

# 計算機ネットワーク II ( Java 編 ) ・ テスト問題用紙

( '08 年 2 月 15 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。

全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、**(T)**==**(N)**と書いて良い。

- (A) ActionListener   (C) class   (D) actionPerformed   (G) getSource  
(I) implements   (J) JApplet   (L) addActionListener   (M) Math   (N) null   (P) public  
(Q) equals   (R) Runnable   (S) System.out.println   (T) this   (V) private   (W) new  
(X) extends

参考のために、BubbleSort1.java, BubbleSort2.java, Point.java, ColorPoint.javaのソースを掲載する。

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() {
        Thread thisThread = Thread.currentThread();
        while(true) {
            int i, j;
            for (i=0; i<args.length-1; i++) {
                for (j=args.length-1; j>i && thread == thisThread; 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() {
        while(true) {
            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];
    }
}
```

- I. 下のプログラムは、8080 番ポートで接続を待ち受け、接続したら、送られてくるデータの 2 行を読み飛ばして、相手のホスト名 (あるいは IP アドレス) を表示する HTML ページを送信する (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: EchoHostname.java

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

public class EchoHostname {
    public static void main(String args[]) {
        try {
            ServerSocket servsock = new (i) (ii);
            while(true){
                Socket sock = servsock.accept();
                BufferedReader in = new BufferedReader(
                    new InputStreamReader(sock.getInputStream()));
                PrintStream out = new PrintStream(sock.getOutputStream());
                int i;
                for(i=0; i<2; i++) {
                    in.readLine();
                }
                out.println("<html>");
                out.println("<head><title>Test</title></head><body>");
                out.println("From " + (sock.getInetAddress()).getHostName());
                out.println("</body></html>");
                sock.close() ;
            }
        } catch (Exception e){
            e.printStackTrace();
            System.exit(1) ;
        }
    }
}
```

例えば、このプログラムを

```
java EchoHostname
```

というコマンドで起動して、ブラウザから

```
http://localhost:8080/
```

という URL でアクセスすると、次のような HTML ファイルを送信する。(斜字体の部分は環境により異なる。)

```
<html>
<head><title>Test</title></head><body>
From localhost.localdomain
</body></html>
```

II. 次のプログラムは、キー入力を受け取り、“H”キーが押されれば左、“J”キーが押されれば下、“K”キーが押されれば上、“L”キーが押されれば右に、10ピクセル分の線を描画して現在位置を移動する、というJava アプレットである。入力されたキーを保存するために総称クラスのArrayListを用いている。

右下のスクリーンショットは“HJLLKKKHHHHHJJJJJJJJ”というキーが入力されたときの画面である。

(i) の空欄(2箇所の内容は共通)を埋めよ

ファイル: KeyDraw.java

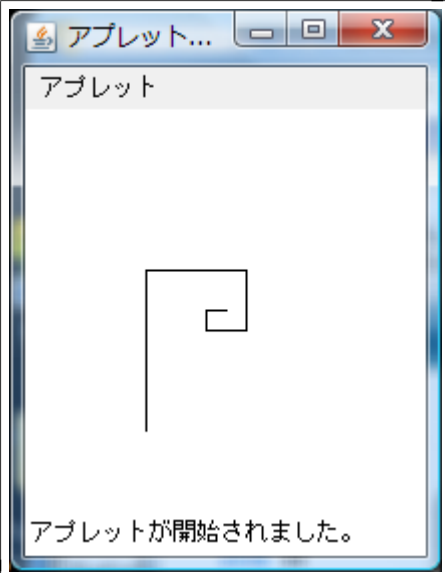
```
import java.util.ArrayList;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class KeyDraw extends JApplet implements KeyListener {
    (i) keys = new (i) ();

    public void keyPressed(KeyEvent e) {
        int k = e.getKeyCode();
        keys.add(k);
        repaint();
    }
    public void keyTyped(KeyEvent e) {}
    public void keyReleased(KeyEvent e) {}

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

    @Override
    public void paint(Graphics g) {
        int i, x0 = 100, y0 = 100;
        for (i=0; i < keys.size(); i++) {
            int x1 = x0, y1 = y0;
            int k = keys.get(i);
            switch (k) {
                case 'H': // left
                    x1 = x0-10; break;
                case 'J': // down
                    y1 = y0+10; break;
                case 'K': // up
                    y1 = y0-10; break;
                case 'L': // right
                    x1 = x0+10; break;
            }
            g.drawLine(x0, y0, x1, y1);
            x0 = x1; y0 = y1;
        }
    }
}
```



III. 次に定義されるクラス Fish を継承して、  
ファイル: Fish.java

```
public class Fish {
    public int age;

    public void grow() {
        age++;
    }

    public void showName() {
        System.out.print("サカナ");
    }

    public void growShowName() {
        showName();
        System.out.print(" ");
        grow();
    }
}
```

3つのクラス Seriola, Mugil, Carp を定義する。

ファイル: Seriola.java

```
public class Seriola  {
    @Override
    public void showName() {
        if (age==0) {
            System.out.print("ツバス");
        } else if (age==1) {
            System.out.print("ハマチ");
        } else {
            System.out.print("ブリ");
        }
    }
}
```

ファイル: Mugil.java

```
public class Mugil  {
    @Override
    public void showName() {
        if (age==0) {
            System.out.print("オボコ");
        } else if (age==1 || age==2) {
            System.out.print("ボラ");
        } else {
            System.out.print("トド");
        }
    }
}
```

ファイル: Carp.java

```
public class Carp  { // 01
    public String color; // 02
    // 03
    public Carp(String c) { color = c; } // 04
    // 05
    @Override // 06
    public void showName() { // 07
        if (age<1000) { // 08
            System.out.print(color+"鯉"); // 09
        } else { // 10
            System.out.print(color+"龍"); // 11
        } // 12
    } // 13
} // 14
```

また、FishTest 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: FishTest.java

```
public class FishTest { // 01
    public static void main(String[] args) { // 02
        Fish[] fish = new Fish[3]; // 03
        fish[0] = new Seriola(); // 04
        fish[1] = new Mugil(); // 05
        Carp c = new Carp("緋"); // 06
        c.color = "緑"; // 07
        fish[2] = c; // 08
    // 09
        int i, j; // 10
        for (i=0; i<3; i++) { // 11
            for (j=0; j<3; j++) { // 12
                fish[j].growShowName(); // 13
            } // 14
            System.out.println(" "); // 15
        } // 16
    } // 17
} // 18
```

- (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) Carp クラスのフィールド color 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、一旦作成した後は、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、FishTest クラスの 7 行目はコンパイル時にエラーになるようにしたい。)   
Carp 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) FishTest クラスの 7 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?(ただし、Fish クラスおよびそのサブクラスのインスタンスが新しく生成されたときのフィールド age の初期値は 0 である。)



- IV. 下のプログラムは、“ ”が左から右に移動し、“ ”が右端から消えると、再び左端から現れるアニメーションを表示する Java アプレットである。このアプレットは「上」「下」の2つのボタン ( JButton ) を持ち、途中で「上」ボタンを押すと、“ ”が 10 ピクセルだけ上に移動し、「下」ボタンを押すと、“ ”が 10 ピクセルだけ下に移動する。  
( 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: SimpleAnimation.java

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

public class SimpleAnimation  (i) {
```

```
    Thread aThread = null;
    int x = 100, dx = 10, y = 100;
    JButton u, d;
```

```
    @Override
    public void init() {
        u = new JButton("上");    d = new JButton("下");
         (ii)
        setLayout(new BorderLayout());
        add(u);    add(d);
    }
```

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

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

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

```
    public void run() {
        Thread thisThread = Thread.currentThread();
        while (aThread==thisThread) {
            x = (x+dx) % 200;
            repaint();
            try {
                Thread.sleep(100);
            } catch (Exception e){}
        }
    }
}
```

```

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == u) {
        y -= 10;
        if (y < 0) {
            y += 200;
        }
    } else if (e.getSource() == d) {
        y = (y+10) % 200;
    }
}
}

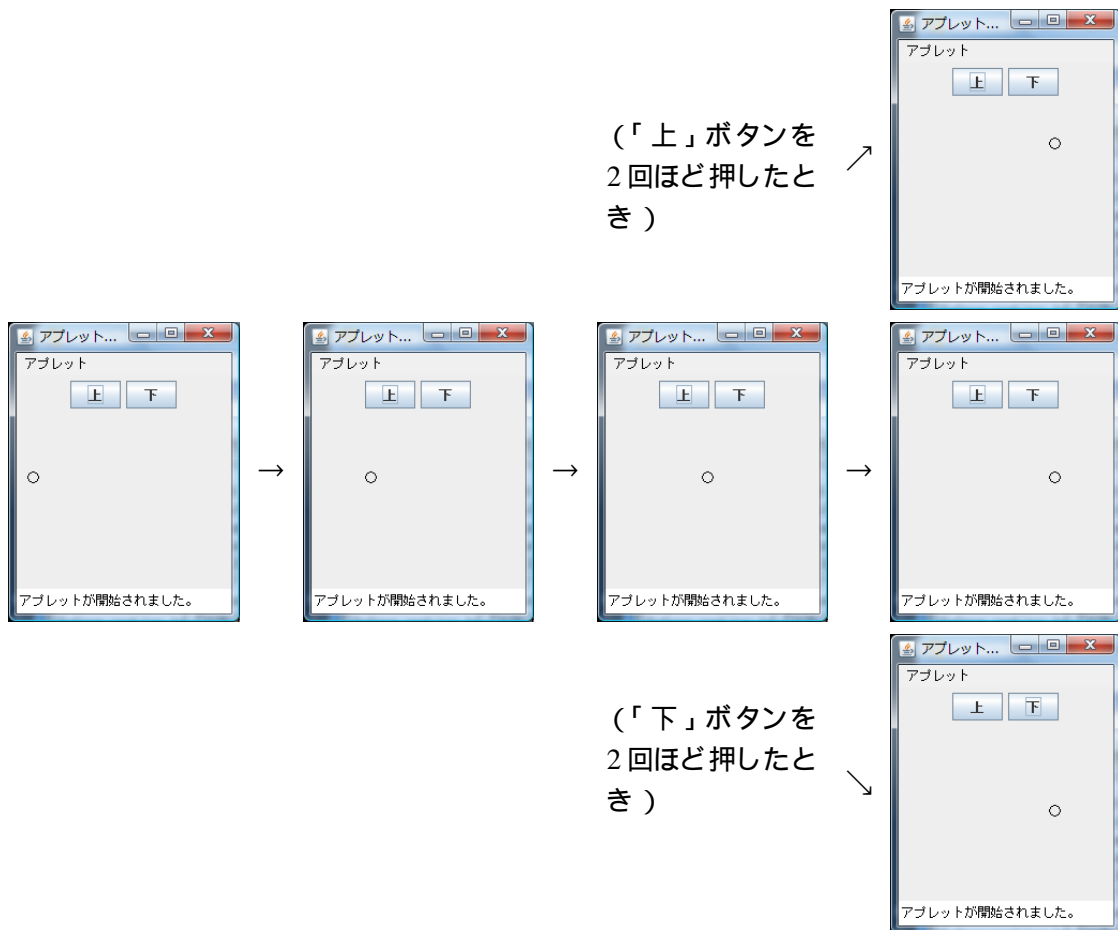
```

```

}

```

以下に実行の様子をスクリーンショットで示す。



(i) ~ (iii) の空欄を埋めてプログラムを完成させよ。

計算機ネットワーク II ( Java 編 ) ・ テスト解答用紙 ( '08 年 2 月 15 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。

授業・テストの感想

---

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

---