

オブジェクト指向言語・期末テスト問題用紙

(2019年08月02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。うち、問 I～III は中間テストの代替問題である。
 - (i) 各大問 (I, II, III) ごとに、中間テストの得点が 6 割に達しなかったものは、対応する代替問題の大問を解答すれば、6 割を上限として良いほうの点数を採用する。
 - (ii) 事情があって中間テストを欠席した者も代替問題を解答すること。(この場合は、上限は設けない。)
- 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 (うち中間テストの代替問題の問 I～III の配点は 40 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。)

Ⓐ ActionListener ⓐA addActionListener ⒶE ActionEvent
Ⓚ KeyListener ⓐK addKeyListener ⓀE KeyEvent
Ⓜ MouseListener ⓐM addMouseListener ⓂE MouseEvent
Ⓟ System.out.println Ⓟf 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 MouseTest.java, LeftRightButton.java, LeftRightButton3.java, LeftRightButton4.java, Guruguru.java, Point.java, ColorPoint.java のソースを掲載する。(main メソッドは省略している。)

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) 文法上 Java のクラス名として使うことができるのは、どれか?
(A). M-1 (B). 2000percent (C). Off_White (D). NMB48
- (ii) Java で要素の型が `double[]` 型であるような `ArrayList` 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<double[]> xs = new ArrayList<>();`
(B). `double[]<ArrayList> xs = new double[]<>();`
(C). `ArrayList[double<>] xs = new ArrayList[]();`
(D). `double<ArrayList[]> xs = new double<>();`

II. 次の枠内の文章は `java.awt.event.MouseEvent` クラスの `getClickCount` メソッドと、`java.lang.Math` クラスの `atan2` 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.event

クラス **MouseEvent**

コンポーネント内でマウスアクションが発生したことを示すイベントです。...

メソッドの詳細

`public int getClickCount()`

このイベントに関連付けられたマウスクリック数を返します。

戻り値:

クリック数を表す整数値

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 atan2(double y, double x)`

直交座標 (x, y) から極座標 (r, θ) への変換から得られる角度 θ を返します。このメソッドは、 $-\pi - \pi$ の範囲で y/x の逆正接 (アーク・タンジェント) を計算することで、位相 θ (シータ) を計算します。...

パラメータ:

y - 縦座標

x - 横座標

戻り値:

直交座標 (デカルト座標) 上の点 (x, y) に対応する極座標上の点 (r, θ) の、 θ (シータ) 成分。

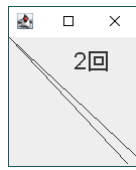
これらのメソッドを使用するプログラムを次のように作成する。

ファイル名: MouseTrainer.java

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class MouseTrainer extends JPanel (i) {
6     private int count = 0;
7     private static final double t1 = 43 * Math.PI / 180,           // 43 度
8                               t2 = 47 * Math.PI / 180;           // 47 度
9
10    public MouseTrainer() {
11        setPreferredSize(new Dimension(128, 128));
12        addMouseListener(this);
13    }
14
15    @Override
16    protected void paintComponent(Graphics g) {
17        super.paintComponent(g);
18        // 目安となる線分を描く
19        g.drawLine(0, 0, (int)(172 * Math.cos(t1)), (int)(172 * Math.sin(t1)));
20        g.drawLine(0, 0, (int)(172 * Math.cos(t2)), (int)(172 * Math.sin(t2)));
21        g.setFont(new Font(Font.SANS_SERIF, Font.PLAIN, 24));
22        g.drawString(count + "回", 64, 32); // 回数を表示する
23    }
24
25    public void mouseClicked(MouseEvent e) {
26        int x = e.getX(), y = e.getY();
27        double theta = (ii);
28        if (t1 <= theta && theta <= t2 && (iii) >= 2) {
29            count++;
30            repaint();
31        }
32    }
33
34    /* 他の MouseListener のメソッドと main メソッドは割愛 */
35 }
```

このプログラムは、原点 $(0, 0)$ とマウスをクリックした点 (x, y) を結ぶ線分と x 軸のなす角が 43 度以上 47 度以下でかつダブルクリックのとき (クリック数が 2 以上のとき) に、回数をインクリメントしそれを表示する。

実行例は次のようになる。



2回ダブルクリックしたとき さらに3回ダブルクリックしたとき

- 空欄 (i) を埋めよ。
- 空欄 (ii) には、原点 $(0, 0)$ とマウスをクリックした点 (x, y) を結ぶ線分と x 軸のなす角を求める式が入る。この空欄を埋めよ。
- 空欄 (iii) には、マウスのクリック数を求める式が入る。この空欄を埋めよ。

III. 漸化式

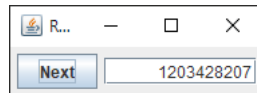
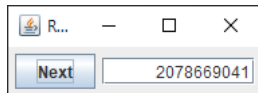
$$x_{n+1} = 48271 \times x_n \text{ mod } 2147483647$$

に基づく擬似乱数列を表示するために、次のプログラム (RandomTester クラス) を GUI アプリケーションとして作成する。これはボタンを1つとテキストフィールドを1つ持ち、最初は1という整数をテキストフィールドに表示していて、

- ボタンをクリックすると、現在テキストフィールドに表示している整数から、漸化式に基づき次の乱数を計算して、テキストフィールドに表示する。
- テキストフィールドに整数を入力すると、その数を次の乱数の“種”として設定する。

という Java プログラムである。

実行例は次のようになる。



数回クリックしたとき ある整数を入力して数回クリックしたとき

ファイル名: RandomTester.java

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class RandomTester extends JPanel implements ActionListener {
6     private long x = 1;
7     private JButton button;
8     private JTextField field;
9
10    public RandomTester() {
11        button = new JButton("Next");
12        (i)
13        field = new JTextField("1", 10);
```

```

14     field.setHorizontalAlignment(JTextField.RIGHT); // 右詰めにする
15     
16     add(button); add(field);
17 }
18
19 public void actionPerformed(ActionEvent ev) {
20     Object source = ;
21     if (source == button) {
22         x = 48271 * x % 2147483647;
23         field.setText("" + x);
24     } else if (source == field) {
25         try {
26             x = Integer.parseInt(field.getText());
27         } catch (NumberFormatException ex) {}
28         field.setText("" + x);
29     }
30
31 }
32
33 /* main メソッドは割愛 */
34 }

```

- 空欄 (i) ~ (iii) を埋めよ。

さらにこの RandomTester.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 RandomTester2.java に書き換える。

ファイル名: RandomTester2.java

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class RandomTester2  {
6     private long x = 1;
7
8     public RandomTester2() {
9         JButton button = new JButton("Next");
10        JTextField field = new JTextField("1", 10);
11        button.addActionListener(
12            
13        );
14
15        field.setHorizontalAlignment(JTextField.RIGHT);
16        field.addActionListener(
17            
18        );
19        add(button); add(field);
20    }

```

```
21
22  /* main メソッドは割愛 */
23 }
```

- 空欄 (iv) を埋めよ。
- 空欄 (v), (vi) を埋めて、RandomTester2 のコンストラクターの定義を完成させよ。ただし、空欄 (v) には匿名クラスを、空欄 (vi) にはラムダ式を使用せよ。適宜、`/* RandomTester.java の ~ 行目と同じ */` のように省略してよい。

【中間テストの代替問題はここまで】

IV. 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). COBOL (C). Haskell (D). Prolog

- (i) この中で関数型言語に分類される言語はどれか? 一つを選べ。
- (ii) この中で最も古くからある言語はどれか? 一つを選べ。

V. 次の Python のリスト内包表記の値は何か?

- (i) `[x * y for x in [1,3,7] for y in [2,5]]`
- (ii) `[x for x in [2,3,4] if x * x % 3 == 1]`

VI. 次に定義されるクラス `animals.Animal` を継承して、
ファイル名: `animals/Animal.java`

```
1 package animals;
2 public class Animal {
3     protected int age;
4     public Animal() { age = 0; }
5     public void say() {
6         System.out.print("...");
7         age++;
8     }
9 }
```

3つのクラス `animals.Dog`, `animals.Cat`, `animals.Pheasant` を定義する。

ファイル名: `animals/Dog.java`

```
1 package animals;
2 public class Dog [ ] (i) {
3     public Dog() { super(); }
4     @Override
5     public void say() {
6         if (age < 1) {
7             System.out.print("Kyan_");
8         } else {
9             System.out.print("Wan_");
10        }
11        age++;
12    }
13 }
```

ファイル名: `animals/Cat.java`

```
1 package animals;
2 public class Cat [ ] (i) {
3     public Cat() { super(); }
4     @Override
5     public void say() {
6         System.out.print("Nyaa_");
7         age++;
8     }
9 }
```

ファイル名: `animals/Pheasant.java`

```
1 package animals;
2 public class Pheasant [ ] (i) {
3     private boolean female; // false -- 雄, true -- 雌
4     public Pheasant(boolean f) { super(); female = f; }
5     @Override
6     public void say() {
```

```
7     if (age < 1) {
8         System.out.print("Piii_");
9     } else if (female) {
10        System.out.print("Chocho_");
11    } else {
12        System.out.print("Keeen_");
13    }
14    age++;
15 }
16 }
```

また、(匿名パッケージの) `AnimalTest` 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: `AnimalTest.java`

```
1 public class AnimalTest {
2     public static void foo(Animal a) {
3         int i;
4         for(i = 0; i < 3; i++) {
5             a.say(); System.out.print("\t");
6         }
7         System.out.println();
8     }
9
10    public static void main(String[] args) {
11        Dog d = new Dog();
12        Cat c = new Cat();
13        Pheasant p = new Pheasant(false);
14        p.female = true;
15        // Dog, Cat, Pheasant
16        foo(d); foo(c); foo(p);
17    }
18 }
```

- (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) `animals.Pheasant` クラスのフィールド `female` の値はクラスの外からは直接アクセスできないため、`AnimalTest.java` の 14 行目はコンパイル時にエラーになるはずである。デバッグのため、一時的にこれをエラーにならないようにしたい。
`animals.Pheasant` クラスのフィールド `female` の可視性を変えて `AnimalTest.java` の 14 行目がエラーにならないようにするためには、`animals/Pheasant.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) `AnimalTest.java` の 14 行目がエラーにならないように `animals/Pheasant.java` を変更し、`AnimalTest` クラスの `main` メソッドを実行するとき、出力はどうなるか?
なお、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。

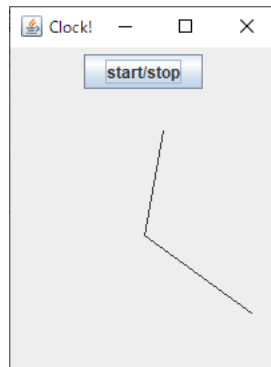
VII. 下のプログラムは、アニメーションを表示する Java GUI アプリケーションである。

ファイル: ClockSimulator.java

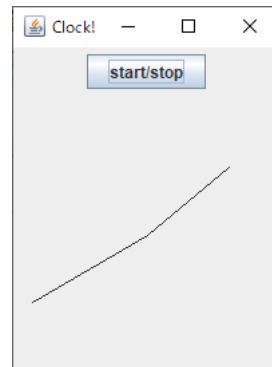
```
1 import java.awt.*;
2 import javax.swing.*;
3
4 public class ClockSimulator extends JPanel (i) {
5     private int min = 0;
6     private volatile Thread thread = null;
7     private final static int cx = 100, cy = 140; // 中心の座標
8
9     public ClockSimulator() {
10         setPreferredSize(new Dimension(200, 240));
11         JButton toggleBtn = new JButton("start/stop");
12         toggleBtn.addActionListener(e -> (ii));
13         add(toggleBtn);
14         toggleThread();
15     }
16
17     private void toggleThread() {
18         if (thread == null) {
19             thread = new Thread(this);
20             (iii);
21         } else {
22             (iv);
23         }
24     }
25
26     @Override
27     public void paintComponent(Graphics g) {
28         super.paintComponent(g);
29         int m = min % 60;
30         double tm = m / 30.0 * Math.PI; // 長針は 60 分で一周
31         double x = Math.sin(tm), y = -Math.cos(tm);
32         g.drawLine(cx, cy, cx + (int)(100 * x), cy + (int)(100 * y));
33         int h = min % 720;
34         double th = h / 360.0 * Math.PI; // 短針は 720 分で一周
35         x = Math.sin(th); y = -Math.cos(th);
36         g.drawLine(cx, cy, cx + (int)(80 * x), cy + (int)(80 * y));
37     }
38
39     public void run() {
40         Thread me = Thread.currentThread();
41         while ((v)) {
42             min++;
43             repaint(); // paintComponent を間接的に呼出す
44             try {
45                 Thread.sleep(100);
46             } catch (InterruptedException e) {}
47         }
48     }
49 }
```

```
48 }  
49  
50 /* main メソッドは割愛 */  
51 }
```

このプログラムは、時計をシミュレートする。ただし、約 100 ミリ秒ごとに 1 分進む。また “start/stop” というボタンをクリックするごとに、アニメーションの実行・停止を切り替える。実行例は次のようになる。



数秒経過したとき



さらに数秒経過したとき

(i) ~ (v) 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 `MouseTest.java`, `LeftRightButton.java`, `LeftRightButton3.java`, `LeftRightButton4.java`, `Guruguru.java`, `Point.java`, `ColorPoint.java` のソースを掲載する。

`MouseTest.java`

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

public class MouseTest extends JPanel implements MouseListener {
    private int x = 50, y = 20;

    public MouseTest() {
        setPreferredSize(new Dimension(150, 150));
        addMouseListener(this);
    }

    public void mouseClicked(MouseEvent e) {
        x = e.getX(); y = e.getY();
        repaint();
    }

    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}

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

    /* main メソッドは割愛 */
}
```

`LeftRightButton.java`

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

public class LeftRightButton extends JPanel implements ActionListener {
    private int x = 20;
    private JButton lBtn, rBtn;

    public LeftRightButton() {
        setPreferredSize(new Dimension(200, 70));
        lBtn = new JButton("Left");
        rBtn = new JButton("Right");
        lBtn.addActionListener(this);
        rBtn.addActionListener(this);
        setLayout(new FlowLayout());
        add(lBtn); add(rBtn);
    }

    @Override
    public void paintComponent(Graphics g) {
        super.paintComponent(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();
}
}
/* main メソッドは割愛 */
}

```

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; }
}
```

このページは計算用紙に使用しても良い。
【冊子から切り離しても良い。】

このページは計算用紙に使用しても良い。
【冊子から切り離しても良い。】

オブジェクト指向言語・期末テスト解答用紙 (2019年 08月 02日)

学籍番号		氏名	
------	--	----	--

I. (3 × 2)

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

II. (4, 6, 6)

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

III. (1, 1, 3, 3, 5, 5)

(i).	
(ii).	
(iii).	
(iv).	
(v).	<pre>button.addActionListener(匿 名 ク ラ ス);</pre>
(vi).	<pre>field.addActionListener(ラ ム ダ 式);</pre>

IV. (3 × 2)

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

V. (3 × 2)

(i).	
(ii).	

VI. (3, 5, 5)

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

VII. (3 × 5)

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

授業・テストの感想

.....

.....

.....

.....

.....

.....

.....
