**《软件测试》**

**实验报告三 ——等价类分析方法**

**姓 名： 黄勇斌 学 号：2019112321**

**院 系： 计算机与信息学院 专 业： 计算机科学与技术**

**实 验 室： 实验日期：2022/3/21**

**总评成绩： 审阅教师：**

目录

[**《软件测试》** 1](#_Toc98773437)

[**实验报告三 ——等价类分析方法** 1](#_Toc98773438)

[一、实验目的： 3](#_Toc98773439)

[二、实验环境： 3](#_Toc98773440)

[三、实验要求： 3](#_Toc98773441)

[四、实验步骤与内容 3](#_Toc98773442)

[五、结论分析与体会 7](#_Toc98773443)

[六、仓库地址 8](#_Toc98773444)

### 一、实验目的：

1. 巩固 git 协作式管理工具的使用；
2. 巩固 gradle 项目构建 Java 应用的基本操作
3. 掌握等价类分析方法设计测试用例的方法

### 二、实验环境：

Eclipse2020,JUnit

### 三、实验要求：

1. 实现电话账单收费功能
2. 采用边界值分析方法设计测试用例，并采用Junit5 编写电话账单收费程序测试用例，并提交到代码仓库
3. 按照[实验报告模板](https://star.jmhui.com.cn/u/cms/www/202203/06160105hi3m.docx) 编写实验报告，以“学号-姓名-软件测试实验三”命名，提交到雨课堂“软件测试实验三”

### 四、实验步骤与内容

1. **题目：电话账单计费**

对于电话账单来说，当春季和秋季标准时间与夏令时时间进行转换时会产生一个很有意思的问题: 春季，这种转换发生在(3月某个)星期日凌晨2:00点，这时要将时钟设置为凌晨3:00点；秋季， 转换通常在11月的第一个星期日，时钟要从2:59:59调回2:00:00。  
请为长途电话服务函数开发计费类，使用等价类分析方法构建测试用例

**采用如下计费规则计算通话费：**

通话时间小于等于20分钟时，每分钟收费0.05美元，通话时间不足1分钟按1分钟计算。

通话时间大于20分钟时，收费1.00美元，外加超过20分钟的部分每分钟0.10美元；

不到1分钟按1分钟计算

**假设：**

通话计费时间从被叫方应答开始计算，到呼叫方挂机时结束；

通话时间的秒数向上进位到分钟；

没有超过30个小时的通话。

1. **测试用例：**

|  |  |  |  |
| --- | --- | --- | --- |
| 开始时间 | 结束时间 | 预测输出 | 实际输出 |
| 20220321161205 | 20220321161206 | 0.05 | 0.05 |
| 20220321161205 | 20220321161105 | 输入错误 | 输入错误 |
| 20220321161205 | 20220322161205 | 143.00 | 143.00 |
| 20220321161205 | 20230321161205 | 输入错误 | 输入错误 |

1. **代码实现：**

|  |
| --- |
| **实现代码：**  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class Bill {  private String startingTime;  private String endingTime;  private long minute;//通话时长  public Date startT;  public Date endT;  public Date StrToDate(String str) {  SimpleDateFormat format = new SimpleDateFormat("yyyyMMddHHmmss");  Date date = null;  try {  date = format.parse(str);  } catch (ParseException e) {  e.printStackTrace();  }  return date;  }  public String DateFormate(Date date){  SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss E");  String sdate = format.format(date);  return sdate;  }  public String getBill(String startingTime,String endingTime){  this.startingTime = startingTime;  this.endingTime = endingTime;  this.startT = StrToDate(this.startingTime);  this.endT = StrToDate(this.endingTime);  long between;  Calendar cal1 = Calendar.getInstance();  cal1.setTime(startT);  Calendar cal2 = Calendar.getInstance();  cal2.setTime(endT);  between = cal2.getTime().getTime()-cal1.getTime().getTime();  if(between<0)  minute =( (cal2.getTime().getTime()-cal1.getTime().getTime())/1000)/60;//得到通话时长  else  minute = (between/1000+59)/60;  //判断接通时间是否在发生转换的日期  if((cal1.get(Calendar.MONTH) ==Calendar.MARCH && cal1.get(Calendar.DATE) >= 29 && cal1.get(Calendar.DATE) <= 31 && cal1.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)  || ((cal1.get(Calendar.MONTH) ==Calendar.APRIL && cal1.get(Calendar.DATE) >= 1 && cal1.get(Calendar.DATE) <= 4 && cal1.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)))  {  System.out.println("需要转换1");  if(cal1.get(Calendar.HOUR\_OF\_DAY)<2 && cal2.get(Calendar.HOUR\_OF\_DAY)>=3)  minute -=60;//时间从2点转换到3点减去1h  }else{  if((cal2.get(Calendar.MONTH) ==3 && cal2.get(Calendar.DATE) >= 29 && cal2.get(Calendar.DATE) <= 31 && cal2.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)  || ((cal2.get(Calendar.MONTH) ==4 && cal2.get(Calendar.DATE) >= 1 && cal2.get(Calendar.DATE) <= 4 && cal2.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)))  //判断结束时间是否在转换的日期  {  if(cal2.get(Calendar.HOUR\_OF\_DAY)>3 )  minute -=60;  }  }  //将位于转换的时间段内的时间都看作已经转换之后的  if(cal1.get(Calendar.MONTH) ==Calendar.OCTOBER && cal1.get(Calendar.DATE) >= 25 && cal1.get(Calendar.DATE) <= 31 && cal1.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)  {  if(cal1.get(Calendar.HOUR\_OF\_DAY) ==2){  minute +=60;  }  }else{  if(cal2.get(Calendar.MONTH) ==Calendar.OCTOBER && cal2.get(Calendar.DATE) >= 25 && cal2.get(Calendar.DATE) <= 31 && cal2.get(Calendar.DAY\_OF\_WEEK)==Calendar.SUNDAY)  {  if(cal2.get(Calendar.HOUR\_OF\_DAY)==2)  {  minute +=60;  }  }  }  if(minute>1800 || minute <0){  return "输入错误";  }else{  double fare = 0;  if(minute <= 20)  fare = (minute \* 0.05);  else  fare = 1 + (minute-20)\*0.1;  return String.format("%.2f", fare);  }  }  }  测试代码：  import org.junit.jupiter.params.ParameterizedTest;  import org.junit.jupiter.params.provider.CsvSource;  import static org.junit.jupiter.api.Assertions.\*;  class mainTest {  Bill bill = new Bill();  @ParameterizedTest  @CsvSource({  "20220321161205, 20220321161206, 0.05",  "20220321161205, 20220321161105, 输入错误",  "20220321161205, 20220322161205, 143.00",  "20220321161205, 20230321161205, 输入错误"  })  //输入year-month-day-minute-seconds 组成的字符串  void test(String startTime, String endTime, String results){  assertEquals(results, bill.getBill(startTime, endTime));  }  }  测试结果：    jacoco报告  checkstyle报告 |

### 五、结论分析与体会

这次实验中在老师的细心带领之下，成功解决了上节课遇到的问题。

同时checkstyle检测下发现自己代码存在很多不规范的错误，导致风格错误达到143处，今后要好好的修改下自己代码的习惯

### 六、仓库地址

<https://github.com/hyb1041739742/Software-Testing>