

# 計算機ネットワーク II ( Java 編 ) ・ テスト問題用紙

( '10年2月12日 ・ 10:30 ~ 12:00 )

## 解答上、その他の注意事項

- I. 問題は、問 I ~ III 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 `KeyTest.java`, `UpDownButton3.java`, (以上、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、8765 番ポートで接続を待ち受け、クライアントと接続したら、送られてくるデータの2行を読み飛ばして、以前そのホストから何秒前に接続があったかを表示するHTMLページを送信する(だけ)のサーバプログラムである。ホスト毎のアクセス時刻は総称クラスのHashMapを用いて記録している。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: .java

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

public class SimpleStopWatch {
    public static void main(String args[]) {
        // サーバ用ソケットの作成
        ServerSocket servsock = null;
        try {
            servsock = new ServerSocket( (i) );
        } catch (Exception e) {
            e.printStackTrace();
            System.exit(1);
        }

        (ii) map = new (ii) ();
        while(true) {
            try {
                Socket sock = servsock.accept(); // 接続要求の受付
                BufferedReader in = new BufferedReader(
                    new InputStreamReader(sock.getInputStream()));
                PrintStream out = new PrintStream(sock.getOutputStream());
                String host = sock.getInetAddress().getHostName();

                int i ;
                for(i=0; i<2; i++) { // 2行読み飛ばす
                    in.readLine();
                }
                out.println("<html>");
                out.println("<head><title>Test</title></head>");
                out.println("<body>");
                Long prev = map.get(host);
                long now = System.currentTimeMillis(); // このメソッドは、後述
                if (prev==null) {
                    out.printf("%sさん、初めてのアクセスですね。%n", host);
                } else {
                    out.printf("%sさん、%d秒ぶりのアクセスですね。%n",
                        host, (now-prev)/1000);
                }
                out.println("</body>");
                out.println("</html>");
                sock.close();
                map.put(host, now);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

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 SimpleStopWatch
```

というコマンドで起動する。同時に、IP アドレス 192.168.0.2 のマシンの Web ブラウザから `http://192.168.0.1:8765/` という URL でアクセスすると、次のようなページが表示される。(アクセス時刻により内容は異なる。)

1 回目のアクセス

```
192.168.0.2 さん、初めてのアクセスですね。
```

2 回目のアクセス

```
192.168.0.2 さん、12 秒ぶりのアクセスですね。
```

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

また、`currentTimeMillis` メソッドの API ドキュメントは以下の通りである。

```
java.lang
クラス System
public static long currentTimeMillis()

    現在の時間をミリ秒で返します。... (以下、省略)

戻り値:
    ミリ秒で測定した、現在時刻と協定世界時の UTC 1970 年 1 月 1 日午前 0 時
    との差
```

- II. 次に定義されるクラス Creature を継承して、3つのクラス Snake, Frog, Lizard を定義する。  
ファイル: Creature.java

```
public class Creature { // 01
    public int age; // 02
    // 03
    public Creature() { // 注1 // 04
        age = 0; // 05
    } // 06
    // 07
    public void grow() { // 08
        age++; // 09
    } // 10
    // 11
    public String getName() { // 12
        return "謎の生物"; // 13
    } // 14
    // 15
    public int getNumLegs() { // 16
        return 8; // 17
    } // 18
    // 19
    public void selfIntroduce() { // 20
        System.out.printf("私は%sです。脚は%d本です。%n", // 21
            getName(), getNumLegs()); // 22
    } // 23
} // 24
```

ファイル: Snake.java

```
public class Snake (i) {
    public Snake() { // 注1
        super();
    }

    @Override
    public String getName() {
        return "へび";
    }

    @Override
    public int getNumLegs() {
        return 0;
    }
}
```

ファイル: Frog.java

```
public class Frog  {
    public Frog() { // 注1
        super();
    }

    @Override
    public String getName() {
        if (age<=1) {
            return "オタマジャクシ";
        } else {
            return "カエル";
        }
    }

    @Override
    public int getNumLegs() {
        if (age==0) {
            return 0;
        } else if (age==1) {
            return 2;
        } else {
            return 4;
        }
    }
}
```

ファイル: Lizard.java

```
public class Lizard  {
    public Lizard() { // 注1
        super();
    }

    @Override
    public String getName() {
        return "トカゲ";
    }

    @Override
    public int getNumLegs() {
        return 4;
    }
}
```

注1: これらのコンストラクタは、実際には定義する必要はない。(デフォルトで定義されるコンストラクタと同等である。)

また、CreatureTest 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: CreatureTest.java

```
public class CreatureTest { // 01
    public static void main(String[] args) { // 02
        int i, j; // 03
        Creature[] creatures = new Creature[3]; // 04
        creatures[0] = new Snake(); // 05
        creatures[1] = new Frog(); // 06
        creatures[2] = new Lizard(); // 07

        creatures[0].age = 100; // 09
        // 10
        for (i=0; i<3; i++) { // 11
            for (j=0; j<3; j++) { // 12
                creatures[j].selfIntroduce(); // 13
                creatures[j].grow(); // 14
            } // 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) Creature クラスのフィールド age の値は、他のオブジェクトのメソッドからは直接変更できないようにしたい。( 例えば、CreatureTest クラスの 9 行目はコンパイル時にエラーになるようにしたい。)  
Creature 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) CreatureTest クラスの 9 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？ ( 空白は気にしなくて良い。 )

- III. 下のプログラムは、キーボードから打ち込まれた文字を画面に表示し、文字の色が変化するアニメーションを表示する Java アプレットである。キーイベントには、匿名クラスで対処する。

ファイル: ColorAnimation.java

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

public class ColorAnimation (i) {
```

```
    String text = "";
    float f = 0;
    Thread you;
```

```
    @Override
    public void start() {
        if (you == null) { // 念のためチェック
            you = new Thread(this);
            you.start();
        }
    }
```

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

```
    @Override
    public void init() {
        addKeyListener((ii) {
            public void keyPressed(KeyEvent e) {}
            public void keyReleased(KeyEvent e) {}
            public void keyTyped(KeyEvent e) {
                text += e.getKeyChar();
                (iii)
            }
        });
    }
```

```
    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.setColor(Color.getHSBColor(f, 1, 1));
        g.drawString(text, 20, 20);
    }
```

```
    public void run() {
        Thread me = Thread.currentThread();
        for(;; you == me; f+=0.02) {
            repaint(); // paintを間接的に呼出す
            try {
                Thread.sleep(200); // 200 ミリ秒お休み
            } catch (InterruptedException e) {}
        }
    }
```

```
}
```

プログラム中で使われている `getHSBColor` メソッドの API ドキュメントを次に示す。(ただし、問題には直接関係はない。)

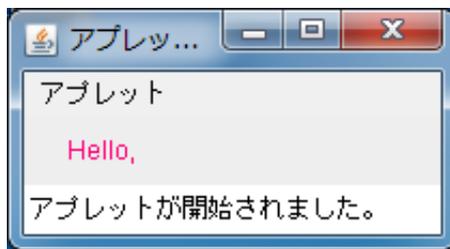
```
java.awt
クラス Color
public static Color getHSBColor(float h, float s, float b)

    HSB カラーモデルに指定された値に基づいて Color オブジェクトを生成し
    ます。

    s 成分および b 成分は、0 と 1 の間の浮動小数点値 (0.0 ~ 1.0 の範囲の数) に
    する必要があります。h 成分はどんな浮動小数点数でもかまいません。...(以
    下、省略)

    パラメータ:
    h - 色相成分 s - 色の彩度 b - 色の明度
    戻り値:
    指定された色相、彩度、明度を持つ Color オブジェクト
```

次にスクリーンショットを示す。



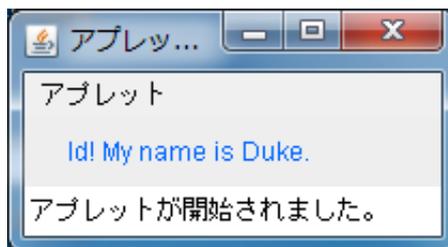
(1) “Hello, ” と打ったところ



(2) さらに、“world! ” と打ったところ

(i) ~ (ii) の空欄を埋めてプログラムを完成させよ。

(iii) このままでは、文字をこのあと打ちこんでいくとウィンドウの右側にあふれて見えなくなってしまう。そこで、画面に表示される文字を、最後に打ち込んだ 20 文字になるように制限して、古い文字は消えていくようにしたい。つまり、上の右側のスクリーンショットのあと、さらに “My name is Duke.” と打ったときに、



のように (“ld! My name is Duke.” だけ) 表示されるようにしたい。

(iii) の空欄部分にコードを挿入して、この機能を実装せよ。

ただし、Stringクラスの次のメソッドを用いること。

```
java.lang
```

```
クラス String
```

```
public int length()
```

この文字列の長さを返します。... (以下、省略)

戻り値:

このオブジェクトによって表される文字シーケンスの長さ

```
public String substring(int beginIndex)
```

この文字列の部分文字列である新しい文字列を返します。部分文字列は指定されたインデックスで始まり、この文字列の最後までになります。

例:

```
"unhappy".substring(2) returns "happy"
```

```
"Harbison".substring(3) returns "bison"
```

```
"emptiness".substring(9) returns "" (an empty string)
```

パラメータ:

beginIndex - 開始インデックス (この値を含む)

戻り値:

指定された部分文字列

以下に参考のために授業配布プリントの `KeyTest.java`, `UpDownButton3.java` (以上、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。

KeyTest.java

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

public class KeyTest extends JApplet implements KeyListener {
    int x=50, y=20;

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

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

    public void keyTyped(KeyEvent e) {
        int k = e.getKeyChar();
        if (k=='u') {
            y-=10;
        } else if (k=='d') {
            y+=10;
        }
        repaint();
    }

    public void keyReleased(KeyEvent e) {}
    public void keyPressed(KeyEvent e) {}
}
```

### UpDownButton3.java

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

public class UpDownButton3 extends JApplet {
    int x=20;

    @Override
    public void init() {
        JButton left = new JButton("Left");
        JButton right = new JButton("Right");
        left.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x-=10;
                repaint();
            }
        });
        right.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x+=10;
                repaint();
            }
        });
        setLayout(new FlowLayout());
        add(left); add(right);
    }

    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.drawString("HELLO WORLD!", x, 55);
    }
}
```

## 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 ('10年 2月 12日)

学籍番号		氏名	
------	--	----	--

I. (5×2)

(i).	
(ii).	

II. (5×3)

(i).	
(ii).	
(iii).	<p>私は 本です。脚は 本です。</p>

III. (5×3)

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

裏面に感想を記入する欄があります。

