

# オブジェクト指向言語・期末テスト問題用紙

(2018年08月03日・10:30～12:00)

## 解答上、その他の注意事項

- I. 問題は、問I～Vまでである。うち、問I～IIは中間試験の代替問題である。
- 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70点(うち中間試験の代替問題の問I～IIの配点は30点)である。合格はレポートの得点を加点して、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ので、下の省略形(○囲み文字)を用いても良い。(必ず○で囲むこと。)

(A) ActionListener   (aA) addActionListener   (AE) ActionEvent  
(K) KeyListener   (aK) addKeyListener   (KE) KeyEvent  
(M) MouseListener   (aM) addMouseListener   (ME) MouseEvent  
(pl) System.out.println   (pf) System.out.printf

また、参考のために問題用紙の末尾に授業配布プリントのLeftRightButton3.java, LeftRightButton4.java, Guruguru.java, Point.java, ColorPoint.java のソースを掲載する。

- I. 次の枠内の文章は `java.awt.Rectangle` クラスのいくつかのメソッドと、`java.lang.Math` クラスの `random` メソッドの Java™ API 仕様からの抜粋（問題を解くのに関係ない部分は割愛）である。

**java.awt**

クラス **Rectangle**

`Rectangle` は、座標空間内の長方形オブジェクトの左上の点  $(x,y)$ 、その幅、およびその高さで囲まれた座標空間内の領域を指定します。 ...

フィールドの詳細

`public int x`

`Rectangle` の左上隅の X 座標。

`public int y`

`Rectangle` の左上隅の Y 座標。

`public int width`

`Rectangle` の幅。

`public int height`

`Rectangle` の高さ。

コンストラクターの詳細

`public Rectangle(int x, int y, int width, int height)`

左上隅が  $(x,y)$  として指定され、幅と高さが `width` 引数と `height` 引数によって指定される新しい `Rectangle` を構築します。

パラメーター:

`x` - 指定された X 座標

`y` - 指定された Y 座標

`width` - `Rectangle` の幅。

`height` - `Rectangle` の高さ。

メソッドの詳細

`public Rectangle intersection(Rectangle r)`

この `Rectangle` と指定された `Rectangle` の共通部分を計算します。2つの長方形の共通部分を表す新しい `Rectangle` を返します。2つの長方形が交差しない場合、空の長方形が返されます。

パラメーター:

`r` - 指定された `Rectangle`

戻り値:

指定された `Rectangle` とこの `Rectangle` の両方に収まる最大の `Rectangle`、これらの長方形が交差しない場合は空の長方形。

`public Rectangle union(Rectangle r)`

この Rectangle と指定された Rectangle の和集合を計算します。2つの長方形の和集合を表す新しい Rectangle を返します。 ...

パラメーター:

r - 指定された Rectangle

戻り値:

指定された Rectangle とこの Rectangle の両方を含む最小の Rectangle。

## java.lang

### クラス Math

Math クラスは、指数関数、対数関数、平方根、および三角関数といった基本的な数値処理を実行するためのメソッドを含んでいます。 ...

### メソッドの詳細

`public static double random()`

0.0 以上で 1.0 より小さい、正の符号の付いた double 値を返します。戻り値は、この範囲からの一様分布によって擬似乱数的に選択されます。 ...

戻り値:

0.0 以上 1.0 未満の擬似乱数 double。

これらのメソッドを使用するプログラムを次のように作成する。

ファイル名: RectangleTest.java

```
1 import java.awt.Color;
2 import java.awt.*;
3 import javax.swing.*;
4
5 public class RectangleTest extends JPanel {
6     private int width = 320, height = 320;
7
8     public RectangleTest() {
9         setPreferredSize(new Dimension(width, height));
10    }
11
12    private Rectangle randomRect(int min, int w, int h) {
13        int rw = (int)(Math.random() * w);
14        int rh = (int)(Math.random() * h);
15        int rx = (int)(min + Math.random() * w);
16        int ry = (int)(min + Math.random() * h);
17        return new Rectangle(rx, ry, rw, rh);
18    }
19
20    @Override
21    protected void paintComponent(Graphics g) {
22        super.paintComponent(g);
23        Rectangle rect1 = randomRect(50, width, height);
```

```

24     Rectangle rect2 = randomRect(50, width, height);
25     Rectangle union = ;
26     g.setColor(Color.BLUE);
27     g.fillRect(union.x, union.y, union.width, union.height);
28     g.setColor(Color.GREEN);
29     g.fillRect(rect1.x, rect1.y, rect1.width, rect1.height);
30     g.setColor(Color.YELLOW);
31     g.fillRect(rect2.x, rect2.y, rect2.width, rect2.height);
32     Rectangle isect = ;
33     g.setColor(Color.RED);
34     g.fillRect(isect.x, isect.y, isect.width, isect.height);
35 }
36
37 /* main メソッドは割愛 */

```

このプログラムは、再描画ごとに乱数で座標が決まる2つの長方形を緑 (green) と黄 (yellow) で描画し、さらに、2つの長方形の両方を含む最小の長方形を青 (blue) で、2つの長方形の両方に収まる最大の長方形を赤で描画する。

- 空欄 (i-1) ~ (i-4) には、4つの式が入る。(i-1) は  $\min$  以上  $w$  未満の乱数、(i-2) は  $\min$  以上  $h$  未満の乱数、(i-3) は  $0$  以上  $(w - rw)$  未満の乱数、(i-4) は  $0$  以上  $(h - rh)$  未満の乱数、を返す式が入る。Mathクラスのrandomメソッドを使って、この空欄を埋めよ。(解答欄には、(i-1) と (i-3) のみ記入せよ。)
- 空欄 (ii) には、rect1 と rect2 の両方を含む最小のRectangleを返す式が入る。この空欄を埋めよ。
- 空欄 (iii) には、rect1 と rect2 の両方に収まる最大のRectangleを返す式が入る。この空欄を埋めよ。

II. フォントの見本を表示するために、次のプログラム (FontTester クラス) を GUI アプリケーションとして作成する。これはテキストフィールドを2つ持ち、最初は、“Hello, World!” という文字列を、サイズ 16 の Times New Roman フォントで表示していて、

- 「Name:」というラベルの付いているほうのテキストフィールドに文字列を入力すると、その文字列から始まるフォント名をフォント名の一覧 (fontList) から検索して、表示されている文字列のフォントの種類を変更する。
- 「Size:」というラベルの付いているほうのテキストフィールドに整数を入力すると、表示されている文字列のフォントのサイズを変更する。

という Java プログラムである。

ファイル名: FontTester.java

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class FontTester extends JPanel implements ActionListener {

```

```

6 String fontName = "Times_New_Roman";
7 int fontSize = 16;
8 Font font = new Font(fontName, Font.PLAIN, fontSize);
9 String[] fontList = { "Times_New_Roman", "Courier_New", "Arial" };
10 JTextField nameTF, sizeTF;
11
12 public FontTester() {
13     setPreferredSize(new Dimension(320, 80));
14     nameTF = new JTextField("Times_New_Roman", 16);
15     sizeTF = new JTextField("16", 3);
16     add(new JLabel("Name:")); add(nameTF); add(new JLabel("Size:")); add(sizeTF);
17     

|       |
|-------|
| (i-1) |
| (i-2) |


19 }
20
21 private String selectFont(String init) {
22     for (String s: fontList) {
23         if (s.toLowerCase().startsWith(init.toLowerCase())) return s;
24     }
25     return Font.SERIF;
26 }
27
28 public void actionPerformed(ActionEvent e) {
29     Object target = e.getSource();
30     if (target == nameTF) {
31         fontName = selectFont(nameTF.getText());
32     } else if (target == sizeTF) {
33         fontSize = Integer.parseInt(sizeTF.getText());
34     }
35     font = new Font(fontName, Font.PLAIN, fontSize);
36     repaint();
37 }
38
39 @Override
40 protected void paintComponent(Graphics g) {
41     super.paintComponent(g);
42     g.setFont(font);
43     g.drawString("Hello, World!", 16, 64);
44 }
45
46 public static void main(String[] args) {
47     SwingUtilities.invokeLater(() -> {
48         JFrame frame = new JFrame("");
49         frame.add(new FontTester());
50         frame.pack();
51         frame.setVisible(true);
52         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
53     });
54 }

```

(i) 空欄 (i-1) ~ (i-2) にはあわせて 2 つの文が入る。この空欄を埋めよ。

さらにこの FontTester.java を匿名クラス・ラムダ式を用いて次のように同等のプログラム FontTester2.java に書き換える。

ファイル名: FontTester2.java

```

1  import java.awt.*;
2  import java.awt.event.*;
3  import javax.swing.*;
4
5  public class FontTester2 [ ] (ii) {
6      String fontName = "Times_New_Roman";
7      int fontSize = 16;
8      Font font = new Font(fontName, Font.PLAIN, fontSize);
9      String[] fontList = { "Times_New_Roman", "Courier_New", "Arial" };
10     JTextField nameTF, sizeTF;
11
12     public FontTester2() {
13         setPreferredSize(new Dimension(320, 80));
14         nameTF = new JTextField("Times_New_Roman", 16);
15         sizeTF = new JTextField("16", 3);
16         add(new JLabel("Name:")); add(nameTF); add(new JLabel("Size:")); add(sizeTF);
17         nameTF.addActionListener(
18             [ ] (iii)
19         );
20         sizeTF.addActionListener(
21             [ ] (iv)
22         );
23     }
24     /* selectFont, paintComponent, main メソッドは割愛 */
25 }
```

- 空欄 (ii) を埋めよ。
- 空欄 (iii), (iv) を埋めて、FontTester2 のコンストラクターの定義を完成させよ。ただし、空欄 (iii) には匿名クラスを、空欄 (iv) にはラムダ式を使用せよ。適宜、`/* FontTester.java の ~ 行目と同じ */` のように省略してよい。

III. 以下の選択問題に答えよ。解答は次の選択肢から選べ。

(A). Java (B). Haskell (C). C (D). Prolog

- 論理型言語に分類される言語はどれか? 一つを選べ。
- 命令型 (手続き型) 言語に分類される言語はどれか? 一つを選べ。

IV. 次に定義されるクラス Fish を継承して、  
ファイル名: Fish.java

---

```
1 public class Fish {
2     protected int age;
3     public void grow() { age++; }
4     public void showName() { System.out.print("サカナ"); }
5     public void growShowName() { showName(); grow(); }
6 }
```

---

3つのクラス Carp, Seriola, Mugil を定義する。

ファイル名: Carp.java

---

```
1 public class Carp (i) {
2     public String color;
3     public Carp(String c) { color = c; }
4
5     @Override
6     public void showName() {
7         if (age < 1000) {
8             System.out.print(color + "鯉");
9         } else {
10            System.out.print(color + "龍");
11        }
12    }
13 }
```

---

ファイル名: Seriola.java

---

```
1 public class Seriola (i) {
2     @Override
3     public void showName() {
4         if (age == 0) {
5             System.out.print("ツバス");
6         } else if (age <= 2) {
7             System.out.print("ハマチ");
8         } else {
9             System.out.print("ブリ");
10        }
11    }
12 }
```

---

ファイル名: Mugil.java

---

```
1 public class Mugil (i) {
2     @Override
3     public void showName() {
4         if (age <= 1) {
5             System.out.print("オボコ");
6         } else {
```

```
7     System.out.print("ボラ");
8     }
9     }
10 }
```

---

また、FishTest 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。

ファイル名: FishTest.java

---

```
1 public class FishTest {
2     public static void main(String[] args) {
3         Carp c = new Carp("緋");
4         c.color = "金";
5         Fish[] fish = { c, new Seriola(), new Mugil() };
6
7         int i, j;
8         for (i = 0; i < 3; i++) {
9             for (j = 0; j < 3; j++) {
10                fish[j].growShowName();
11            }
12            System.out.println();    // 改行を出力
13        }
14        for (i = 0; i < 998; i++) {
15            for (j = 0; j < 3; j++) {
16                fish[j].grow();
17            }
18        }
19        for (j = 0; j < 3; j++) {
20            fish[j].showName();
21        }
22        System.out.println();    // 改行を出力
23    }
24 }
```

---

- (i) の空欄 (3 箇所共通) を埋めて、クラスの定義を完成させよ。
- (ii) Carp クラスのフィールド color の値はコンストラクターでのみ与えることができるものとし、一旦作成した後は、クラスの外からは直接アクセスできないようにしたい。(例えば、FishTest.java の 4 行目はコンパイル時にエラーになるようにしたい。) Carp.java の何行目をどのように変更すれば良いか?( 行の左端の数字がファイルの中での行数を表す。)
- (iii) FishTest.java の 4 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? (ただし、Fish クラスおよびそのサブクラスのインスタンスが新しく生成されたときのフィールド age の初期値は 0 である。) 解答用紙に記入するとき、空白の有無や数は気にしなくて良い。

V. 下のプログラムは、アニメーションを表示する Java GUI アプリケーションである。

ファイル: TrafficLight.java

```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class TrafficLight extends JPanel  {
5     private int state = 2;
6     private volatile Thread thrd = null;
7
8     private Color changeColor(Color c, int s) {
9         if (state == s) return c;
10        return Color.DARK_GRAY;
11    }
12
13    public TrafficLight() {
14        setPreferredSize(new Dimension(300, 150));
15        JButton startBtn = new JButton("start");
16        startBtn.addActionListener();
17        JButton stopBtn = new JButton("stop");
18        stopBtn.addActionListener();
19        JButton nextBtn = new JButton("next");
20        nextBtn.addActionListener();
21        add(startBtn); add(stopBtn); add(nextBtn);
22    }
23
24    private void startAnimation() {
25        if (thrd == null) {
26            thrd = ;
27            thrd.start();
28        }
29    }
30
31    private void stopAnimation() {
32        thrd = null;
33    }
34
35    public void next() {
36        state++;
37        state %= 3;
38        repaint();
39    }
40
41    @Override
42    protected void paintComponent(Graphics g) {
43        super.paintComponent(g);
44        g.setColor(changeColor(Color.GREEN, 0));
45        g.fillOval(10, 60, 80, 80);
46        g.setColor(changeColor(Color.YELLOW, 1));
47        g.fillOval(110, 60, 80, 80);
```

```

48     g.setColor(changeColor(Color.RED, 2));
49     g.fillOval(210, 60, 80, 80);
50 }
51
52 public void run() {
53     Thread me = Thread.currentThread();
54     while ( (vi) ) {
55         next();
56         try {
57             Thread.sleep(1500); // 1.5 秒お休み
58         } catch (InterruptedException e) {}
59     }
60 }
61
62 public static void main(String[] args) {
63     SwingUtilities.invokeLater(() -> {
64         JFrame frame = new JFrame("");
65         frame.add(new TrafficLight());
66         frame.pack();
67         frame.setVisible(true);
68         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
69     });
70 }
71 }

```

---

このプログラムは、緑、黄、赤の3つの色に対応する円を描画し、“start”というボタンをクリックすると、アニメーションを開始して、約1.5秒ごとに、緑 黄 赤 緑、の順に色がつく円を切り換える（他の円は灰色にする）。“stop”というボタンをクリックすると、このアニメーションを停止する。“next”というボタンをクリックしても（アニメーションの実行中かどうかにかかわらず）、緑 黄 赤 緑、の順に色を切り換える。

(i) ~ (vi) の空欄を埋めてプログラムを完成させよ。

以下に参考のために授業配布プリントの `LeftRightButton3.java`, `LeftRightButton4.java`, `Guruguru.java`, `Point.java`, `ColorPoint.java` のソースを掲載する。

`LeftRightButton3.java`

---

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

public class LeftRightButton3 extends JPanel {
    private int x = 20;

    public LeftRightButton3() {
        setPreferredSize(new Dimension(200, 70));
        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 paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("HELLO_WORLD!", x, 55);
    }

    /* main メソッドは割愛 */
}
```

---

`LeftRightButton4.java`

---

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

public class LeftRightButton4 extends JPanel {
    private int x = 20;

    public LeftRightButton4() {
        setPreferredSize(new Dimension(200, 70));
        JButton lBtn = new JButton("Left");
        JButton rBtn = new JButton("Right");
        lBtn.addActionListener(e -> {
            x -= 10;
            repaint();
        });
        rBtn.addActionListener(e -> {
            x += 10;
            repaint();
        });
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

    @Override
    public void paintComponent(Graphics g) {
```

```

    super.paintComponent(g);
    g.drawString("HELLO_WORLD!", x, 55);
}

/* main メソッドは割愛 */
}

```

---

Guruguru.java

```

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

public class Guruguru extends JPanel implements Runnable {
    private int r = 50, x = 110, y = 70;
    private double theta = 0; // 角度
    private volatile Thread thread = null;

    public Guruguru() {
        setPreferredSize(new Dimension(200, 180));
        JButton startBtn = new JButton("start");
        startBtn.addActionListener(e -> startThread());
        JButton stopBtn = new JButton("stop");
        stopBtn.addActionListener(e -> stopThread());
        setLayout(new FlowLayout());
        add(startBtn); add(stopBtn);
        startThread();
    }

    private void startThread() {
        if (thread == null) {
            thread = new Thread(this);
            thread.start();
        }
    }

    private void stopThread() {
        thread = null;
    }

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(g); // スーパークラスの paintComponent を呼び出す
        // 全体を背景色で塗りつぶす。
        g.drawString("Hello, World!", x, y);
    }

    public void run() {
        Thread thisThread = Thread.currentThread();
        for (; thread == thisThread; theta += 0.02) {
            x = 60 + (int)(r * Math.cos(theta)); y = 100 - (int)(r * Math.sin(theta));
            repaint(); // paintComponent を間接的に呼出す
            try {
                Thread.sleep(30); // 30 ミリ秒お休み
            } catch (InterruptedException e) {}
        }
    }

    /* main メソッドは割愛 */
}

```

---

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。
    }

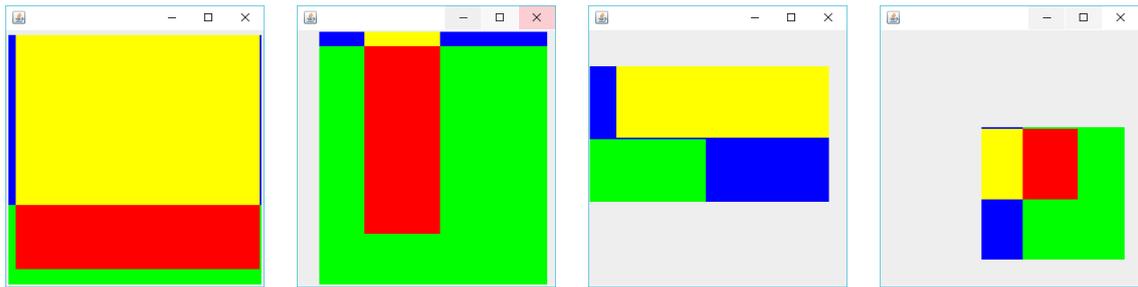
    public ColorPoint(int x, int y, String c) {
        super(x, y);
        setColor(c);
        if (color == null) color = "black";
    }

    public String getColor() { return color; }
}

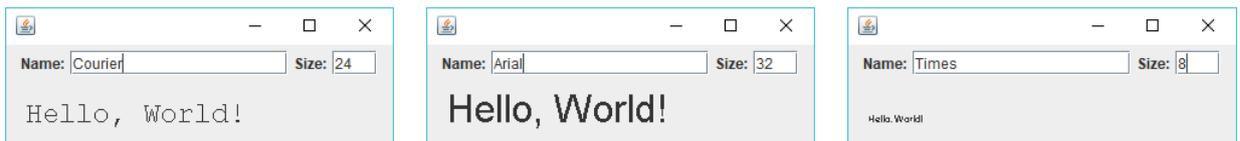
```

---

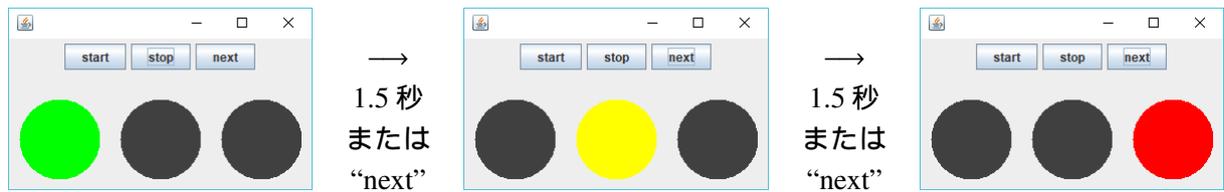
### RectangleTest.java のスクリーンショット



### FontTester.java のスクリーンショット



### TrafficLight.java のスクリーンショット





III. ( 3 × 2 )

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

IV. ( 4, 6, 6 )

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

V. ( 3 × 6 )

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

授業・テストの感想

---

.....

.....

.....

.....

.....

.....

.....

.....

.....

---