

NumberedTicketGenerator.java

```
package j2.exam.ex01;

public abstract class NumberedTicketGenerator {
    protected int serialNo;

    public NumberedTicketGenerator() {
        super();
        this.serialNo = 1000;
    }

    public void setSerialNo(int initNo){
        this.serialNo = initNo;
    }

    public void doPrint(int num) {
        for (int i = 0; i < num; i++) {
            ;
            ;
            ;
            ;
        }
    }

    protected void printSeparator() {
        System.out.println("-----");
    }

    protected void printSerial() {
        System.out.println("> NO." + this.serialNo);
        this.serialNo++;
    }

    protected abstract void printHeader();
    protected abstract void printTrailer();
}
```

PartyTicketGenerator.java

```
package j2.exam.ex01;

public class PartyTicketGenerator extends NumberedTicketGenerator {
    protected void printHeader() {
        System.out.println("New Year Party");
    }

    protected void printTrailer() {
        System.out.println("Fee 1000 yen");
    }
}
```

Ex01.java

```
package j2.exam.ex01;

public class Ex01 {
    public static void main(String[] args) {
        NumberedTicketGenerator ticketGenerator = ;
        ticketGenerator.setSerialNo();
        ticketGenerator.doPrint();
    }
}
```

Shape.java

```
package j2.exam.ex02;
public abstract class Shape {
    abstract double getArea();
    abstract double getCircumference();
}
```

Triangle.java

```
package j2.exam.ex02;
public class Triangle extends Shape {
    private double a, b, c;

    public Triangle(double a, double b, double c){
        (1).a = a;
        (1).b = b;
        (1).c = c;
    }

    public double getArea(){
        double s = (a + b + c)/2.0;
        return Math.sqrt(s*(s-a)*(s-b)*(s-c));
    }

    public double getCircumference(){
        return a + b + c;
    }
}
```

Parallelogram.java

```
package j2.exam.ex02;

public class Parallelogram extends Shape{
    // 2辺 a,b と挟む角 angle で作られる平行四辺形
    private double a;
    private double b;
    private double angle; // should be in radian

    public Parallelogram(double a, double b, double angle){
        (1).a = a;
        (1).b = b;
        (1).angle = angle;
    }

    public double getArea(){
        return a*b*Math.sin(angle);
    }

    public double getCircumference(){
        return (2);
    }
}
```

Cylinder.java

```
package j2.exam.ex02;

public class Cylinder {
    private Shape base;
    private double height;

    public Cylinder(Shape base, double height){
        this.base = base;
        this.height = height;
    }

    public double getVolume(){
        return base.getArea() * height;
    }

    public double getSurfaceArea(){
        return ;
    }
}
```

Ex02.java

```
package j2.exam.ex02;

public class Ex02 {
    public static void main(String[] args) {
        Shape t = ;
        Shape p = ;
        Cylinder ct = new Cylinder(t, 2.0);
        Cylinder pt = new Cylinder(p, 3.0);
        System.out.println("ct' volume -> " + ct.getVolume());
        System.out.println("ct' surface -> " + ct.getSurfaceArea());
        System.out.println("pt' volume -> " + pt.getVolume());
        System.out.println("pt' surface -> " + pt.getSurfaceArea());
    }
}
```

Ex03.java

```
package j2.exam.ex03;

import java.io.*;
import java.util.*;

public class Ex03 {
    static int numberOfFormatErrors = 0;
    static int sum = 0;

    public static void main(String[] args) {
        try {
            Collection data = drain();
            printData(data);
        } catch (IOException e) {
            System.out.println(e);
            return;
        }

        System.out.println("非整数データの個数 -> "
            + numberOfFormatErrors);
        System.out.println("合計 = " + sum);
    }

    // 次ページに続く
```

```

private static Collection drain() throws IOException {
    BufferedReader reader = new BufferedReader(new FileReader(
        "C:/temporary/data.txt"));
    Collection input = new LinkedList();
    while (true) {
        String line = reader.readLine();
        if (line.equals("end")) {
            // while ループを終了
            break;
        } else if (line.equals("IO error")) {
            // ここで人為的な IO エラーを起こす
            IOException e = new IOException("人為的 IO エラー");
            throw e;
        } else {
            // 入力データの処理
            try {
                int x = Integer.parseInt(line);
                sum += x;
                input.add(line);
            } catch (NumberFormatException e) {
                numberOfFormatErrors++;
            }
        }
    }
    return input;
}

private static void printData(Collection list)
throws IOException {
    Iterator it = list.iterator();
    while (it.hasNext()) {
        String str = (String) it.next();
        System.out.println(str);
    }
}
}

```

ButtonListenerA.java

```
package j2.exam.ex04;
import java.awt.event.*;
public class ButtonListenerA implements ActionListener {
    private final Ex04 c;
    public ButtonListenerA(Ex04 c) { this.c = c; }
    public void actionPerformed(ActionEvent e) {
        ;
    }
}
```

ButtonListener.java

```
package j2.exam.ex04;
import java.awt.event.*;
public class ButtonListenerB implements ActionListener {
    private final Ex04 c;
    public ButtonListenerB(Ex04 c) { this.c = c; }
    public void actionPerformed(ActionEvent e) {
        ;
    }
}
```

ButtonListenerC.java

```
package j2.exam.ex04;
import java.awt.event.*;
public class ButtonListenerC implements ActionListener{
    private final Ex04 c;

    public ButtonListenerC(Ex04 c) {this.c = c;}
    public void actionPerformed(ActionEvent e) {
        c.reset();
    }
}
```

Ex04.java

```
package j2.exam.ex04;

import java.awt.*;
import javax.swing.*;

public class Ex04 {
    private int count;
    private JFrame frame;
    private JLabel label;
    protected void init() {
        this.count = 0;
        this.frame = new JFrame("Counter");
        this.frame.setDefaultCloseOperation(
            JFrame.DISPOSE_ON_CLOSE);
        this.label = new JLabel("0");
        this.label.setHorizontalAlignment(JLabel.CENTER);
        JPanel panel = new JPanel(new GridLayout(1, 3));
        JButton b1 = new JButton("1+");
        JButton b2 = new JButton("10+");
        JButton b3 = new JButton("reset");
        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        this.frame.getContentPane().setLayout(new BorderLayout());
        this.frame.getContentPane().add(
            this.label, BorderLayout.CENTER);
        this.frame.getContentPane().add(panel, BorderLayout.SOUTH);
        this.frame.setSize(200, 100);
        this.frame.setVisible(true);
        b1.addActionListener(new ButtonListenerA( (3) ));
        b2.addActionListener(new ButtonListenerB( (3) ));
        b3.addActionListener(new ButtonListenerC( (3) ));
    }
}

// 次ページに続く
```

```
void incByOne() {
    this.count += 1;
    this.label.setText(String.valueOf(this.count));
}

void incByTen() {
    this.count += 10;
    this.label.setText(String.valueOf(this.count));
}

void reset(){
    this.count = 0;
    this.label.setText(String.valueOf(this.count));
}

public static void main(String[] args) {
    Ex04 main = new Ex04();
    main.init();
}
}
```

ExamRecord.java

```
package j2.exam.ex05;

public class ExamRecord {
    private String name; // Name

    private int ma; // mathematics
    private int jp; // japanese
    private int en; // english

    // コンストラクタ
    public ExamRecord(String name, int ma, int jp, int en) {
        this.name = name;
        this.ma = ma;
        this.jp = jp;
        this.en = en;
    }

    // 名前を返す
    public String getName() {
        return this.name;
    }

    // 3科目(数学,国語,英語)の合計を返す
    public int getTotal() {
        return this.ma + this.jp + this.en;
    }

    // 3科目(数学,国語,英語)の成績を表示する
    public void show() {
        System.out.println(name + ", ma=" + ma + ", jp=" + jp + ", "
            + " en=" + en + ", total=" + getTotal());
    }
}
```

```

package j2.exam.ex05;
import java.util.*;

public class Ex05 {
    public static void main(String[] args) {
        // 長さ 5 の ExamRecord の配列を生成
        ExamRecord[] rec = ;
        rec[0] = new ExamRecord("Taro", 50, 90, 100);
        rec[1] = new ExamRecord("Hanako", 70, 30, 80);
        rec[2] = new ExamRecord("Kenji", 60, 50, 70);
        rec[3] = new ExamRecord("Aya", 40, 100, 90);
        rec[4] = new ExamRecord("Mai", 80, 80, 80);

        // 全レコードの表示
        for (int i = 0; i < rec.length; i++) {
            ;
        }

        // 総合点の最高得点と最高得点者を求める
        int max = -1; // 最高得点
        Collection idxs = new LinkedList(); // 最高得点者の index のリスト

        for (int i = 0; i < rec.length; i++){
            int total = rec[i].getTotal();
            if(total > max){// ここまでの最大を超えていたら
                max = total;
                ; // 最高得点者のインデックスリストを空にして
                idxs.add(new Integer(i)); // i をコレクションに追加
            } else if (total == max) {
                idxs.add(new Integer(i)); // i をコレクションに追加
            } else {
                // do nothing
            }
        }
    }
}
// 次ページに続く

```

```
System.out.println("総合点の最高得点 = " + max );
System.out.println("総合点の最高得点者一覧");

Iterator it = idxs.iterator();
while(it.hasNext()){
    int idx = Integer.valueOf((Integer)it.next());
    System.out.println( [ (4) ] );
}
}
```