

計算機ネットワーク I (Java 編)・テスト問題用紙

('09年7月31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。

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、`Ⓓ==Ⓔ` と書いて良い。(必ず○で囲むこと。)

Ⓐ ActionListener Ⓒ class Ⓓ actionPerformed Ⓔ ActionEvent
Ⓖ getSource Ⓘ implements Ⓜ JApplet Ⓛ addActionListener Ⓜ Math
Ⓝ null Ⓟ public Ⓡ repaint Ⓢ System.out.println Ⓣ this Ⓥ void
Ⓦ new Ⓧ extends

また、参考のために問題用紙の末尾に授業プリントの `UpDownButton.java`, `UpDownButton3.java` のソースを掲載する。

I. 次の各多肢選択問題に答えよ。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 のクラスの名前として (文法的に) 許されないのは、どれか?

- (A). A_B_C (B). Rock'n'Roll (C). 123Daaah (D). Lotus123

(ii) 次の Java に関する文章のうち正しいものはどれか?

- (A). Java は、オブジェクト指向という概念を最初に導入したプログラミング言語として知られ、C#や Ruby、Python など、多くのその後設計された言語に影響を与えている。
(B). Java アプレット・Java サブレットは、どちらも Web のクライアント側 (ブラウザ上) で動作するが、採用するセキュリティモデルが少し異なる。
(C). Java が純粋なコンパイラ方式でもインタプリタ方式でもなく中間言語方式をとるのは、主に機種非依存性と効率を両立するためである。
(D). JavaScript は Java のサブセットであり、JavaScript のプログラムは Java プログラムとしても実行することができる。

(iii) 要素の型が Integer 型であるような、ArrayList 型 (サイズ変更可能な配列の型) の変数 xs を宣言したい。正しい書き方を 1つ 以下の選択肢から選べ。

- (A). ArrayList<Integer> xs = new ArrayList<Integer>();
(B). Integer<ArrayList> xs = new Integer<ArrayList>();
(C). ArrayList.Integer<> xs = new ArrayList.Integer<>();
(D). Integer.ArrayList<> xs = new Integer.ArrayList<>();

(iv) 1 から 5 までの数とその階乗を次のよう出力したい。

```
n が 1 のとき n! は 1
n が 2 のとき n! は 2
n が 3 のとき n! は 6
n が 4 のとき n! は 24
n が 5 のとき n! は 120
```

次のプログラム (の一部):

```
int n;
int f = 1;
for (n=1; n<=5; n++) {
    f *= n;
    ;
}
```

の空欄 にふさわしい式を下の選択肢の中から選べ。

- (A). System.out.printf("n が%d のとき n! は%d%n", n, f)
(B). System.out.println("n が"+n+"のとき n! は"+f)
(C). System.out.println("n が'n' のとき n! は'f'")
(D). System.out.println("n が+n+のとき n! は+f+")

II. 次のプログラムは、コマンドライン引数として与えられた整数の積を計算する。

```
public class ProductArgs {
    public static void main( (i) args) {
        int k, p = 1;
        for (k=0; k<args.length; k++) {
            int a = (ii)(args[k]);
            p *= a;
        }
        System.out.printf("コマンドライン引数の積は %d です。 %n", p);
    }
}
```

実行例は以下のようなになる。

```
% java ProductArgs 1 2 3 4 5
コマンドライン引数の積は 120 です。
% java ProductArgs 1 2 4 8 16
コマンドライン引数の積は 1024 です。
```

(i) ~ (ii) の空欄にふさわしいものを以下の選択肢から 1つずつ 選べ。

(i) の選択肢

(A). char[][] (B). String[] (C). char** (D). String<>

(ii) の選択肢

(A). Integer.toString (B). Double.parseDouble
(C). atoi (D). Integer.parseInt

III. 次の文章はFont クラス (java.awt.Font クラス) の deriveFont メソッドの説明の Java™ API 仕様からの抜粋である。

```
public Font deriveFont(float size)
    現在の Font オブジェクトを複製し新しいサイズを適用することにより、新しい
    Font オブジェクトを生成します .

    パラメータ:
        size - 新しい Font のサイズ

    戻り値:
        新しい Font オブジェクト
```

このメソッドを使用し、テストするプログラム (アプレット) を次のように作成する。

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

public class DeriveFontTest extends JApplet {
    public void paint(Graphics g) {
        float sz = 8;
        int i, y = 12;
        Font f = new Font("Serif", Font.PLAIN, (int)sz);

        for (i=0; i<6; i++) {
            sz *= 1.414; // 生成したいフォントのサイズ
             // deriveFont でフォントを生成
            g.setFont(f); // 実際にセット
            y += (int)sz + 1; // 描画開始位置の y 座標をずらす
            g.drawString("Hello World!", 10, y);
        }
        return;
    }
}
```

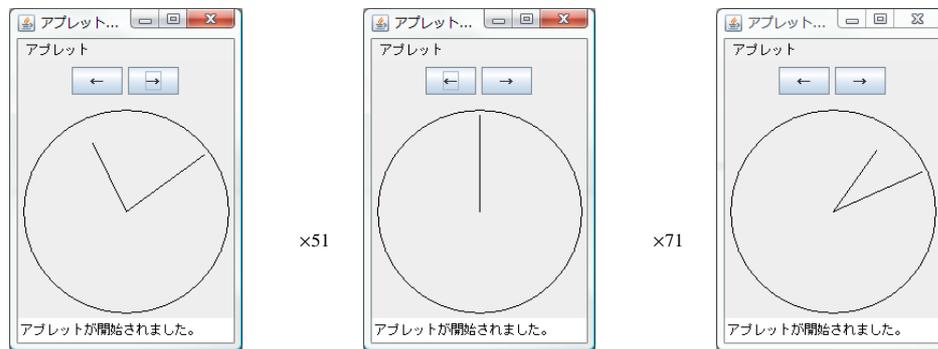
このプログラムは、フォントサイズを 8×1.414 から $8 \times (1.414)^6$ まで 6 段階変化させて、“Hello World!” とアプレットのウインドウ上に表示する。

スクリーンショットは次のようになる。



上のプログラムの空欄 を埋めて、プログラムを完成させよ。

IV. 次のプログラム (ToyClock クラス) は、“←”, “→” という 2 つのボタンと、時計の針を表示し、“←” ボタンを押せば時計が一分戻り、“→” ボタンを押せば時計が一分進む、という“おもちゃの時計” アプレットである。



スクリーンショット

以下の空欄 (i) ~ (iii-2) を埋めて、プログラムを完成させよ。

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

public class ToyClock (i) {
    JButton cBtn, aBtn;
    int m = 0;

    @Override
    public void init() {
        aBtn = new JButton(" ");
        cBtn = new JButton(" ");
        (ii)
        setLayout(new FlowLayout());
        add(aBtn); add(cBtn);
    }

    public void actionPerformed(ActionEvent aEvt) {
        if ((iii-1)) {
            m--;
        } else if ((iii-2)) {
            m++;
        }
        repaint();
    }
}
```

(paint メソッドは次ページに掲載)

(ToyClock クラスつづき)

```
@Override
public void paint(Graphics g) {
    super.paint(g);
    int cx = 100, cy = 140;           // 中心の座標
    g.drawOval(cx-95, cy-95, 190, 190); // 縁
    double angle1 = m * 6 / 180.0 * Math.PI; // 1分で長針は 6度動く
    int x1 = (int)(cx+90*Math.sin(angle1));
    int y1 = (int)(cy-90*Math.cos(angle1));
    g.drawLine(cx, cy, x1, y1);      // 長針
    double angle2 = m * 0.5 / 180 * Math.PI; // 1分で短針は 0.5度動く
    int x2 = (int)(cx+70*Math.sin(angle2));
    int y2 = (int)(cy-70*Math.cos(angle2));
    g.drawLine(cx, cy, x2, y2);     // 短針
}
}
```

さらに、同じ動作をするプログラムを匿名(無名)クラスを利用して ToyClock2 クラスとして実装する。以下の空欄 (iv) ~ (v-2) を埋めて、プログラムを完成させよ。

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

public class ToyClock2 (iv) {
    int m = 0;

    @Override
    public void init() {
        JButton aBtn = new JButton(" ");
        JButton cBtn = new JButton(" ");
        aBtn.addActionListener(
            (v-1)
        );
        cBtn.addActionListener(
            (v-2)
        );
        setLayout(new FlowLayout());
        add(aBtn); add(cBtn);
    }

    /* paint メソッドは ToyClock クラスと同一なので省略 */
}
```

参考: (例題 4.6.1: ボタン 2 つを使ってテキストを左右に移動する。)

ファイル UpDownButton.java

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

public class UpDownButton extends JApplet implements ActionListener {
    int x=20;
    JButton left, right;

    @Override
    public void init() {
        left = new JButton("Left");
        right = new JButton("Right");
        left.addActionListener(this);
        right.addActionListener(this);
        setLayout(new FlowLayout());
        add(left); add(right);
    }

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

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == left) { // Left が押された
            x-=10;
        }
        else if (e.getSource() == right) { // Right が押された
            x+=10;
        }
        repaint();
    }
}
```

参考: (例題 4.8.1: ボタン 2 つを使ってテキストを左右に移動する。匿名クラス版)

ファイル 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);
    }
}
```

計算機ネットワーク I (Java 編)・テスト解答用紙 ('09年 7月 31日)

学籍番号		氏名	
------	--	----	--

I. (4,4,4,4)

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

II. (4,4)

(i).		(ii).	
------	--	-------	--

III. (6)

--

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

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

(裏面に授業・テストの感想を書く欄があります)

