**《面向对象程序设计Java》作业789**

班级： 学号： 姓名：

**/\* 上面的信息请填写完整，然后完成下面的要求，最后将本文档在BB平台上提交 \*/**

**- Experiment 7 -**

//把你最后完成的第**1**小题的Java程序复制粘贴在下面，然后对各个运行结果进行分析比较

public class ExceptionTest {

  public static void main(String[] args) {

    String output[] = {

      "The",

      "brown",

      "fox",

      "jumps",

      "over",

      "the",

      "lazy",

      "dog",

    };

    int i = 0;

*while* (i < 10) {

*try* {

        System.out.println(output[i++]);

      } *catch* (ArrayIndexOutOfBoundsException e) {

        System.out.println("\n 下标越界异常处理！");

        System.out.println(e);

*break*;

      } *finally* {

        System.out.println("Execute no matter what!");

      }

    }

    System.out.println("Haha...");

  }

}

//把你编写（并运行正常）的第**2**小题的Java程序复制粘贴在下面

import java.util.Scanner;

*/\*\**

*\* MyExceptionTestExceptionTest*

*\*/*

public class MyExceptionTest {

  public static int multiply(int a, int b) throws MyException {

    int result;

    result = a \* b;

*if* (result < -32768 || result > 32767) {

*throw* *new* MyException("Out of range of [-32768, 32767]");

    }

*return* result;

  }

  public static void main(String[] args) {

    Scanner scanner = *new* Scanner(System.in);

    out*:while* (true) {

      System.out.println("Input 2 integer: ");

      int a = scanner.nextInt(), b = scanner.nextInt();

*if* (a == 0 && b == 0) {

*break* out;

      }

*try* {

        System.out.println(a + "\*" + b + "=" + multiply(a, b));

      } *catch* (MyException e) {

        System.out.println(e);

      }

    }

    scanner.close();

  }

}

class MyException extends Exception {

  public MyException() {

*super*();

  }

  public MyException(String msg) {

*super*(msg);

  }

}

//把你编写（并运行正常）的第**3**小题的Java程序复制粘贴在下面

import java.util.Scanner;

public class TriangleTest {

  public static void main(String[] args) {

    Triangle triangle = *new* Triangle();

    Scanner scanner = *new* Scanner(System.in);

    double a, b, c;

*while* (true) {

      System.out.println("Input 3 sides of a triangle: ");

      a = scanner.nextDouble();

      b = scanner.nextDouble();

      c = scanner.nextDouble();

*if* (a == 0 && b == 0 && c == 0) {

        System.out.println("Bye!");

*break*;

      }

*try* {

        triangle.setSides(a, b, c);

        System.out.println(triangle);

      } *catch* (TriangleException e) {

        System.out.println(e);

      }

    }

    scanner.close();

  }

}

class Triangle {

  private double a, b, c;

  public Triangle() {}

  public void setSides(double a, double b, double c) throws TriangleException {

*if* (a \* b \* c != 0 && a + b > c && a + c > b && b + c > a) {

*this*.a = a;

*this*.b = b;

*this*.c = c;

    } *else* {

*throw* *new* TriangleException("Invalid sides!");

    }

  }

  public double area() {

    double p = (a + b + c) / 2;

*return* Math.sqrt(p \* (p - a) \* (p - b) \* (p - c));

  }

  public double perimeter() {

*return* a + b + c;

  }

  @Override

  public String toString() {

*return* (

      "Triangle [a=" +

      a +

      ", b=" +

      b +

      ", c=" +

      c +

      ", area=" +

      area() +

      ", perimeter=" +

      perimeter() +

      "]"

    );

  }

}

class TriangleException extends Exception {

  private static final long serialVersionUID = 1L;

  public TriangleException() {

*super*();

  }

  public TriangleException(String message) {

*super*(message);

  }

}

//把你编写（并运行正常）的第**4**小题的Java程序复制粘贴在下面

*/\*\**

*\* ComplexTest*

*\*/*

public class ComplexTest {

  static Complex testCases[][] = {

    { *new* Complex(1, 2), *new* Complex(0, 1) },

    { *new* Complex(1, 2), *new* Complex(0, 0) },

    { *new* Complex(1, 2), *new* Complex(3, 4) },

    { *new* Complex(1, 2), *new* Complex(2, 0) },

  };

  public static void main(String[] args) {

*for* (Complex[] testCase *:* testCases) {

      System.out.println("c1: " + testCase[0]);

      System.out.println("c2: " + testCase[1]);

      System.out.println("c1 + c2: " + Complex.add(testCase[0], testCase[1]));

      System.out.println("c1 - c2: " + Complex.sub(testCase[0], testCase[1]));

      System.out.println("c1 \* c2: " + Complex.mul(testCase[0], testCase[1]));

*try* {

        System.out.println("c1 / c2: " + Complex.div(testCase[0], testCase[1]));

      } *catch* (ComplexException e) {

        System.out.println(e);

      }

    }

  }

}

class Complex {

  private double real, imag;

  public Complex() {

*this*(0, 0);

  }

  public Complex(double real, double imag) {

*this*.real = real;

*this*.imag = imag;

  }

  @Override

  public String toString() {

*return* real + " + " + imag + "i";

  }

  public static Complex add(Complex c1, Complex c2) {

*return* *new* Complex(c1.real + c2.real, c1.imag + c2.imag);

  }

  public static Complex sub(Complex c1, Complex c2) {

*return* *new* Complex(c1.real - c2.real, c1.imag - c2.imag);

  }

  public static Complex mul(Complex c1, Complex c2) {

*return* *new* Complex(

      c1.real \* c2.real - c1.imag \* c2.imag,

      c1.real \* c2.imag + c1.imag \* c2.real

    );

  }

  public static Complex div(Complex c1, Complex c2) throws ComplexException {

*if* (c2.real == 0 && c2.imag == 0) {

*throw* *new* ComplexException("Divisor cannot be zero!");

    }

    double denominator = c2.real \* c2.real + c2.imag \* c2.imag;

*return* *new* Complex(

      (c1.real \* c2.real + c1.imag \* c2.imag) / denominator,

      (c1.imag \* c2.real - c1.real \* c2.imag) / denominator

    );

  }

}

class ComplexException extends Exception {

  private static final long serialVersionUID = 1L;

  public ComplexException() {

*super*();

  }

  public ComplexException(String message) {

*super*(message);

  }

}

**/\* 如果某题一直没能调试通过，请在下面补充说明 \*/**

**题号：**

**源程序代码：**

**编译（或运行）出现的问题描述：**

**- Experiment 8 -**

//把你编写（并运行正常）的第**1**小题的Java程序复制粘贴在下面

import java.util.ArrayList;

*/\*\**

*\* ArrayListTest*

*\*/*

public class ArrayListTest {

  public static void main(String[] args) {

    ArrayList<String> names = *new* ArrayList<String>();

    names.add("James");

    names.add("Tom");

    names.add("Steven");

    names.add("Alice");

    names.add("Bob");

    names.add("Eve");

    names.add("Mary");

    names.add("Jerry");

*for* (int i = 0; i < names.size(); i++) {

      System.out.print(names.get(i) + " ");

    }

    System.out.println();

    names.remove("Tom");

    names.sort(null);

*for* (String string *:* names) {

      System.out.print(string + " ");

    }

    System.out.println();

  }

}

//把你编写（并运行正常）的第**2**小题的Java程序复制粘贴在下面

import java.util.LinkedList;

public class LinkedListTest {

  public static void main(String[] args) {

    LinkedList<String> names = *new* LinkedList<String>();

    names.add("James");

    names.add("Tom");

    names.add("Steven");

    names.add("Alice");

    names.add("Bob");

    names.add("Eve");

    names.add("Mary");

    names.add("Jerry");

*for* (int i = 0; i < names.size(); i++) {

      System.out.print(names.get(i) + " ");

    }

    System.out.println();

    names.remove("Tom");

    names.sort(null);

*for* (String string *:* names) {

      System.out.print(string + " ");

    }

    System.out.println();

  }

}

//把你编写（并运行正常）的第**3**小题的Java程序复制粘贴在下面

interface Arithmetical<T> {

    T add(T obj);

    T subtract(T obj);

    T multiply(T obj);

    T divide(T obj);

}

class Complex implements Arithmetical<Complex> {

    double real, imag;

    Complex(double r, double i) {

        real = r;

        imag = i;

    }

    public Complex add(Complex obj) {

*return* *new* Complex(real + obj.real, imag + obj.imag);

    }

    public Complex subtract(Complex obj) {

*return* *new* Complex(real - obj.real, imag - obj.imag);

    }

    public Complex multiply(Complex obj) {

*return* *new* Complex(real \* obj.real - imag \* obj.imag, real \* obj.imag + imag \* obj.real);

    }

    public Complex divide(Complex obj) {

        double denominator = obj.real \* obj.real + obj.imag \* obj.imag;

*return* *new* Complex((real \* obj.real + imag \* obj.imag) / denominator,

                (imag \* obj.real - real \* obj.imag) / denominator);

    }

}

class Fraction implements Arithmetical<Fraction> {

    int numerator, denominator;

    Fraction(int n, int d) {

        numerator = n;

        denominator = d;

    }

    public Fraction add(Fraction obj) {

        int n = numerator \* obj.denominator + obj.numerator \* denominator;

        int d = denominator \* obj.denominator;

*return* *new* Fraction(n, d);

    }

    public Fraction subtract(Fraction obj) {

        int n = numerator \* obj.denominator - obj.numerator \* denominator;

        int d = denominator \* obj.denominator;

*return* *new* Fraction(n, d);

    }

    public Fraction multiply(Fraction obj) {

        int n = numerator \* obj.numerator;

        int d = denominator \* obj.denominator;

*return* *new* Fraction(n, d);

    }

    public Fraction divide(Fraction obj) {

        int n = numerator \* obj.denominator;

        int d = denominator \* obj.numerator;

*return* *new* Fraction(n, d);

    }

}

public class MyGenericProgram {

    public static void main(String[] args) {

        Complex c1 = *new* Complex(1, 2);

        Complex c2 = *new* Complex(3, 4);

        Complex c3 = c1.add(c2);

        Complex c4 = c1.subtract(c2);

        Complex c5 = c1.multiply(c2);

        Complex c6 = c1.divide(c2);

        System.out.println("Complex addition: " + c3.real + " + " + c3.imag + "i");

        System.out.println("Complex subtraction: " + c4.real + " + " + c4.imag + "i");

        System.out.println("Complex multiplication: " + c5.real + " + " + c5.imag + "i");

        System.out.println("Complex division: " + c6.real + " + " + c6.imag + "i");

        Fraction f1 = *new* Fraction(1, 2);

        Fraction f2 = *new* Fraction(3, 4);

        Fraction f3 = f1.add(f2);

        Fraction f4 = f1.subtract(f2);

        Fraction f5 = f1.multiply(f2);

        Fraction f6 = f1.divide(f2);

        System.out.println("Fraction addition: " + f3.numerator + "/" + f3.denominator);

        System.out.println("Fraction subtraction: " + f4.numerator + "/" + f4.denominator);

        System.out.println("Fraction multiplication: " + f5.numerator + "/" + f5.denominator);

        System.out.println("Fraction division: " + f6.numerator + "/" + f6.denominator);

    }

}

//把你编写（并运行正常）的第**4**小题的Java程序复制粘贴在下面

import java.util.HashMap;

import java.util.Map;

public class HashMapTest {

  public HashMapTest() {

*// Create a HashMap to store Student objects*

    Map<String, Student> students = *new* HashMap<>();

*// Add students to the HashMap*

    students.put("17H002", *new* Student("17H002", "孙悟空", 2000));

    students.put("17H001", *new* Student("17H001", "唐僧", 40));

    students.put("17H003", *new* Student("17H003", "猪八戒", 1000));

    students.put("17H004", *new* Student("17H004", "沙僧", 900));

*// Output information of the student with id "17H003"*

    Student student17H003 = students.get("17H003");

    System.out.println("Information of the student with id 17H003:");

    System.out.println(student17H003);

*// Output information of all students*

    System.out.println("Information of all students:");

*for* (Student student *:* students.values()) {

      System.out.println(student);

    }

*// Remove "唐僧"*

    students.remove("17H001");

*// Convert HashMap to an array of type Student*

    Student[] studentArray = students.values().toArray(*new* Student[0]);

*// Print out all elements of the array using a for-each loop*

    System.out.println("Elements of the Student array:");

*for* (Student student *:* studentArray) {

      System.out.println(student);

    }

  }

  public static void main(String[] args) {

*// Create a HashMapTest object to call the no-argument constructor*

*new* HashMapTest();

  }

}

class Student {

  private String id;

  private String name;

  private int age;

  public Student(String id, String name, int age) {

*this*.id = id;

*this*.name = name;

*this*.age = age;

  }

  public Student() {

*// No-argument constructor*

  }

  public String getName() {

*return* name;

  }

  @Override

  public String toString() {

*return* "Student{" +

        "id='" + id + '\'' +

        ", name='" + name + '\'' +

        ", age=" + age +

        '}';

  }

}

**/\* 如果某题一直没能调试通过，请在下面补充说明 \*/**

**题号：**

**源程序代码：**

**编译（或运行）出现的问题描述：**

**- Experiment 9 -**

//把你编写（运行正常或不正常）的GUI Java程序复制粘贴在下面

**/\* 如果程序不正常，请在下面说明 \*/**

**编译（或运行）出现的问题描述：**

import java.awt.*\**;

import java.awt.event.*\**;

import javax.swing.*\**;

*/\*\**

*\* TODO: 填充文字w*

*\*/*

public class Gui {

  private JFrame frame;

  private JMenuBar menuBar;

  private JMenu menu1, menu2, menu3;

  private JMenuItem menuItem1, menuItem2, menuItem3, menuItem4, menuItem5, menuItem6, menuItem7, menuItem8, menuItem9;

  private JTextField textField;

  public Gui() {

    frame = *new* JFrame("Experiment9");

    frame.setSize(600, 500);

    frame.setLocation(100, 100);

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

    menuBar = *new* JMenuBar();

    menu1 = *new* JMenu("成长的我");

    menu2 = *new* JMenu("温馨一刻");

    menu3 = *new* JMenu("关于我");

    menuItem1 = *new* JMenuItem("我的小学");

    menuItem2 = *new* JMenuItem("我的初中");

    menuItem3 = *new* JMenuItem("我的高中");

    menuItem4 = *new* JMenuItem("我的大学");

    menuItem5 = *new* JMenuItem("儿时玩伴");

    menuItem6 = *new* JMenuItem("青梅竹马");

    menuItem7 = *new* JMenuItem("大学室友");

    menuItem9 = *new* JMenuItem("关于我");

    menuItem8 = *new* JMenuItem("我的爱好");

    menu1.add(menuItem1);

    menu1.add(menuItem2);

    menu1.add(menuItem3);

    menu1.add(menuItem4);

    menu2.add(menuItem5);

    menu2.add(menuItem6);

    menu2.add(menuItem7);

    menuBar.add(menu1);

    menuBar.add(menu2);

    menuBar.add(menu3);

    textField = *new* JTextField(20);

    textField.setEditable(false);

    frame.setJMenuBar(menuBar);

    frame.getContentPane().add(textField, BorderLayout.CENTER);

    frame.pack();

    frame.setVisible(true);

    menuItem1.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame schoolFrame = *new* JFrame("我的小学");

          JTextField schoolField = *new* JTextField("XXXX");

          schoolField.setEditable(false);

          schoolFrame.getContentPane().add(schoolField, BorderLayout.CENTER);

          schoolFrame.pack();

          schoolFrame.setVisible(true);

        }

      }

    );

    menuItem2.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame schoolFrame = *new* JFrame("我的初中");

          JTextField schoolField = *new* JTextField("YYYY");

          schoolField.setEditable(false);

          schoolFrame.getContentPane().add(schoolField, BorderLayout.CENTER);

          schoolFrame.pack();

          schoolFrame.setVisible(true);

        }

      }

    );

    menuItem3.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame schoolFrame = *new* JFrame("我的高中");

          JTextField schoolField = *new* JTextField("ZZZZ");

          schoolField.setEditable(false);

          schoolFrame.getContentPane().add(schoolField, BorderLayout.CENTER);

          schoolFrame.pack();

          schoolFrame.setVisible(true);

        }

      }

    );

    menuItem4.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame schoolFrame = *new* JFrame("我的大学");

          JTextField schoolField = *new* JTextField("CJLU");

          schoolField.setEditable(false);

          schoolFrame.getContentPane().add(schoolField, BorderLayout.CENTER);

          schoolFrame.pack();

          schoolFrame.setVisible(true);

        }

      }

    );

    menuItem5.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame photoFrame = *new* JFrame("儿时玩伴");

          JLabel photoLabel = *new* JLabel(*new* ImageIcon("R.jpg"));

          photoFrame.getContentPane().add(photoLabel, BorderLayout.CENTER);

          photoFrame.pack();

          photoFrame.setVisible(true);

        }

      }

    );

    menuItem6.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame photoFrame = *new* JFrame("青梅竹马");

          JLabel photoLabel = *new* JLabel(*new* ImageIcon("R.jpg"));

          photoFrame.getContentPane().add(photoLabel, BorderLayout.CENTER);

          photoFrame.pack();

          photoFrame.setVisible(true);

        }

      }

    );

    menuItem7.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame photoFrame = *new* JFrame("大学室友");

          JLabel photoLabel = *new* JLabel(*new* ImageIcon("R.jpg"));

          photoFrame.getContentPane().add(photoLabel, BorderLayout.CENTER);

          photoFrame.pack();

          photoFrame.setVisible(true);

        }

      }

    );

    menu3.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JOptionPane.showMessageDialog(

            frame,

            "姓名：张三\n年龄：25\n职业：软件工程师",

            "关于我",

            JOptionPane.INFORMATION\_MESSAGE

          );

        }

      }

    );

    menu3.add(menuItem9);

    menuItem9.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame aboutFrame = *new* JFrame("关于我");

          JTextField textField;

          textField = *new* JTextField("About Me Text");

          textField.setEditable(false);

          aboutFrame.getContentPane().add(textField, BorderLayout.CENTER);

          aboutFrame.setSize(500, 500);

          aboutFrame.pack();

          aboutFrame.setVisible(true);

        }

      }

    );

    menu3.add(menuItem8);

    menuItem8.addActionListener(

*new* ActionListener() {

        public void actionPerformed(ActionEvent e) {

          JFrame hobbyFrame = *new* JFrame("我的爱好");

          JComboBox<String> comboBox;

          JTextField textField;

*// 创建下拉选框*

          comboBox = *new* JComboBox<String>();

          comboBox.addItem("阅读技术书籍");

          comboBox.addItem("编写代码");

          comboBox.addItem("参加编程竞赛");

          comboBox.addItem("学习新技术");

          comboBox.addItem("开发自己的应用程序");

          textField = *new* JTextField("Pls choose an option...");

          textField.setEditable(false);

          comboBox.addActionListener(

*new* ActionListener() {

              public void actionPerformed(ActionEvent e) {

*if* (

                  comboBox.getSelectedItem().toString().equals("学习新技术")

                ) {

                  textField.setText("恭喜！你回答正确了！!");

                }

              }

            }

          );

          hobbyFrame.getContentPane().add(comboBox, BorderLayout.NORTH);

          hobbyFrame.getContentPane().add(textField, BorderLayout.CENTER);

          hobbyFrame.setSize(500, 500);

          hobbyFrame.pack();

          hobbyFrame.setVisible(true);

        }

      }

    );

  }

  public static void main(String[] args) {

    Gui gui = *new* Gui();

  }

}