

計算機ネットワーク工学 I・テスト問題用紙

(’02年7月18日・13:00～14:30)

解答上、その他の注意事項

- I. 問題は、問 I～VI 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. 解答欄がマス目になっている場合は、1字に1マスを用いること。特に空白にも必ず1マスを用いること
- V. 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。
- VI. ノート・プリント・参考書などは持ち込み可である。プリントは1冊 (やむを得ない場合は2冊) にまとめること。
- VII. 合格は100点満点中60点以上とする。
(配点 — 期末テスト70点, レポート各15点)

I. 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。

- Java アプレットはすべて **Applet** というクラスを継承して作成する。 **Applet** クラスにあらかじめ用意されているメソッドのうち、 メソッドは、ブラウザが最初にアプレットを読み込んだ時に呼ばれ、 メソッドは、アプレットをブラウザ上に描画する必要が生じた時に呼ばれる。 メソッドは メソッドを間接的に呼び出す事によって、強制的に画面の再描画を行う。

(1)~(3) の選択肢

(A) `init` (B) `start` (C) `stop` (D) `paint` (E) `repaint` (F) `run`

- Java で "128" という文字列を 128 という整数に変換するには、 というメソッドを用いる。これは C 言語の `atoi` 関数に相当する。

(4) の選択肢

(A) `String.toInteger` (B) `Integer.intValue`

(C) `String.intValue` (D) `Integer.parseInt`

II. Java でアプレットを作成する時に、「中間言語方式」のどのような点が利点になるか、「インタプリタ方式」「コンパイラ方式」と対比して簡潔に説明せよ。

III. CharPoint クラスは、次のPoint クラスを継承し、

```
class Point {
    public int x, y;
    public void print () {
        System.out.print("(" + x + ", " + y + ")");
    }
    public Point(int x0, int y0) {
        x=x0; y=y0;
    }
}
```

2つの新しいchar型のフィールドcharX, charYを持つ。printメソッドは再定義されていて、数字を出力する代わりにcharXをx個、charYをy個出力するものとする。(x, yが負の数の場合は考えなくて良い。)例えばxが2, yが4でcharXが*, charYが+のとき(**++++)と出力する。

(1)~(3)の空欄を埋めてクラスの定義を完成させよ。

```
class (1) {
    private (2)

    public void print() {
        System.out.print("(");
        (3)
        System.out.print(")");
    }

    public CharPoint(int x0, int y0, char c1, char c2) {
        super(x0, y0);
        charX = c1; charY = c2;
    }
}
```

さらに以下の文章の(4), (5)の空欄を埋めるのにもっとも適切な語を後の選択肢から選べ。

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(1, 2);
pts[1] = new CharPoint(3, 4, '!', '?');
int i;
for (i=0; i<pts.length; i++) {
    pts[i].print();
}
...
```

“(1, 2)(!!!!????)”と出力される。これは、printメソッドが(4)されているためである。もし、JavaがC++のように(5)を採用していれば、“(1, 2)(3, 4)”と出力されるは

ずである。

(4), (5) の選択肢

(A) 公開化 (B) カプセル化 (C) 動的束縛 (D) 静的束縛

IV. 下のプログラム (Type0) は、右のような出力をする。

```
import java.io.*;

public class Type0 {
    String message0 =
        "The quick brown fox jumps over the lazy dog.";
    String message = "        "+message0+"        ";
    public static void main(String[] args) {
        int i;
        for(i=0; i<=message.length()-8; i++) {
            System.out.println(message.substring(i, i+8));
        }
    }
}
```

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

つまり、`message` 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“タイピング練習”アプレット (Type) を作成せよ。つまり、文字列の4文字の部分文字列を先頭から取り出して `drawString` を用いて表示し、先頭の文字と同じ文字がキーボードから入力されれば、次の部分文字列が順番に表示されるようにせよ。ただし「.」が入力された時点で最初に戻る。

解答は下のプログラムの空欄を埋めるものを、次のアイテムから適当なものを選んで書け。

- ① `extends Applet`
- ② `g.drawString(message.substring(i, i+4), 10, 40);`
- ③ `implements KeyListener`
- ④ `repaint();`
- ⑤ `i=0;`
- ⑥ `i++;`
- ⑦ `else`
- ⑧ `addKeyListener(this);`
- ⑨ `if (k==message.charAt(i))`
- ⑩ `{`
- ⑪ `}`

ただし、一つの解答欄に一つのアイテムとは限らず、複数のアイテムを並べて書く必要がある場合がある。

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class Type  {
    String message =
        "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.  ";
    int i=0;

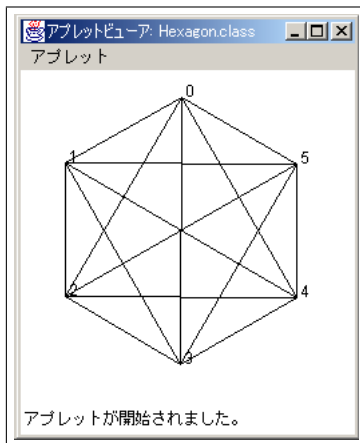
    public void init() {
        
    }

    public void paint(Graphics g) {
        
    }

    public void keyPressed(KeyEvent e) {
        int k=e.getKeyCode();
        
        if (k=='.') i=0;
    }
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
}
```

V. 次のプログラムは、下に示すような図形を描く。

```
import java.awt.*;
import java.applet.*;
public class Hexagon extends Applet {
    Color[] cs = {Color.red, Color.orange,
                  Color.green, Color.blue};
    public void paint(Graphics g) {
        int i, j, k=0;
        for (i=0; i<6; i++) {
            double theta_i = 2.0*Math.PI*i/6;
            int x_i = (int)(120-Math.sin(theta_i)*100);
            int y_i = (int)(120-Math.cos(theta_i)*100);
            g.setColor(Color.black);
            g.drawString(" "+i, x_i, y_i);
            for (j=i+1; j<6; j++) {
                double theta_j = 2.0*Math.PI*j/6;
                int x_j = (int)(120-Math.sin(theta_j)*100);
                int y_j = (int)(120-Math.cos(theta_j)*100);
                g.setColor(cs[k%4]);
                g.drawLine(x_i, y_i, x_j, y_j);
                k++;
            }
        }
    }
}
```



- (1) 点 1-4 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

VI. 次のようなスレッドを利用したプログラムを考える。

```
public class ThreadTest implements Runnable {
    int c=1;

    public void run() {
        while(c<100) {
            synchronized(this) { c*=2; } // ㉑
            synchronized(this) { System.out.print("A: "+c+", "); } // ㉒
        }
    }

    public void run2() {
        while(c<100) {
            synchronized(this) { c+=2; } // ㉓
            synchronized(this) { System.out.print("B: "+c+", "); } // ㉔
        }
    }

    public void test() {
        Thread thread = new Thread(this);
        thread.start();

        run2();
    }

    static public void main(String[] args){
        ThreadTest t = new ThreadTest();
        t.test();
    }
}
```

要は、run と run2 メソッドを別々のスレッドで並行に実行するプログラムである。

(1) 次のようなこのプログラムの出力例の最初の部分

B: 3, A: 6, A: 16, B: 16, ...

では、㉑, ㉒, ㉓, ㉔の文はどのような順番で実行されているか？

解答例: ㉑㉒㉓㉔㉓㉔㉓㉔

- (2) さらにrun とrun2 メソッドの定義をsynchronized の入れ方を変えて、次のように書き換える。

```
public void run() {
    while(c<100) {
        synchronized(this) {
            c*=2;
            System.out.print("A: "+c+", ");
        }
    }
}

public void run2() {
    while(c<100) {
        synchronized(this) {
            c+=2;
            System.out.print("B: "+c+", ");
        }
    }
}
```

このように書き換えた時の出力として**有り得ない**のは次の選択肢のうちどれか、1つ選べ。

- (A). A: 2, B: 4, A: 8, B: 10, B: 12, A: 24,...
- (B). A: 2, B: 4, A: 8, B: 10, A: 20, B: 22, ...
- (C). B: 3, A: 6, A: 16, B: 32, A: 34, B: 36, ...
- (D). B: 3, A: 6, B: 8, A: 16, B: 18, A: 36, ...

計算機ネットワーク工学 I・テスト解答用紙('02年7月18日)

学籍番号		氏名	
------	--	----	--

I. (3点×4)

(1).		(2).		(3).		(4).	
------	--	------	--	------	--	------	--

II. (7点)

.....

.....

.....

.....

.....

.....

.....

.....

.....

III. (4点×3, 3点×2)

(1).			
(2).			
(3).			
(4).		(5).	

IV. (4点×4)

(1).		(2).	
(3).		(4).	

V. (4点×2)

(1).		(2).	
------	--	------	--

VI. (5点, 4点)

(1).		(2).	
------	--	------	--

授業・テストの感想

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....
