

計算機ネットワーク I・テスト問題用紙

('06年7月27日・13:00 ~ 14:30)

解答上、その他の注意事項

- I. 問題は、問 I ~ V 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。
- VI. 合格は 100 点満点中 60 点以上とする。
(配点 — 期末テスト 80 点, レポート各 10 点)

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 (W) new (X) extends

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

(Point.java, ColorPoint.java のソースは問題文中に掲載する。)

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) {}
                }
            }
        }
    }
}
```

I. 次の各選択問題に答えよ。解答は各問の指示する選択肢から選べ。ただし、特に指定しない限り、選ぶべき選択肢は必ずしも 1つとは限らない。

(i) 次のうち、Java のソースファイルの名前としてふさわしくないのは、どれか？

- (A). Foo1.java (B). Foo_1.java (C). Foo-1.java (D). 1_1.java

(ii) 次のうち、Bar.java という Java のソースファイルをコンパイルし、実行するためのコマンドとして、正しい組み合わせはどれか？ 選択肢の中から 1つ選べ。

コンパイル 実行

- (A) javac Bar.java java Bar.class
(B) javac Bar.java java Bar
(C) javac Bar java Bar
(D) javac Bar java Bar.class

(iii) 0 から 4 までの数とその 2 倍を次のよう出力したい。

```
x が 0 のとき x の 2 倍は 0
x が 1 のとき x の 2 倍は 2
x が 2 のとき x の 2 倍は 4
x が 3 のとき x の 2 倍は 6
x が 4 のとき x の 2 倍は 8
```

次のプログラム:

```
int i;
for(i=0; i<5; i++) {
    _____;
}
```

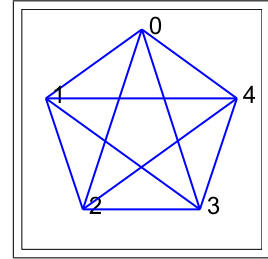
の空欄にはいる式を下の選択肢の中から 1つ選べ。

- (A). System.out.println("xが+i+のとき "x"の 2 倍は+(2*i)+)
(B). System.out.println("xが"+"i"+"のとき x の 2 倍は"+"2*i")
(C). System.out.println("xが"+"i+"のとき x の 2 倍は"+"(2*i)")
(D). System.out.println("xが"+"i+"のとき x の 2 倍は"+"(2*i)+")

(iv) 下のプログラムは、右のような図形を描く。

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

public class Pentagon extends JApplet {
    Color[] cs = {Color.red, Color.orange,
                 Color.green, Color.blue};
    public void paint(Graphics g) {
        int i, j, k=0;
        for (i=0; i<5; i++) {
            double theta1 = 2.0*Math.PI*i/5;
            int x1 = (int)(120-Math.sin(theta1)*100);
            int y1 = (int)(120-Math.cos(theta1)*100);
            g.setColor(Color.black);
            g.drawString(" "+i, x1, y1);
            for (j=i+1; j<5; j++) {
                double theta2 = 2.0*Math.PI*j/5;
                int x2 = (int)(120-Math.sin(theta2)*100);
                int y2 = (int)(120-Math.cos(theta2)*100);
                g.setColor(cs[k%4]);
                g.drawLine(x1, y1, x2, y2);
                k++;
            }
        }
    }
}
```



(1) 点 1-3 を結ぶ線は何色になるか？

(2) 点 2-4 を結ぶ線は何色になるか？

それぞれ下の選択肢から選べ。

(A) red (B) orange

(C) green (D) blue

(v) 次の文章は String クラスの charAt メソッドと length メソッドの説明の Java™ 2 Platform Standard Edition 5.0 API 仕様からの抜粋である。

```
public int charAt(int index)
```

指定されたインデックス位置にある char 値を返します。インデックスは、0 から length() - 1 の範囲になります。配列のインデックス付けの場合と同じように、シーケンスの最初の char 値のインデックスは 0、次の文字のインデックスは 1 と続きます。(後略)

パラメータ:

index – char 値のインデックス

戻り値:

文字列内の指定されたインデックス位置にある char 値。最初の char 値のインデックスが 0 になる

例外:

IndexOutOfBoundsException – index 引数が負の値、または文字列の長さ以上である場合

```
public int length()
```

この文字列の長さを返します。(後略)

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

このメソッドを使用しテストするプログラムを次のように作成する。

ファイル名: CharAtTest.java

```
public class CharAtTest {
    public static void main(String[] args) {
        String str1 = args[0];
        String str2 = args[1];

        int i;
        for (i=0; i<str1.length(); i++) {
            char c = ;
            if (c=='?') {
                System.out.print(str2);
            } else {
                System.out.print(c);
            }
        }
        System.out.println();
    }
}
```

このプログラムは、コマンドライン引数を2つ受け取り、第1引数の中の'?'という文字を、第2引数の文字列に置換して表示する。

実行例は次のようになる。

```
prompt> java CharAtTest ?_?_?_John Hello
Hello_Hello_Hello_John
prompt> java CharAtTest ?_?_?_Mary Bonjour
Bonjour_BonJour_BonJour_Mary
```

上のプログラムの空欄 を埋めよ。

II. 次の文章の空欄を埋めるのもっとも適切な言葉を指示された選択肢から選べ。

IPは必ずしもネットワーク的に隣接していない機器の間で、通信を行なうためのプロトコルである。IPではネットワーク機器に割り振られた (i) という、全世界で一意的番号を用いる。現行のIPv4では (i) は (ii) の整数である。

TCPはIPの上位層のプロトコルで、ポートという番号で、どのプロセスと通信するかを指定することができる。

JavaでTCPを使って通信を行なう時は、クライアント側はSocket、サーバ側はServerSocketというクラスのオブジェクトを用いる。SocketクラスのString型とint型の2引数のコンストラクタの場合は第2引数(int型の引数)に (iii) を、ServerSocketクラスのint型の1引数のコンストラクタの場合は、引数に (iv) を指定する。

TCPは順序制御や誤り訂正を含むため、ある程度重いプロトコルである。TCPと同じ階層に属する、より軽量なプロトコルとして、(v) がある。Javaで (v) を使って通信するためには、DatagramSocketクラスのオブジェクトを用いる。

(i) の選択肢

- (A) MACアドレス (B) IPアドレス (C) TCPアドレス
(D) メールアドレス (E) ポート (F) ドメイン名

(ii) の選択肢

- (A) 4バイト(32ビット) (B) 8バイト(64ビット)
(C) 16バイト(128ビット) (D) 32バイト(256ビット)

(iii), (iv) の選択肢

- (A) 自分のポート (B) 自分のファイルディスクリプタ
(C) 接続相手のポート (D) 接続相手のファイルディスクリプタ

(v) の選択肢

- (A) DHCP (B) ARP (C) HTTP (D) UDP

III. 円とユーロを両方向に変換するプログラムを Java Applet として作成する。この Applet は、2つのテキストフィールドがあり、片方のテキストフィールドの値を変えると、もう片方のテキストフィールドの値が変わる。ただし、1ユーロ = 145 円で計算し、端数は切り捨てる。次のプログラム中の空欄を埋めて、Applet を完成させよ。

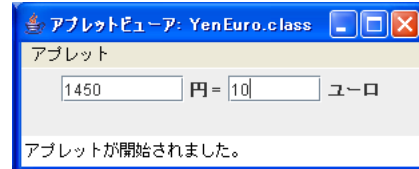
ファイル: YenEuro.java

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

public class YenEuro (i) {
    JTextField yen, euro;

    public void init() {
        yen = new JTextField("0", 8);
        euro = new JTextField("0", 6);
        (ii)
        setLayout(new FlowLayout());
        add(yen); add(new JLabel("円 ="));
        add(euro); add(new JLabel("ユーロ"));
    }

    public void actionPerformed(ActionEvent evt) {
        (iii)
    }
}
```



IV. 次の Point クラスを継承して、

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

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

    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();
    }
}
```

GridPoint クラスを定義する。GridPoint クラスは、1つの新しい int 型のフィールド grid を持つ。print メソッドは再定義されていて、x, y の正確な値の代わりに、それよりも小さい grid の倍数のうち、正確な値にもっとも近いものを表示する。例えば grid が 4 で x が 9, かつ y が 15 の時、(8, 12) と表示する。(x, y が負の数の場合は考慮する必要はない。)

ここで、参考のために ColorPoint クラスの定義を掲載する。

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

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

    @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;
            }
        }
    }

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

(1)~(3)の空欄を埋めて GridPoint クラスの定義を完成させよ。

```
public class (i) {
    public (ii)
    public GridPoint(int x0, int y0, int g0) {
        super(x0, y0); grid = g0;
    }
    @Override
    public void print() {
        (iii)
    }
}
```

さらに以下の文章の (iv), (v) の空欄を埋めよ。

「もし次のようなテスト用プログラムがあるとすると、

```
...
Point[] pts = new Point[2];
pts[0] = new Point(2, 4);
pts[1] = new GridPoint(12, 17, 5);
int i;
for (i=0; i<pts.length; i++) {
    pts[i].print();
}
...
```

この部分では“(iv)”と出力される。これは、print メソッドが動的束縛されているためである。もし、Java が C++ のように静的束縛を採用していれば、“(v)”と出力されるはずである。」

V. 下のプログラムは、右のような出力をする。

```
import java.io.*;

public class App5 {
    String str =
        "    The quick brown fox jumps over the lazy dog.    ";

    public static void main(String[] args) {
        int i;
        for(i=0; i<=str.length()-8; i++) {
            System.out.println(str.substring(i, i+8));
        }
    }
}
```

```
T
Th
The
The
The q
The qu
The qui
The quic
he quick
e quick
quick b
quick br
uick bro
ick brow
(以下略)
```

つまり、str という文字列から、8 文字の部分文字列を先頭から順番に取り出して標準出力に出力するプログラムである。

このプログラムを参考にして、いわゆる“電光掲示板”アプレット (Denko) を作成せよ。つまり、文字列の 8 文字の部分文字列を先頭から順番に取り出して次々に、drawString を用いて表示する。スレッドを用いて、100 ミリ秒間隔でアニメーションになるようにせよ。



Denko.java

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

public class Denko (i) {
    Thread thread=null;
    String str = "    The quick brown fox jumps over the lazy dog.    ";
    int i=0;

    @Override
    public void start() {
        if (thread==null) {
            (ii)
        }
    }

    @Override
    public void stop() {
        (iii)
    }

    @Override
    public void paint(Graphics g) {
        (iv)
    }

    public void run() {
        Thread thisThread = Thread.currentThread();
        while (thread==thisThred) {
            (v)
        }
    }
}
```

計算機ネットワーク I・テスト解答用紙 ('06年 7月 27日)

学籍番号		氏名	
------	--	----	--

I. (4, 3, 3, 3, 3, 3)

(i).		(ii).		(iii).	
(iv)-(1).		(iv)-(2).			
(v).					

II. (3, 3, 3, 3, 3)

(i).		(ii).		(iii).		(iv).		(v).	
------	--	-------	--	--------	--	-------	--	------	--

III. (3, 3, 5)

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

IV. (3, 3, 4, 4, 4)

(i).	
(ii).	
(iii).	
(iv).	
(v).	

V.

(3, 3, 3, 3, 5)

(i).	
(ii).	
(iii).	
(iv).	
(v).	

授業・テストの感想

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....
