**实验一 黑盒测试**

班级：信1405-1班 学号：20142863 姓名：杨程鑫

**一、实验目的**

* 1. 掌握黑盒测试的基础知识；
  2. 掌握黑盒测试的检查内容及测试目的；
  3. 掌握黑盒测试的几种基本测试方法：等价类划分方法、边界值分析方法、因果图法和决策表法；

**二、实验要求**

1. 复习有关内容，理解黑盒测试；
2. 掌握等价类划分、边界值分析方法、因果图法和决策表法，并能设计出测试用例；
3. 对具体软件，能分别使用相应的黑盒测试方法设计测试用例，并实施测试、分析测试结果。

**三、实验内容**

**第1题：设计函数实现输入日期显示星期几，并用等价类及边界值法测试**

**源代码：**

public class MainClass

{

private static Component frame = null;

public static void main(String args[]) throws ParseException

{

String dataString = "";

ConvertTools convertTools = new ConvertTools();

while(true)

{

dataString = JOptionPane.showInputDialog("请输入日期,(00010101)");

if(dataString == null)

{

System.exit(0);

}

if(dataString.length() != 8)

{

JOptionPane.showMessageDialog(frame ,"输入日期长度应为8。");

continue;

}

if(convertTools.IsGreaterThanFiveNumber(dataString))

{

if(convertTools.isDate(dataString))

{

break;

}

else

{

JOptionPane.showMessageDialog(frame ,"月份或日期不在范围。");

}

}

else

{

JOptionPane.showMessageDialog(frame ,"输入日期不满足格式。");

}

}

Date data = convertTools.convertToDate(dataString);

String[] info = data.toString().split(" ");

System.out.println(info[0]);

}

}

**等价类划分**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 唯一标识 | 无效等价类 | 唯一标识 |
| 输入八位整数 | 整数 | 1 | 小数 | 2 |
| 含有其他字符串 | 3 |
| 八位 | 4 | 大于八位 | 5 |
| 小于八位 | 6 |
| 年份 | 闰年 | 7 | Year<0 | 9 |
| 非闰年 | 8 |
| 月份 | 0<Month<13 | 10 | Month<1 | 11 |
| Month>12 | 12 |
| 0<Day<29 | 13 | Day<1 | 17 |
| 日期 | 29 | 14 |
| 30 | 15 | Day>31 | 18 |
| 31 | 16 |

**测试用例表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入数据 | 覆盖等价类 | 输出结果 |
| 1 | 20000228 | 1，4，7，10，13 | Mon |
| 2 | 20011229 | 1，4，8，10，14 | Sat |
| 3 | 20000230 | 1，4，7，10，15 | 月份或日期不在范围 |
| 4 | 20011131 | 1，4，8，10，16 | 月份或日期不在范围 |
| 5 | 2000.228 | 2，4，7，10，13 | 输入不满足格式 |
| 6 | 2001a229 | 3，4，8，10，14 | 输入不满足格式 |
| 7 | 020001230 | 1，5，7，10，15 | 输入长度应为8 |
| 8 | 2000231 | 1，6，8，10，16 | 输入长度应为8 |
| 9 | -1000228 | 1，4，9，10，13 | 输入不满足格式 |
| 11 | 20000030 | 1，4，7，11，15 | 月份或日期不在范围 |
| 12 | 20011331 | 1，4，8，12，16 | 月份或日期不在范围 |
| 13 | 200012-1 | 1，4，7，10，17 | 输入不满足格式 |
| 14 | 20011232 | 1，4，8，10，18 | 月份或日期不在范围 |

**边界法取值表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Min- | Min | Nom | Max | Max+ |
| Year | 0000 | 0001 | 5000 | 9999 | 10000 |
| Month | 00 | 01 | 06 | 12 | 13 |
| Day | 00 | 01 | 16 | 31 | 32 |

**边界法测试用例表**

|  |  |  |
| --- | --- | --- |
| 序号 | 组合结果 | 输出结果 |
| 1 | 50000616 | Mon |
| 2 | 00000616 | 输入不满足格式 |
| 3 | 00010616 | Thu |
| 4 | 99990616 | Wed |
| 5 | 100000616 | 输入长度应为8 |
| 6 | 50000016 | 月份或日期不在范围 |
| 7 | 50000116 | Thu |
| 8 | 50001216 | Tue |
| 9 | 50001316 | 月份或日期不在范围 |
| 10 | 50000600 | 月份或日期不在范围 |
| 11 | 50000601 | Sun |
| 12 | 50000631 | 月份或日期不在范围 |
| 13 | 50000632 | 月份或日期不在范围 |

**第二题:找零钱最佳组合**

**源代码：**

public class MainClass

{

private static Component frame;

public static void main(String[] args)

{

int cost = 0;

int money = 0;

ConvertTools convertTools = new ConvertTools();

while(true)

{

String costString = JOptionPane.showInputDialog("请输入花销,(1-100)");

if(costString == null)

{

System.exit(0);

}

if(!convertTools.isNumber(costString))

{

JOptionPane.showMessageDialog(frame,"请输入正确花销");

continue;

}

cost = Integer.valueOf(costString);

if(cost >= 1 && cost <= 100)

{

break;

}

else

{

JOptionPane.showMessageDialog(frame,"花销不在范围");

}

}

while(true)

{

String moneyString = JOptionPane.showInputDialog("请输入钱数,(1-100)");

if(moneyString == null)

{

System.exit(0);

}

if(!convertTools.isNumber(moneyString))

{

JOptionPane.showMessageDialog(frame,"请输入正确钱数");

}

money = Integer.valueOf(moneyString);

if(money >= 1 && money <= 100)

{

if(money >= cost)

{

break;

}

else

{

JOptionPane.showMessageDialog(frame,"钱数应大于花销");

}

}

else

{

JOptionPane.showMessageDialog(frame,"钱数不在范围");

}

}

int change = money - cost;

int numberOfN50 = 0;

int numberOfN10 = 0;

int numberOfN5 = 0;

int numberOfN1 = 0;

boolean isOver = false;

if(change >= 50 && !isOver)

{

numberOfN50 = change / 50;

change %= 50;

if(change == 0)

{

isOver = true;

}

}

if(change >= 10 && !isOver)

{

numberOfN10 = change / 10;

change %= 10;

if(change == 0)

{

isOver = true;

}

}

if(change >= 5 && !isOver)

{

numberOfN5 = change / 5;

change %= 5;

if(change == 0)

{

isOver = true;

}

}

if(change >= 1 && !isOver)

{

numberOfN1 = change / 1;

change %= 1;

if(change == 0)

{

isOver = true;

}

}

if(change == 0)

{

isOver = true;

}

if(isOver)

{

System.out.print("50元:\t" + numberOfN50 + "张\n"

+ "10元:\t" + numberOfN10 + "张\n"

+ "5元:\t" + numberOfN5 + "张\n"

+ "1元:\t" + numberOfN1 + "张");

}

}

}

**划分等价类**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 唯一标识 | 无效等价类 | 唯一标识 |
| 花销/元 | 50-100 | 1 | >100 | 5 |
| 10-49 | 2 |
| 5-9 | 3 | <1 | 6 |
| 1-4 | 4 |
| 付款/元 | 50-100 | 7 | >100 | 11 |
| 10-49 | 8 |
| 5-9 | 9 | <1 | 12 |
| 1-4 | 10 |
| 非整数 | - | - | 小数 | 13 |
| 含有其他字符 | 14 |

**测试用例表**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 输入数据(花销/付款) | | 覆盖等价类 | 输出结果(N50,N10,N5,N1) | | | |
| 1 | 99 | 1 | 1，10 | 付款应大于花销 | | | |
| 2 | 49 | 5 | 2，9 | 付款应大于花销 | | | |
| 3 | 9 | 10 | 3，8 | 0 | 0 | 0 | 1 |
| 4 | 4 | 50 | 4，7 | 0 | 4 | 1 | 1 |
| 5 | 101 | - | 5 | 花销不在范围 | | | |
| 6 | 0 | - | 6 | 花销不在范围 | | | |
| 7 | - | 101 | 11 | 付款不在范围 | | | |
| 8 | - | 0 | 12 | 付款不在范围 | | | |
| 9 | 10.0 | - | 13 | 请输入正确花销 | | | |
| 10 | - | 10a | 14 | 请输入正确付款 | | | |

**边界法取值表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Min | Min+ | Nom | Max- | Max |
| 花销 | 1 | 2 | 50 | 99 | 100 |
| 付款 | 1 | 2 | 50 | 99 | 100 |

**边界法测试用例表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 组合结果(花销/付款) | 输出结果(N50,N10,N5,N1) | | | |
| 1 | 50，50 | 0 | 0 | 0 | 0 |
| 2 | 1，50 | 0 | 4 | 1 | 4 |
| 3 | 2，50 | 0 | 4 | 1 | 3 |
| 4 | 99，50 | 付款应大于花销 | | | |
| 5 | 100，50 | 付款应大于花销 | | | |
| 6 | 50，1 | 付款应大于花销 | | | |
| 7 | 50，2 | 付款应大于花销 | | | |
| 8 | 50，99 | 付款应大于花销 | | | |
| 9 | 50，100 | 付款应大于花销 | | | |

**第三题:客房预订系统**

**源代码：**

public class MainFrame

{

private JFrame frame;

public static void main(String[] args)

{

EventQueue.invokeLater(new Runnable()

{

public void run()

{

try

{

MainFrame window = new MainFrame();

window.frame.setVisible(true);

}

catch(Exception e)

{

e.printStackTrace();

}

}

});

}

public MainFrame()

{

initialize();

}

private void initialize()

{

frame = new JFrame();

frame.setTitle("客房预订系统");

frame.setBounds(450,170,420,275);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel costTypeLabel = new JLabel("房款支付");

costTypeLabel.setHorizontalAlignment(SwingConstants.CENTER);

costTypeLabel.setBounds(250,40,60,20);

frame.getContentPane().add(costTypeLabel);

ButtonGroup buttonGroup = new ButtonGroup();

JRadioButton fullPaymentRadio = new JRadioButton("支付全款");

fullPaymentRadio.setHorizontalAlignment(SwingConstants.CENTER);

fullPaymentRadio.setBounds(180,75,100,25);

buttonGroup.add(fullPaymentRadio);

frame.getContentPane().add(fullPaymentRadio);

JRadioButton depositPaymentRadio = new JRadioButton("支付定金");

depositPaymentRadio.setBounds(300,75,100,25);

buttonGroup.add(depositPaymentRadio);

frame.getContentPane().add(depositPaymentRadio);

JLabel roomTypeLabel = new JLabel("房间类型");

roomTypeLabel.setHorizontalAlignment(SwingConstants.CENTER);

roomTypeLabel.setBounds(175,145,60,20);

frame.getContentPane().add(roomTypeLabel);

JButton singleRoomButton = new JButton("单人间");

singleRoomButton.addMouseListener(new MouseAdapter()

{

private Component frameTemp;

public void mouseClicked(MouseEvent e)

{

if(fullPaymentRadio.isSelected())

{

FullPaymentSuccess fullPaymentSuccess = new FullPaymentSuccess

(fullPaymentRadio.getText(),singleRoomButton.getText());

fullPaymentSuccess.setVisible(true);

}

else if(depositPaymentRadio.isSelected())

{

DepositPaymentWarning depositPaymentWarning = new DepositPaymentWarning

(depositPaymentRadio.getText(),singleRoomButton.getText());

depositPaymentWarning.setVisible(true);

}

else

{

JOptionPane.showMessageDialog(frameTemp,"请选择支付类型");

}

}

});

singleRoomButton.setBounds(40,180,90,25);

frame.getContentPane().add(singleRoomButton);

JButton doubleRoomButton = new JButton("双人间");

doubleRoomButton.addMouseListener(new MouseAdapter()

{

private Component frameTemp;

public void mouseClicked(MouseEvent e)

{

if(fullPaymentRadio.isSelected())

{

FullPaymentSuccess fullPaymentSuccess = new FullPaymentSuccess(

fullPaymentRadio.getText(),doubleRoomButton.getText());

fullPaymentSuccess.setVisible(true);

}

else if(depositPaymentRadio.isSelected())

{

DepositPaymentWarning depositPaymentWarning = new DepositPaymentWarning

(depositPaymentRadio.getText(),singleRoomButton.getText());

depositPaymentWarning.setVisible(true);

}

else

{

JOptionPane.showMessageDialog(frameTemp,"请选择支付类型");

}

}

});

doubleRoomButton.setBounds(160,180,90,25);

frame.getContentPane().add(doubleRoomButton);

JButton DeluxeRoomButton = new JButton("豪华间");

DeluxeRoomButton.addMouseListener(new MouseAdapter()

{

private Component frameTemp;

public void mouseClicked(MouseEvent e)

{

if(fullPaymentRadio.isSelected())

{

FullPaymentSuccess fullPaymentSuccess = new FullPaymentSuccess(

fullPaymentRadio.getText(),DeluxeRoomButton.getText());

fullPaymentSuccess.setVisible(true);

}

else if(depositPaymentRadio.isSelected())

{

DepositPaymentWarning depositPaymentWarning = new DepositPaymentWarning

(depositPaymentRadio.getText(),singleRoomButton.getText());

depositPaymentWarning.setVisible(true);

}

else

{

JOptionPane.showMessageDialog(frameTemp,"请选择支付类型");

}

}

});

DeluxeRoomButton.setBounds(280,180,90,25);

frame.getContentPane().add(DeluxeRoomButton);

JPanel panel = new JPanel()

{

private static final long serialVersionUID = 1L;

public void paint(Graphics g)

{

g.drawImage(Toolkit.getDefaultToolkit().getImage("src\\test1\_3\\image.png"),0,0,this);

}

};

panel.setBounds(44,30,116,89);

frame.getContentPane().add(panel);

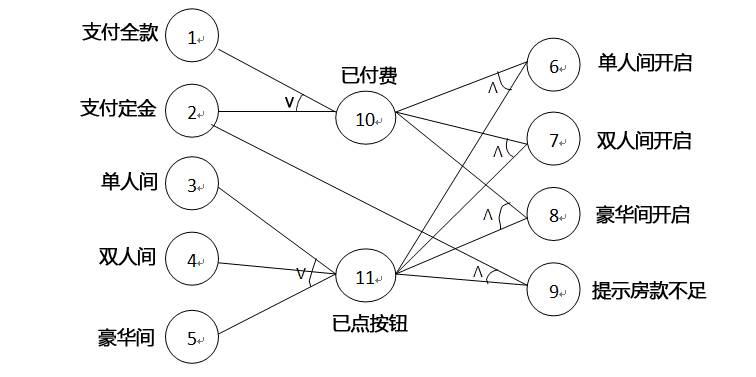
}

}

原因：支付全款，支付定金，选择单人间按钮，选择双人间按钮，选择豪华间按钮；

结果：单人间开启，双人间开启，豪华间开启，提示房款不足。

**因果图：**



**决策表：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 支付全款 | T | T | T | F | F | F | - |
| 支付定金 | F | F | F | T | T | T | - |
| 单人间 | T | F | F | T | F | F | - |
| 双人间 | F | T | F | F | T | F | - |
| 豪华间 | F | F | T | F | F | T | - |
| 不可能情况 | F | F | F | F | F | F | T |
| 对应房间开启 | T | T | T | T | T | T | F |
| 提示房款不足 | F | F | F | T | T | T | F |

**测试用例表：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 支付情况 | 选择的房间 | 预期结果 |
| 1 | 支付全款 | 单人间 | 单人间开启成功 |
| 2 | 支付全款 | 双人间 | 双人间开启成功 |
| 3 | 支付全款 | 豪华间 | 豪华间开启成功 |
| 4 | 支付定金 | 单人间 | 单人间开启成功，但房款不足 |
| 5 | 支付定金 | 双人间 | 双人间开启成功，但房款不足 |
| 6 | 支付定金 | 豪华间 | 豪华间开启成功，但房款不足 |

**第四题：航空服务查询问题**

**源代码：**

public class MainClass

{

private JFrame frame;

public static void main(String[] args)

{

EventQueue.invokeLater(new Runnable()

{

public void run()

{

try

{

MainClass window = new MainClass();

window.frame.setVisible(true);

}

catch(Exception e)

{

e.printStackTrace();

}

}

});

}

public MainClass()

{

initialize();

}

private void initialize()

{

frame = new JFrame();

frame.setBounds(100,100,340,350);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JPanel airLinePanel = new JPanel();

airLinePanel.setBounds(33, 22, 260, 48);

frame.getContentPane().add(airLinePanel);

airLinePanel.setLayout(null);

JLabel airLineLabel = new JLabel("航线：");

airLineLabel.setBounds(10, 2, 60, 20);

airLinePanel.add(airLineLabel);

ButtonGroup airLineButtonGroup = new ButtonGroup();

JRadioButton airLineType1 = new JRadioButton("欧美");

airLineType1.setSelected(true);

airLineType1.setBounds(5, 20, 60, 20);

airLinePanel.add(airLineType1);

airLineButtonGroup.add(airLineType1);

JRadioButton airLineType2 = new JRadioButton("国外非欧美");

airLineType2.setBounds(75, 20, 100, 20);

airLinePanel.add(airLineType2);

airLineButtonGroup.add(airLineType2);

JRadioButton airLineType3 = new JRadioButton("国内");

airLineType3.setBounds(185, 20, 60, 20);

airLinePanel.add(airLineType3);

airLineButtonGroup.add(airLineType3);

JPanel typePanel = new JPanel();

typePanel.setBounds(33, 90, 260, 48);

frame.getContentPane().add(typePanel);

typePanel.setLayout(null);

JLabel typeLabel = new JLabel("航位：");

typeLabel.setBounds(10, 2, 60, 20);

typePanel.add(typeLabel);

ButtonGroup typeButtonGroup = new ButtonGroup();

JRadioButton businessClassRadio = new JRadioButton("商务舱");

businessClassRadio.setSelected(true);

businessClassRadio.setBounds(40, 20, 80, 20);

typeButtonGroup.add(businessClassRadio);

typePanel.add(businessClassRadio);

JRadioButton touristClassRadio = new JRadioButton("经济舱");

touristClassRadio.setBounds(140, 20, 80, 20);

typeButtonGroup.add(touristClassRadio);

typePanel.add(touristClassRadio);

JPanel timePanel = new JPanel();

timePanel.setBounds(33, 159, 260, 48);

frame.getContentPane().add(timePanel);

timePanel.setLayout(null);

JLabel timeLabel = new JLabel("飞行时间：");

timeLabel.setBounds(10, 2, 60, 20);

timePanel.add(timeLabel);

ButtonGroup timeButtonGroup = new ButtonGroup();

JRadioButton withinTwoRadio = new JRadioButton("两小时以内");

withinTwoRadio.setSelected(true);

withinTwoRadio.setBounds(40, 20, 100, 20);

timeButtonGroup.add(withinTwoRadio);

timePanel.add(withinTwoRadio);

JRadioButton moreThanTwoRadio = new JRadioButton("超过两小时");

moreThanTwoRadio.setBounds(140, 20, 100, 20);

timeButtonGroup.add(moreThanTwoRadio);

timePanel.add(moreThanTwoRadio);

JLabel resultLabel = new JLabel("航空服务：");

resultLabel.setBounds(50, 230, 60, 20);

frame.getContentPane().add(resultLabel);

JTextField resultTextField = new JTextField();

resultTextField.setBounds(120, 230, 150, 20);

frame.getContentPane().add(resultTextField);

resultTextField.setColumns(10);

JButton selectButton = new JButton("查询");

selectButton.addMouseListener(new MouseAdapter()

{

public void mouseClicked(MouseEvent e)

{

if(airLineType1.isSelected())

{

resultTextField.setText("有食物供应，可播放电影");

}

else if(airLineType2.isSelected())

{

if(businessClassRadio.isSelected())

{

resultTextField.setText("有食物供应，可播放电影");

}

else

{

resultTextField.setText("有食物供应");

}

}

else if(airLineType3.isSelected())

{

if(businessClassRadio.isSelected())

{

resultTextField.setText("有食物供应");

}

else if(touristClassRadio.isSelected())

{

if(moreThanTwoRadio.isSelected())

{

resultTextField.setText("有食物供应");

}

else

{

resultTextField.setText("无服务");

}

}

}

else

{

resultTextField.setText("无服务");

}

}

});

selectButton.setBounds(45, 265, 90, 25);

frame.getContentPane().add(selectButton);

JButton exitButton = new JButton("退出");

exitButton.addMouseListener(new MouseAdapter()

{

public void mouseClicked(MouseEvent e)

{

System.exit(0);

}

});

exitButton.setBounds(180, 265, 90, 25);

frame.getContentPane().add(exitButton);

}

}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 欧美 | T | F | F | F | F | F |
| 国内 | - | F | F | T | T | T |
| 商务舱 | - | T | F | T | F | F |
| 超过2h | - | - | - | - | T | F |
| 不可能 | F | F | F | F | F | F |
| 有食物 | T | T | T | T | T | F |
| 看电影 | T | T | F | F | F | F |

**测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 航线 | 航位 | 飞行时间 | 预期输出 |
| 1 | 欧美 | 经济舱 | 不超过2h | 有食物和电影 |
| 2 | 国外非欧美 | 商务舱 | 不超过2h | 有食物和电影 |
| 3 | 国外非欧美 | 经济舱 | 不超过2h | 有食物，无电影 |
| 4 | 国内 | 商务舱 | 不超过2h | 有食物，无电影 |
| 5 | 国内 | 经济舱 | 超过2h | 有食物，无电影 |
| 6 | 国内 | 经济舱 | 不超过2h | 无食物和电影 |

**四、实验思考**

① 在实际的测试中，如何设计测试用例才能达到用最少的测试用例检测出最多的缺陷；

② 在进行用例设计时，如何考虑软件测试用例的充分性和减少软件测试用例的冗余性；

我觉得测试用例都是在某一方面达到了测试的目的，在该方面会很贴切，但是总是有遗漏的方面，所以并不是设计一种用例就可以完成测试的。有的特殊情况可能什么方法效果都一般，所以才有经验法。用最少的测试用例检测出最多的缺陷本就是矛盾的，像等价类覆盖和决策表的方法，用排列组合的方式组合用例，虽然会有一些无效的用例，但无疑这样很全。最大覆盖和合并一些情况所出的结果也都是建立在问题和人的经验之上的，我觉得这样可能还会出现其他的错误。所谓测试就应该建立在客观的角度遍历所有的情况。但是又因为测试的可行性，所以不得做出调整。

调整，根据以上的几个例子。为了测试充分，我是根据题意和决策表来设计测试用例的，如第四题中航线只要是欧美，那么无所谓航位和飞行时间，所以以航线为欧美的四组测试用例都可以合并为一个。为了减少冗余性，其实在设计决策表时就应该考虑了，和上面一样，也是根据题意设计的第一列，以欧美航线为首的情况。有简略就可能会出现缺陷。因为程序就是以这种思想设计的，测试也用这种思想，如果没有错误还好，如果当时犯了错误，测试也会过滤掉那种错误。