watch 监视

除了 **variable** 可以查看当前变量，还可以将 **variable** 中的变量拖入 **Watch** 中单独查看与追踪。

问题五：判断字符串是否为回文的

解决思路：

首先需要新建用户输入模块，并用字符串的按位取字的方式取出中央对称的两个字符做比较，只要有一对不相同，判断既为假，回文具有对称性，所以一边只要取到 $\text{Length}/2$ 即可，方便计算。若输入字符串长度为奇数，可以不管最中间一个字符的问题。

代码实现：

```
package com.company;

import java.util.Scanner;

public class MOOM_NAAN {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("请输入一串字符串：");
        String Huiwen = input.nextLine();
        int Length=Huiwen.length();
        boolean Ans=true;
        for(int i=Length-1;i>=0;i--){
            if(Huiwen.charAt(i)!=Huiwen.charAt(Length-i-1))
                Ans=false;
        }
        if (Ans)
            System.out.print(Huiwen+"是回文字符串。");
        else System.out.print(Huiwen+"不是一个回文字符串。");
    }
}
```

效果图展示：

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...
请输入一串字符串： yncacny
yncacny是回文字符串。
Process finished with exit code 0
```

```
/Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java ...  
请输入一串字符串: asdasdq  
asdasdq不是一个回文字符串。  
Process finished with exit code 0
```

通过这次实验，我了解到了字符串的强大功能，位操作都可以使用 **String** 类下的 **charAt** 函数处理。