

オブジェクト指向言語・期末テスト問題用紙

(2013 年 8 月 09 日 ・ 10:30 ~ 12:00)

解答上、その他の注意事項

- I. 問題は、問 I ~ VII 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。筆記用具・時計・学生証以外のものは、かばんの中などにしまうこと。
- VI. テストの配点は 80 点である。合格はレポートの得点を加えて、100 点満点中 60 点以上とする。

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ため、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 Ⓐ addActionListener Ⓒ class Ⓓ actionPerformed
Ⓔ ActionEvent Ⓒ getSource Ⓓ implements Ⓙ JApplet Ⓚ KeyListener
Ⓚ addKeyListener Ⓜ MouseListener Ⓜ addMouseListener Ⓝ null
Ⓟ public Ⓒ equals Ⓡ Runnable Ⓢ 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 `KeyTest.java`, `UpDownButton.java`, `UpDownButton3.java`, `BubbleSort1.java`, `BubbleSort2.java`, `Point.java`, `ColorPoint.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 一つとは限らない。

(i) 次のうち Java のクラスの名前として使うことができるのは、どれか?

(A). Apple_2 (B). 2_2_1 (C). X___ (D). Toi:2

(ii) 次の Java に関する文章のうち正しいものはどれか?

(A). Java は並行に複数の処理を行なうためのスレッドを、言語に標準の機構として持っていないので、スレッドを使ったプログラムは機種に依存する。

(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、使用する API が異なる。

(C). Java は中間言語方式を取っている。JDK では中間言語へのコンパイラーが `javac`、中間言語のインタプリターのコマンドが `java` である。

(D). Java のプログラムが HTML ファイルなどにソースコードの形のままで埋め込まれたものが JavaScript である。

(iii) Java で要素の型が `int[]` 型であるような、`ArrayList` 型(サイズ変更可能な配列の型)の変数 `a` 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). `int[]<ArrayList> a = new int[]<ArrayList>();`

(B). `int[<ArrayList>] a = new int[<ArrayList>]();`

(C). `ArrayList[int[]] a = new ArrayList[int[]]();`

(D). `ArrayList<int[]> a = new ArrayList<int[]>();`

II. Java でコマンドライン引数として与えられた整数とその 10 倍の数を次のよう出力したい。

java Test2 2 3 5 7 11 の実行結果

```
n が 2 のとき 10 倍返しは 20
n が 3 のとき 10 倍返しは 30
n が 5 のとき 10 倍返しは 50
n が 7 のとき 10 倍返しは 70
n が 11 のとき 10 倍返しは 110
```

次のプログラムについて、

```
public class Test2 {
    public static void main(String[] args) {
        int i, k = (i);
        for (i=0; i<k; i++) {
            int n = (ii);
            (iii);
        }
    }
}
```

空欄 (i) ~ (iii) にふさわしい式を下の選択肢の中から選べ。特に指定しない限り、選ぶべき選択肢は必ずしも 一つとは限らない。

(i) の選択肢

- (A). args.length
- (B). length(args)
- (C). args[length]
- (D). sizeof(args)

(ii) の選択肢

- (A). atoi(args[i])
- (B). atoi(args(i))
- (C). Integer.parseInt(args(i))
- (D). Integer.parseInt(args[i])

(iii) の選択肢

- (A). System.out.println("n が " n " のとき 10 倍返しは " n*10)
- (B). System.out.printf("n が %d のとき 10 倍返しは %d%n", n, n*10)
- (C). System.out.println("n が ¥"n¥" のとき 10 倍返しは ¥"n*10¥")
- (D). System.out.println("n が "+n+" のとき 10 倍返しは "+n*10)

- III. test.TestA クラス (test パッケージの TestA クラス)、 TestB クラス (無名パッケージの TestB クラス) および TestTest クラス (無名パッケージの TestTest クラス) をそれぞれ次のように定義する

パス: test/TestA.java

```
package test;

public class TestA {
    private String x;
    String y;
    public String z;

    public TestA(String a, String b, String c) {
        x = a; y = b; z = c;
    }
}
```

パス: ./TestB.java

```
public class TestB {
    private String x;
    String y;
    public String z;

    public TestB(String a, String b, String c) {
        x = a; y = b; z = c;
    }
}
```

パス: ./TestTest.java

```
1 import test.TestA;
2
3 public class TestTest {
4     public static void main(String[] args) {
5         TestA testA = new TestA("あ", "い", "う");
6         TestB testB = new TestB("あ", "い", "う");
7         System.out.println(testA.x); // A
8         System.out.println(testA.y); // B
9         System.out.println(testA.z); // C
10        System.out.println(testB.x); // D
11        System.out.println(testB.y); // E
12        System.out.println(testB.z); // F
13    }
14 }
```

A~Fの行 (TestTest.java の 7~12 行目) で (コンパイル時に) エラーにならない行にはを、エラーになる行には×をつけよ。(A~C と D~F はそれぞれ完答しなければ得点にならない。)

IV. 次の文章は java.awt.Color クラスの getHSBColor メソッドと getRGB メソッドの説明の Java™ API 仕様からの抜粋である。

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。この数のフロア (この数以下で最大の整数) が減算され、0 から 1 の間的小数部が作成されます。この小数点数に 360 が乗算され、HSB カラーモデルの色相角度が生成されます。

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

public int getRGB()

デフォルトの sRGB ColorModel の色を表す RGB 値を返します (ビットのうち 24 ~ 31 はアルファ、16 ~ 23 は赤、8 ~ 15 は緑、0 ~ 7 は青)。

戻り値:

デフォルトの sRGB ColorModel の色の RGB 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: HSBCColorTest.java

```
import java.awt.Color;
```

```
public class HSBCColorTest {  
    public static void main(String[] args) {  
        int t;  
        for (t = 0; t < 360; t += 30) {  
            float h = t / (float)360;  
            Color c = ;  
            int rgb = ;  
            System.out.printf("t=%3d, rgb=%08x%n", t, rgb);  
        }  
    }  
}
```

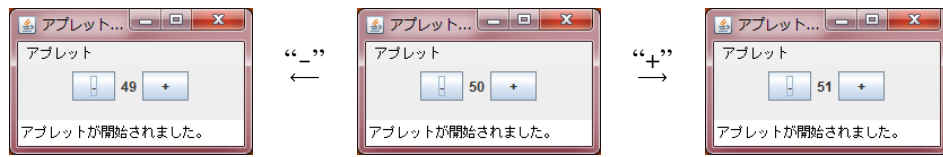
このプログラムは、 t が 0, 30, 60, ..., 330 と 30 刻みの値を取るときに、 $(h, s, b) = (\frac{t}{360}, 1, 1)$ の 3 つを引数とする getHSBColor メソッドの結果の RGB 値を 16 進数で表示する。

実行例は次のようになる。

```
t = 0, rgb = ffff0000
t = 30, rgb = ffff8000
t = 60, rgb = fffffff0
t = 90, rgb = ff80ff00
t = 120, rgb = ff00ff00
t = 150, rgb = ff00ff80
t = 180, rgb = ff00ffff
t = 210, rgb = ff0080ff
t = 240, rgb = ff0000ff
t = 270, rgb = ff8000ff
t = 300, rgb = ffff00ff
t = 330, rgb = ffff0080
```

上のプログラムの空欄 (i) に $(h, 1, 1)$ を引数とする `getHSBColor` メソッドの値を表す式、空欄 (ii) に `Color` 型の変数 `c` の RGB 値を表す式、をそれぞれ埋めよ。

- V. 以下は、“-”, “+” という 2 つのボタンと、整数の表示されたラベルを表示し、“-” ボタンを押せば、ラベル上の数がデクリメントされ、“+” ボタンを押せば、ラベル上の数がインクリメントされる、という Java アプレットのプログラムである。



スクリーンショット

(i) ~ (iii) の空欄を埋めて、プログラムを完成させよ。

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class IncDecButton extends JApplet (i) {
6      JButton decB, incB;
7      JLabel numL;
8      int x = 50;
9
10     @Override
11     public void init() {
12         decB = new JButton("-");
13         numL = new JLabel(""+x);
14         incB = new JButton("+");
15         (ii)
16         setLayout(new FlowLayout());
17         add(decB); add(numL); add(incB);
18     }
19
20     public void actionPerformed(ActionEvent e) {
21         if ((iii)) {
22             x--;
23         } else if ( /* (iii) のヒントになるので省略 */ ) {
24             x++;
25         }
26         numL.setText(""+x);
27     }
28 }

```

また、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、IncDecButton2 クラスとして実装する。(iv) ~ (v) の空欄を埋めて、プログラムを完成させよ。節約のため、必要なら「IncDecButton.javaの 行めから 行めと同じ」というような書き方をしても良い。

```

1 // import は同一なので省略する

```

```
2
3 public class IncDecButton2 extends JApplet {
4     int x = 50;
5
6     @Override
7     public void init() {
8         JButton decB = new JButton("-");
9         (iv) JLabel numL = new JLabel(""+x);
10        JButton incB = new JButton("+");
11        decB.addActionListener(
12            (v)
13        );
14        incB.addActionListener(
15            /* (v) のヒントになるため省略 */
16        );
17        setLayout(new FlowLayout());
18        add(decB); add(numL); add(incB);
19    }
20 }
```

VI. 次に定義されるクラス Creature を継承して、

ファイル: Creature.java

```
1 public class Creature {
2     public int age;
3
4     public Creature() { // 注 1
5         age = 0;
6     }
7
8     public int getAge() {
9         return age;
10    }
11
12    public void grow() {
13        age++;
14    }
15
16    public String getName() {
17        return "何かの生物";
18    }
19
20    public int getNumLegs() {
21        return -1;
22    }
23
24    public void selfIntroduce() {
25        System.out.printf("私は%sです。%d歳です。脚は%d本です。%n",
26                            getName(), getAge(), getNumLegs());
27    }
28 }
```

3つのクラス Snake, Frog, Butterfly を定義する。

ファイル: Snake.java

```
1 public class Snake (i) {
2     public Snake() { // 注 1
3         super();
4     }
5
6     @Override
7     public String getName() {
8         return "へび";
9     }
10
11    @Override
12    public int getNumLegs() {
13        return 0;
14    }
```

15 }

ファイル: Frog.java

```
1 public class Frog  {
2     public Frog() { // 注1
3         super();
4     }
5
6     @Override
7     public String getName() {
8         if (getAge()<=1) {
9             return "オタマジヤクシ";
10        } else {
11            return "カエル";
12        }
13    }
14
15    @Override
16    public int getNumLegs() {
17        if (getAge()==0) {
18            return 0;
19        } else if (getAge()==1) {
20            return 2;
21        } else {
22            return 4;
23        }
24    }
25 }
```

ファイル: Butterfly.java

```
1 public class Butterfly  {
2     public Butterfly() { // 注1
3         super();
4     }
5
6     @Override
7     public String getName() {
8         if (getAge()==0) {
9             return "イモムシ";
10        } else if (getAge()==1){
11            return "サナギ";
12        } else {
13            return "チョウ";
14        }
15    }
16
17    @Override
18    public int getNumLegs() {
```

```

19         if (getAge()==0) {
20             return 16;
21         } else if (getAge()==1) {
22             return 0;
23         } else {
24             return 6;
25         }
26     }
27 }

```

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

```

1 public class CreatureTest {
2     public static void main(String[] args) {
3         int i, j;
4         Creature[] creatures = new Creature[3];
5         creatures[0] = new Snake();
6         creatures[1] = new Frog();
7         creatures[2] = new Butterfly();
8
9         creatures[0].age = 100; // あとでコメントアウトする
10
11        for (i=0; i<3; i++) {
12            for (j=0; j<3; j++) {
13                creatures[j].selfIntroduce();
14                creatures[j].grow();
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、変更ができないようにしたい。(例えば、CreatureTest.java 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.java の定義の何行目をどのように変更すれば良いか?

(行の左端の数字が行番号を表す。)

なお、この問で定義されるクラスはすべて同一パッケージにあると仮定する。

(iii) CreatureTest.java 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。)

VII. 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: TypingAnimation.java

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

public class TypingAnimation extends JApplet (i) {
    String text = "Please_type_in!";
    double t = 0;
    Thread thrd;

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

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

    @Override
    public void init() {
        addKeyListener(new KeyListener() {
            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(new Color((int)(127.5 + 127.5 * Math.sin(t)),
                               (int)(127.5 - 127.5 * Math.sin(t)),
                               255));
        g.drawString(text, 20, 20);
    }

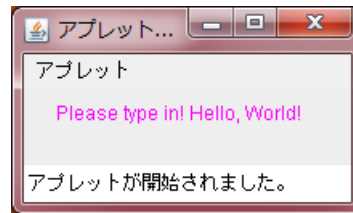
    public void run() {
        Thread me = Thread.currentThread();
    }
}
```

```

for( ; (ii); t += 0.1) {
    repaint(); // paintを間接的に呼出す
    try {
        Thread.sleep(200); // 200 ミリ秒お休み
    } catch (InterruptedException e) {}
}
}
}

```

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イ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 30 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. I live in Japan.” と打ったときに、



のように (“name is Duke. I live in Japan.” 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`, `UpDownButton.java`, `UpDownButton3.java`, `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() {
        setFocusable(true);
        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) {}
}
```

UpDownButton.java

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

public class UpDownButton extends JApplet implements ActionListener {
    int x=20;
    JButton lBtn, rBtn;

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

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

    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();
        if (source == lBtn) {          // lBtnが押された
            x-=10;
        } else if (source == rBtn) {  // rBtnが押された
        }
    }
}
```

```

        x+=10;
    }
    repaint();
}
}

```

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 lBtn = new JButton("Left");
        JButton rBtn = new JButton("Right");
        lBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x-=10;
                repaint();
            }
        });
        rBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                x+=10;
                repaint();
            }
        });
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

    @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;
    // start, stop, paint メソッドは BubbleSort1.java と同一なので省略する。

    @Override
    public void init() {
        JButton step = new JButton("Step");
        step.addActionListener(this);
        setLayout(new FlowLayout());
        add(step);
    }

    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) {}
}
}
thread = null;
}
}

```

Point.java

```

public class Point {
    public int x, y;

    public void move(int dx, int dy) {
        x += dx; y += dy;
    }

    public double distance() {
        return Math.sqrt(x*x+y*y);
    }

    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", ..., "white"};
    private String color;

    @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 = c; 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);
        if (color==null) color = "black";
    }

    public String getColor() { return color; }
}

```

オブジェクト指向言語・期末テスト解答用紙(2013年8月09日)

学籍番号		氏名	
------	--	----	--

I. (3×3)

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

II. (3×3)

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

III. (3×2)

A	B	C	D	E	F

IV. (5×2)

(i).	
(ii).	

V. (4, 4, 4, 4, 6)

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

VI.

(4 × 3)

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

VII.

(4 × 3)

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

授業・テストの感想

.....

.....

.....

.....

.....

.....
