

# 計算機ネットワーク I・テスト問題用紙

( '05 年 7 月 28 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。プリントは冊子としてまとめること。
- VII. 合格は 100 点満点中 60 点以上とする。  
(配点 — 期末テスト 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、`①==②`と書いて良い。

Ⓐ ActionListener    Ⓒ class    Ⓓ actionPerformed    Ⓔ getSource  
Ⓘ implements    Ⓜ JApplet    Ⓛ addActionListener    Ⓜ Math    Ⓝ null    Ⓟ public  
Ⓠ equals    Ⓡ Runnable    Ⓢ System.out.println    Ⓣ this    Ⓡ new    Ⓝ extends

I. 次の各問に答えよ。

- (i) 次の中で Java のクラス名として、問題なく使用できるものには、使用できるが慣習上望ましくないものには、使用できないものには×をつけよ。と×には理由も記せ。  
(A). Apple (B). kaki (C). Pear2 (D). Pine-Apple (E). 2Peaches
- (ii) 次の文章のうち Java の特徴を述べた文章としてふさわしくないものは、どれか?  
(A)~(D) の選択肢のなかから選べ。(解答の選択肢は必ずしもひとつとは限らない。)  
(A). Java は動的束縛を標準として提供せず、その分の高速化を図っている。  
(B). Java はゴミ集めの機能を標準で持っていないので、メモリが少なくても高速に動作する。  
(C). Java はオブジェクト指向言語であるが、C 言語や C++ 言語との互換性は持っていない。  
(D). Java はポインタをプログラマに提供しないことで、安全性を確保している。
- (iii) 次のプログラムは画面に“Good Luck!!”と表示するだけのアプレット(ファイル名: Nanika.java)である。空欄を埋めよ。

ファイル名: Nanika.java

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

public [ ] (iii) {
    public void paint(Graphics g) {
        g.drawString("Good Luck!!", 50, 50);
    }
}
```

- (iv) 下のプログラムは、13 番ポートでクライアントからの接続を待ち受け、接続したら現在の時刻を送信する(だけの)サーバプログラムである。空欄を埋めよ。

ファイル名: Daytime.java

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

public class Daytime {
    public static void main(String args[]){
        try {
            ServerSocket servsock = [ ] (iv);
            while(true){
                Socket sock = servsock.accept();
                PrintStream out = new PrintStream(sock.getOutputStream());
                out.println(new Date().toString());
                sock.close();
            }
        } catch (Exception e) {}
    }
}
```

- (v) 次の文章は String クラスの compareTo 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 compareTo(String anotherString)
```

2つの文字列を辞書的に比較します。比較は文字列内のそれぞれの文字の Unicode 値に基づいて行われます。この String オブジェクトによって表される文字シーケンスが、引数文字列によって表される文字シーケンスと辞書的に比較されます。この String オブジェクトが辞書的に引数文字列より前にある場合は、結果は負の整数になります。この String オブジェクトが辞書的に引数文字列の後ろにある場合、結果は正の整数になります。文字列が等しい場合、結果は 0 になります。（後略）

パラメータ:

anotherString – 比較対象の String

戻り値:

引数文字列がこの文字列に等しい場合は、値 0。この文字列が文字列引数より辞書的に小さい場合は、0 より小さい値。この文字列が文字列引数より辞書的に大きい場合は、0 より大きい値

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: CompareToTest.java

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

        (v)

        if (result > 0) {
            System.out.println(str1+" > "+str2);
        } else if (result < 0) {
            System.out.println(str1+" < "+str2);
        } else {
            System.out.println(str1+" = "+str2);
        }
    }
}
```

このプログラムは、コマンドラインパラメータを2つ受け取り、その2つの文字列を比較して、>, <, =のいずれかを判定する。実行例は次のようになる。

```
prompt> java CompareToTest baz bar
baz > bar
prompt> java CompareToTest bar bar
bar = bar
prompt> java CompareToTest bar baz
bar < baz
```

上のプログラムの空欄を埋めよ。

II. 次のようなクラスを Java で定義する。

「クラス Foo は 2 つの public な int 型のフィールド a, b とメソッド print を持ち、print は、a, b の値とその和を表示する。さらに 2 つの int 型の引数をとる public なコンストラクタを持ち、このコンストラクタの引数は、それぞれフィールド a, b の初期値となる。

クラス Bar は Foo を継承し、public な int 型のフィールド c を持ち、print メソッドは、a, b, c の値とその和を表示する。さらに 3 つの int 型の引数をとる public なコンストラクタを持ち、このコンストラクタの引数は、それぞれフィールド a, b, c の初期値となる。」

- (i) ~ (ii) 下の空欄をフィールドとコンストラクタの定義で埋めて、クラス Foo, Bar の定義を完成させよ。

ファイル名: FooBarTest.java

```
class Foo {
    (i)
    public void print() {
        System.out.println(""+a+""+b+"="+a+b);
    }
}

class Bar extends Foo {
    (ii)
    public void print() {
        System.out.println(""+a+""+b+""+c+"="+a+b+c);
    }
}
```

- (iii) Bar の print メソッドのように、サブクラスでスーパークラスのメソッドを書き換えることをオブジェクト指向の用語で何と言うか?

(A). オーバーヒート (B). オーバーロード (C). オーバーライド (D). オーバーラン

- (iv) つぎのようなコードの振舞に対する説明として正しいものを、つぎの選択肢から選べ、

ファイル名: FooBarTest.java ( 続き )

```
public class FooBarTest {
    public static void main(String[] args) {
        Bar b = new Bar(1, 2, 3); // 1行目
        Foo f = b; // 2行目
        b.print(); // 3行目
        f.print(); // 4行目
    }
}
```

- (A). 2行目がエラーとなってコンパイルできない。  
(B). コンパイルできるが実行時に2行目でエラーとなる。  
(C). コンパイルできるが実行時に4行目でエラーとなる。  
(D). 3行めでは Bar の print メソッドが、4行めでは Foo の print メソッドが起動される。  
(E). 3行め、4行めともに Bar の print メソッドが起動される。  
(F). 3行め、4行めともに Foo の print メソッドが起動される。

III. 下のプログラムは、“<”、“>”というラベルのついた2つのボタンと円を表示し、“<”ボタンを押せば円が小さくなり、“>”ボタンを押せば円が大きくなる、というJava アプレットである。

ファイル名: BallButton.java

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

public class BallButton (i) {
    int r=50;
    JButton b1, b2;
    public void init() {
        b1 = new JButton("<");
        b2 = new JButton(">");
        (ii)
        getContentPane().setLayout(new FlowLayout());
        getContentPane().add(b1);
        getContentPane().add(b2);
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.fillOval(100-r, 100-r, 2*r, 2*r);
    }

    public void (iii) (ActionEvent evt) {
        if ( (iv) ) {
            r--;
        } else if ( (v) ) {
            r++;
        }
        repaint();
    }
}
```

空欄を埋めてプログラムを完成させよ。

IV. 下のプログラムは、円が時間の経過とともに大きくなったり小さくなったりするアニメーションを表示する Java アプレットである。

ファイル名: BallThread.java

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

public class BallThread (i) {
    int r=50;
    int n=0;
    Thread myThread=null;

    public void start() {
        if (myThread==null) {
            myThread = new Thread(this);
            myThread.start();
        }
    }

    public void stop() {
        (ii)
    }

    public void run() {
        Thread t0 = Thread.currentThread();
        while((iii) ) {
            r = 50+(int)(50*Math.sin(Math.PI*n/50));
            repaint();
            try {
                Thread.sleep(30);
            } catch (InterruptedException e) {}
            n++;
        }
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.fillOval(100-r, 100-r, 2*r, 2*r);
    }
}
```

空欄を埋めてプログラムを完成させよ。

V. 下のプログラムはある図形を描画する Java アプレットである。

ファイル: Hatena.java

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

public class Hatena extends JApplet {
    public static Color[] colors = {Color.red, Color.yellow, Color.green,
                                     Color.cyan, Color.blue, Color.magenta};

    public void paint(Graphics g) {
        int i, tmp, n=18, x0=200, y0=200;
        int x=0, y=0, a=1, b=1, dx=1, dy=1;
        for (i=0; i<n; i++) {
            g.setColor(colors[i%colors.length]);
            (i)
            (ii)
            x+=a*dx; y+=a*dy;
            tmp=a+b; a=b; b=tmp;
            tmp=-dy; dy=dx; dx=tmp;
        }
    }
}
```

- (i) Graphics クラスの fillRect メソッドは、x, y, width, height の 4 つの引数を受け取って、左上の座標が (x, y), 幅 width, 高さ height の長方形を塗りつぶす。この場合 width, height は正でなければならない。このメソッドを利用して、dx, dy の正負に関わらず、(x0+x, y0+y) と (x0+x+a\*dx, y0+y+a\*dy) を対角線とする正方形を塗りつぶすコードを記述せよ。(上のプログラムの (i) の部分に入る。)

次の Math クラスのクラスメソッド (static メソッド) を適宜利用して良い。

```
public static int abs(int a)
    int 値の絶対値を返します。
public static int max(int a, int b)
    2 つの int 値のうち大きいほうを返します。
public static int min(int a, int b)
    2 つの int 値のうち小さいほうを返します。
```

- (ii) さらにデバッグのためにその時点の x と y の値を「x=~ , y=~」( ~の部分には具体的な数値が入る ) のようなメッセージを標準出力 (System.out) に出力するコードを追加せよ。(上のプログラムの (ii) の部分に入る)
- (iii) for ループの最初のところ (g.setColor(~) を実行する直前) で、i が 1~3 のときの a, b, dx, dy の値を答えよ。
- (iv) 座標 (190, 200) (= (x0-10, y0+0)) の点は何色に塗られるか?  
(R). red, (Y). yellow, (G). green, (C). cyan, (B). blue, (M). magenta, (X). grey のうちから選べ。(最後の grey は背景色のままという意味である。また、このプログラムは正方形の周上の点を除き、おなじ場所を 2 回以上塗ることはないので、最初に塗られる色を答えれば良い。)

計算機ネットワーク I・テスト解答用紙 ('05 年 7 月 28 日)

学籍番号		氏名	
------	--	----	--

I. (5, 3, 3, 3, 5)

(i).	クラス名	判定	理由
	(A). Apple		
	(B). kaki		
	(C). Pear2		
	(D). Pine-Apple		
	(E). 2Peaches		
(ii).			
(iii).			
(iv).			
(v).			

II. (4, 4, 3, 5)

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

III. (3, 3, 3, 2, 2)

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

IV. (3, 4, 4)

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

V. (5, 5, 6, 5)

(i).					
(ii).					
(iii).	i	a	b	dx	dy
	1				
	2				
	3				
(iv).					

授業・テストの感想

---

.....

.....

.....

.....

.....

.....

.....

.....

.....

---