

計算機ネットワーク II (Java 編) ・ テスト問題用紙

('09年2月13日 ・ 10:30 ~ 12:00)

解答上、その他の注意事項

- I. 問題は、問 I ~ IV までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. 解答欄を間違えないよう注意すること。
- IV. 解答中の文字 (特に a と d) がはっきりと区別できるよう注意すること。
- V. 持ち込みは不可である。

すべての問に対する補足:

プログラムの空欄を埋める問題では、解答が長くなる可能性があるので、下の省略形 (○囲み文字) を用いても良い。例えば `this==null` と書く代わりに、 $\textcircled{T}==\textcircled{N}$ と書いて良い。(必ず○に囲んで書くこと。)

\textcircled{A} ActionListener \textcircled{C} class \textcircled{D} actionPerformed \textcircled{G} getSource
 \textcircled{I} implements \textcircled{K} KeyListener \textcircled{J} JApplet \textcircled{L} addMouseListener
 \textcircled{M} MouseListener \textcircled{N} null \textcircled{P} public \textcircled{Q} equals \textcircled{R} Runnable
 \textcircled{S} System.out.println \textcircled{T} this \textcircled{V} private \textcircled{W} new \textcircled{X} extends

また、参考のために問題用紙の末尾に授業配布プリントの `Tree.java`, `MouseTest.java` (計算機ネットワーク I), `BubbleSort1.java`, `BubbleSort2.java`, `Point.java`, `ColorPoint.java` のソースを掲載する。

- I. 下のプログラムは、8017番ポートで接続を待ち受け、クライアントと接続したら「本日の名言」(quote of the day)を送信する(だけ)のサーバプログラムである。次の空欄(i)~(ii)を埋めて、プログラムを完成させよ。

ファイル: Qotd.java

```
import java.net.*;
import java.io.*;

public class Qotd {
    static final String[] quotes = {
        "Gravitation is not responsible for people falling in love."
        + " -- Albert Einstein",
        "The best way to predict the future is to invent it. -- Alan Key",
        "In mathematics, you don't understand things."
        + " You just get used to them. -- John von Neumann",
        "人苦不知足、既平臙、復望蜀。 -- 劉秀(光武帝)"
    };

    public static void main(String args[]) {
        try {
            (i) servsock = new ServerSocket((ii));
            while(true){
                Socket sock = servsock.accept();
                PrintStream out = new PrintStream(sock.getOutputStream());
                long now = System.currentTimeMillis(); // 現在の時刻に応じて
                int len = quotes.length;
                out.println(quotes[(int)(now%len)]); // 名言を表示
                sock.close();
            }
        } catch (Exception e){
            System.exit(1);
        }
    }
}
```

(参考) ファイル: TCP_RO.java (クライアント側のプログラム)

— 授業配布プリント 第A章 例題 A.1.1 と同じ

```
import java.net.*;
import java.io.*;

public class TCP_RO {
    public static void main(String[] argv) {
        try {
            Socket readSocket = new Socket(argv[0], Integer.parseInt(argv[1]));
            InputStream instrm = readSocket.getInputStream();
            while(true) {
                int c = instrm.read();
                if (c==-1) break;
                System.out.write(c);
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

例えば、このサーバプログラムを IP アドレス 192.168.0.1 のマシンで

```
java Qotd
```

というコマンドで起動する。同時に、クライアント側プログラムを同一または別のマシンで、

```
java TCP_R0 192.168.0.1 8017
```

というコマンドで起動すると、次のようなテキストファイルを表示する。

(アクセス時刻により内容は異なる。)

```
The best way to predict the future is to invent it. -- Alan Key
```

II. 次のプログラムは、標準入力から行で区切られた複数の整数を受け取り、それをグラフにして出力する、というJavaプログラムである。動作例は以下の通りである。(斜字体は入力を表す。↵は改行を入力したことを表す。)

```

12↵
3↵
23↵ ( この改行のあとに Windows の場合 Ctrl-C を入力 )
*****
***
*****

```

入力された整数を保存するために総称クラスの ArrayList を用いる。

(i) の空欄 (2箇所の内容は共通) を埋めよ

ファイル: CharGraph.java

```

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

public class CharGraph {
    public static void main(String[] args) {
        BufferedReader reader
            = new BufferedReader(new InputStreamReader(System.in));
        (i) ns = new (i) ();

        String line;
        try {
            while ((line=reader.readLine())!=null) {
                try {
                    int n = Integer.parseInt(line);
                    ns.add(n);
                } catch(NumberFormatException e) {}
            }
        } catch (IOException e) {}

        for (int n : ns) {
            int i;
            for (i=0; i<n; i++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

なお、Java のプリミティブ型とラッパークラスとの対応を以下に挙げる。

プリミティブ型	ラッパークラス
int	Integer
char	Character
double	Double
boolean	Boolean

III. 次に定義されるクラス NumD を継承して、

ファイル: NumD.java

```
public class NumD {
    public int num;

    public void print() { // 通常の 10 進数で表示
        System.out.printf("%d", num);
    }

    public void increment() {
        num++;
    }

    public void incrementAfterPrint() {
        print();
        System.out.printf(" ");
        increment();
    }
}
```

3 つのクラス NumR, NumX, NumY を定義する。

ファイル: NumR.java

```
public class NumR  {
    private static String[] roman = { "I", "II", "III", "IV", "V", "VI",
        "VII", "VIII", "IX", "X", "XI", "XII" };
    @Override
    public void print() { // ローマ数字で表示
        if (num<=12) {
            System.out.print(roman[num-1]);
        } else {
            System.out.print("??");
        }
    }
}
```

ファイル: NumX.java

```
public class NumX  { // 01
    public boolean flag; // 02
    // 03
    public NumX(boolean b) { // 04
        super(); // 05
        flag = b; // 06
    } // 07
    // 08
    @Override // 09
    public void print() { // 10 進数で表示
        if (flag) { // A~F は大文字で表示 // 11
            System.out.printf("%X", num); // 12
        } else { // a~f は小文字で表示 // 13
            System.out.printf("%x", num); // 14
        } // 15
    } // 16
} // 17
```

ファイル: NumY.java

```
public class NumY ◆ {
    private static String[] kazu = { "ひい", "ふう", "みい", "よう", "いつ",
        "むう", "なな", "やあ", "この", "とお" };

    @Override
    public void print() { // やまとことばで表示
        if (num<=10) {
            System.out.print(kazu[num-1]);
        } else {
            System.out.print("たくさん");
        }
    }
}
```

また、NumTest クラスはこれらのクラスのテスト用の main メソッドを持つ。

ファイル: NumTest.java

```
public class NumTest { // 01
    public static void main(String[] args) { // 02
        NumD[] ns = new NumD[3]; // 03
        ns[0] = new NumR(); // 04
        NumX x = new NumX(true); // 05
        x.flag = false; // 06
        ns[1] = x; // 07
        ns[2] = new NumY(); // 08
        // 各クラスの違いが出るように、値を 9 にセットする // 09
        ns[0].num = 9; ns[1].num = 9; ns[2].num = 9; // 10
        // 11
        int i, j; // 12
        for (i=0; i<4; i++) { // 13
            for (j=0; j<ns.length; j++) { // 14
                ns[j].incrementAfterPrint(); // 15
            } // 16
            System.out.println(" "); // 改行する // 17
        } // 18
    } // 19
} // 20
```

- (i) ◆ の空白 (3 箇所共通) を埋めて、クラスの定義を完成させよ。
- (ii) NumX クラスのフィールド flag の値はコンストラクタでのみ与えることができるものとし、いったん作成した後は、クラスの外からは直接アクセスできないようにしたい。(例えば、NumTest クラスの 6 行目はコンパイル時にエラーになるようにしたい。) NumX クラスの定義の何行めをどのように変更すれば良いか? (プログラム中の行の末尾の数字がクラスの中での行数を表す。)
- (iii) NumTest クラスの 6 行目をコメントアウトし、このクラスの main メソッドを実行するとき、出力はどうなるか?

- IV. 下のプログラムは、“ ”がある向きに移動し、“ ”が右端から消えると、再び左端から現れ、“ ”が下端から消えると上端から現れる Java アプレットである。また、マウスがクリックされると“ ”はクリックされた位置に“瞬間移動”する。
(改ページの都合でソースファイルをいくつか分割して示す。)

ファイル: BallAnimation.java

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

public class BallAnimation  {
```

```
    Thread ballThread = null;
    int x = 100, dx = 10, y = 100, dy = 16;
```

```
    @Override
    public void init() {
        
    }
}
```

```
    @Override
    public void start() {
        if (ballThread==null) {
            
            ballThread.start();
        }
    }
}
```

```
    @Override
    public void stop() {
        ballThread = null;
    }
}
```

```
    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.drawString(" ", x, y);
    }
}
```

```
    public void run() {
        Thread thisThread = Thread.currentThread();
        while (ballThread==thisThread) {
            x += dx; y += dy;
            x %= 200; y %= 200;
            repaint();
            try {
                Thread.sleep(300);
            } catch (InterruptedException e){}
        }
    }
}
```

```
public void mouseClicked(MouseEvent e) {
    x = e.getX();
    y = e.getY();
}
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}
public void mousePressed(MouseEvent e) {}
public void mouseReleased(MouseEvent e) {}
}
```

(i) ~ (iii) の空欄を埋めてプログラムを完成させよ。

以下に参考のために授業配布プリントの `Tree.java`, `MouseTest.java` (計算機ネットワーク I), `BubbleSort1.java`, `BubbleSort2.java`, `Point.java`, `ColorPoint.java` のソースを掲載する。

Tree.java

```
import java.awt.*;
import javax.swing.*;
import java.util.ArrayList;
import static java.lang.Math.*;

public class Tree extends JApplet {
    ArrayList<int[]> data = new ArrayList<int[]>();

    public void drawTree(int d, double x, double y, double r, double t) {
        if (d==0) return;
        data.add(new int[] {(int)x, (int)y, (int)(x+r*cos(t)), (int)(y+r*sin(t))});
        drawTree(d-1, x+r*cos(t), y+r*sin(t), 0.5*r, t+0.2);
        drawTree(d-1, x+(0.55*r)*cos(t), y+(0.55*r)*sin(t), 0.5*r, t+1.25);
        drawTree(d-1, x+(0.45*r)*cos(t), y+(0.45*r)*sin(t), 0.5*r, t-1.3);
    }

    @Override
    public void init() {
        drawTree(6, 128, 255, 128, -PI/2);
    }

    @Override
    public void paint(Graphics g) {
        for(int[] pts : data) {
            g.drawLine(pts[0], pts[1], pts[2], pts[3]);
        }
    }
}
```

MouseTest.java

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

public class MouseTest extends JApplet implements MouseListener {
    int x=50, y=20;

    @Override
    public void init() {
        addMouseListener(this);
    }

    public void mouseClicked(MouseEvent e) {
        x = e.getX(); y = e.getY();
        repaint();
        return;
    }

    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

    @Override
    public void paint(Graphics g) {
        super.paint(g); // 背景を描画
        g.drawString("HELLO WORLD!", x, y);
    }
}
```

BubbleSort1.java

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

public class BubbleSort1 extends JApplet implements Runnable {
    int[] args = {10, 3, 46, 7, 23, 34, 8, 12, 4, 45, 44, 52};
    Color[] cs = {Color.RED, Color.ORANGE, Color.GREEN, Color.BLUE};
    Thread thread=null;

    @Override
    public void start() {
        if (thread == null) {
            thread = new Thread(this);
            thread.start();
        }
    }

    @Override
    public void stop() {
        thread = null;
    }

    @Override
    public void paint(Graphics g) {
        int i;
        super.paint(g);
        for(i=0; i<args.length; i++) {
            g.setColor(cs[args[i]%cs.length]);
            g.fillRect(0, i*10, args[i]*5, 10);
        }
    }

    public void run() {
        int i, j;
        Thread thisThread = Thread.currentThread();
        for (i=0; i<args.length-1; i++) {
            for (j=args.length-1; thread == thisThread && j>i; j--) {
                if (args[j-1]>args[j]) { // スワップする。
                    int tmp=args[j-1]; args[j-1]=args[j]; args[j]=tmp;
                }
                repaint();
                try { // repaint の後でしばらく止まる
                    Thread.sleep(500);
                } catch (InterruptedException e) {}
            }
        }
    }
}
```

BubbleSort2.java

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

public class BubbleSort2 extends JApplet implements Runnable, ActionListener {
    int[] args = { 10, 3, 46, 7, 23, 34, 8, 12, 4, 45, 44, 52};
    Color[] cs = {Color.RED, Color.ORANGE, Color.GREEN, Color.BLUE};
    Thread thread=null;
    private boolean threadSuspended=true;

    @Override
    public void init() {
        JButton step = new JButton("Step");
        step.addActionListener(this);
        setLayout(new FlowLayout());
        add(step);
    }

    // start, stop, paint メソッドは BubbleSort1.java と同一なので省略する。

    public synchronized void actionPerformed(ActionEvent e) {
        threadSuspended=false;
        notify();
    }

    public void run() {
        int i, j;

        for (i=0; i<args.length-1; i++) {
            for (j=args.length-1; j>i; j--) {
                if (args[j-1]>args[j]) { // スワップする。
                    int tmp=args[j-1]; args[j-1]=args[j]; args[j]=tmp;
                }
                repaint();
                try { // repaint の後で止まる
                    synchronized(this) {
                        while (threadSuspended) {
                            wait();
                        }
                        threadSuspended=true;
                    }
                } catch (InterruptedException e) {}
            }
        }
    }
}
```

Point.java

```
public class Point {
    public int x, y;

    public void move(int dx, int dy) {
        x += dx; y += dy;
    }

    public void print() {
        System.out.printf("(%d, %d)", x, y);
    }

    public void moveAndPrint(int dx, int dy) {
        print(); move(dx, dy); print();
    }

    public Point(int x0, int y0) {
        x = x0; y = y0;
    }
}
```

ColorPoint.java

```
public class ColorPoint extends Point {
    private String[] cs = {"black", "red", "green", "yellow",
                          "blue", "magenta", "cyan", "white"};
    private int color; // 0-黒 1-赤 2-緑 3-黄 4-青 5-紫 6-水 7-白

    @Override
    public void print() {
        System.out.printf("<font color='%s'>", getColor()); // 色の指定
        super.print();
        System.out.print("</font>"); // 色を戻す
    }

    public void setColor(String c) {
        int i;
        for (i=0; i<cs.length; i++) {
            if (c.equals(cs[i])) {
                color = i; return;
            }
        }
        // 対応する色がなかったら何もしない。
    }

    public ColorPoint(int x, int y, String c) {
        super(x, y);
        setColor(c);
    }

    public String getColor() {
        return cs[color];
    }
}
```

計算機ネットワーク II (Java 編) ・ テスト解答用紙 ('09 年 2 月 13 日)

学籍番号		氏名	
------	--	----	--

I. (4×2)

(i).		(ii).	
------	--	-------	--

II. (4)

(i).	
------	--

III. (5×3)

(i).	
(ii).	
(iii).	

IV. (5, 4, 4)

(i).	
(ii).	
(iii).	

裏面に感想を記入する欄があります。

